

UNTERSUCHUNG DER EIGNUNG DER WINDOWS SHAREPOINT SERVICES
ALS PLATTFORM ZUR VERTEILTEN MODELLIERUNG MIT SEMTALK

Diplomarbeit zur Erlangung des akademischen Grades
Diplom-Wirtschaftsinformatiker (FH)
an der Fachhochschule Brandenburg

vorgelegt von

Richard Ernst

Brandenburg an der Havel, 2006

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Problemstellung.....	1
1.2	Anliegen.....	2
1.3	Aufbau der Arbeit.....	3
2	Grundlagen.....	6
2.1	Verteilte Modellierung mit SemTalk.....	12
2.1.1	Verfahren zur verteilten Modellierung mit SemTalk.....	14
2.1.2	Exkurs: Referenzmodelle – Modellierung und Nutzung mit SemTalk.....	22
2.1.3	Exkurs: Zyklisch referenzierte Objekte und verteilte Datenbanken.....	24
2.2	Methoden und Organisation.....	25
2.2.1	Vorgehensmodelle nach Sure.....	26
2.2.2	Vorgehensmodell nach Simon.....	28
2.2.3	Vorgehensmodell nach Weichhardt.....	29
2.2.4	Muster in den Vorgehensmodellen.....	29
2.3	Plattformen für die gemeinsame Erstellung von Ontologien.....	31
2.4	Die Windows SharePoint Services.....	34
2.4.1	Personalisierung.....	37
2.4.2	Benutzerverwaltung und Rechte.....	40
2.4.3	Dynamische Inhalte und Webpublishing.....	41
2.4.4	Webanwendungen.....	43
2.4.5	Microsoft Office System Integration.....	44
2.4.6	Unterstützung von Zusammenarbeit.....	47
3	Idee, Anforderungen und Entwurf.....	50
3.1	Windows SharePoint Services als Plattform für die verteilte Modellierung.....	50
3.2	Anforderungen an eine Plattform zur verteilten Modellierung.....	52
3.2.1	Anforderungen aus bisherigen Untersuchungen.....	53
3.2.2	Eigene Untersuchungen.....	56
3.2.3	Zusammenfassung und erste Gegenüberstellung.....	61
3.2.4	Unterstützung bestehender Techniken zur verteilten Modellierung.....	70
3.3	Entwurf der Erweiterung für die verteilte Modellierung.....	71
3.3.1	Verfügbare Technologien.....	71
3.3.2	Distribution-Index.....	82
3.3.3	Aktualisierung des Distribution-Indexes.....	86
4	Umsetzung der Komponenten.....	99
4.1	Codezugriffssicherheit im .NET-Framework.....	99
4.2	Anpassung von SemTalk.....	100
4.3	Integration der entwickelten Komponenten.....	103
4.4	Der Webdienst und die Codezugriffssicherheit.....	105
5	Abschließende Betrachtung.....	108
5.1	Bewertung der ausgearbeiteten Lösung.....	108
5.1.1	Strategische und wirtschaftliche Aspekte.....	108
5.1.2	Bewertung des Nutzens.....	110
5.2	Ausblick.....	114
	Anhang I – Vorgehensmodelle.....	116
	Anhang II – Referenzarchitektur für Portalsoftware.....	120
	Anhang III – Fragebogen.....	121
	Anhang IV – Entwurf.....	124
	Anhang V – UML Sequenzdiagramme.....	125

Anhang VI – Tabelle der Gewichtungen.....	129
Glossar.....	130
Literatur.....	135

Abbildungsverzeichnis

Abbildung 1. Abdeckung der vier Anforderungs-Bereiche.....	2
Abbildung 2. Thematische Zusammenhänge in dieser Arbeit.....	4
Abbildung 3. Verteiltes Modellieren.....	7
Abbildung 4. Gemeinsames Modellieren.....	7
Abbildung 5. Team-Organisation.....	12
Abbildung 6. Stablinienorganisation.....	12
Abbildung 7. Externe Verfeinerung einer Aktivität in der KSA-Methode.....	15
Abbildung 8. Der globale Geschäftsprozess Einkauf als externe Verfeinerung der lokalen Prozesse.....	19
Abbildung 9. Hinweis auf eine externe Verfeinerung.....	20
Abbildung 10. Merlin mit Hinweis über ein vorhandenes externes Objekt.....	21
Abbildung 11. Zyklisch gerichteter Graph von externen Referenzen in SemTalk.....	24
Abbildung 12. Semantic Collaborative Environment Architecture. Quelle: [Hodgson 2005:29]	33
Abbildung 13. Hierarchie von Sites in den WSS. Quelle: [Microsoft 2003c].....	35
Abbildung 14. Eine typische Webpartseite.....	37
Abbildung 15. Hinzufügen eines Webparts auf eine Webpart-Seite.....	38
Abbildung 16. Webpart mit Liste auf einer ASPX-Webseite der WSS.....	42
Abbildung 17. Architektur der WSS.....	44
Abbildung 18. Kontextmenü einer MS Visio-Datei in einer Dokumentbibliothek.....	45
Abbildung 19. Freigegebener Arbeitsbereich, eingeblendet in MS Visio.....	45
Abbildung 20. Das 3-K-Modell. Quelle: [Borghoff; Schlichter 1998:128].....	48
Abbildung 21. Die WSS als Plattform für die verteilte Modellierung.....	50
Abbildung 22. Überprüfung der Konsistenz in SemTalk – Anwendungsfälle.....	67
Abbildung 23. Verwaltung externer Referenzen durch die WSS – Anwendungsfälle.....	68
Abbildung 24. Verwaltung externer Verfeinerungen durch die WSS – Anwendungsfälle.....	69
Abbildung 25. Verteilte Modellierung mit SemTalk und den WSS – Anwendungsfälle für SemTalk.....	70
Abbildung 26. Verwalteter und nicht-verwalteter Code in den WSS. Quelle: [Microsoft 2004].....	73
Abbildung 27. JavaScript-Funktion: Am Quellort veröffentlichen.....	81
Abbildung 28. Die externe Referenz als gerichteter Graph.....	83
Abbildung 29. Distribution-Index mit Informationen über sämtliche externen Referenzen....	83
Abbildung 30. Struktur-Baum des Distribution-Index.....	86
Abbildung 31. Klassendiagramm der Ereignisroutine und des semtalk-Webdienstes.....	90
Abbildung 32. Klasse ModelLibEvtHndler.EventHandler und Schnittstelle IListEventSink...	91
Abbildung 33. Klasse semtalk.DistributionIndex.....	92
Abbildung 34. Klasse semtalk.ReferencesManager.....	93
Abbildung 35. Klasse semtalk.Email.....	93
Abbildung 36. Klasse semtalk.TaskList.....	94
Abbildung 37. Ein Modell, geöffnet in MS Visio mit SemTalk-Add-In und Anzeige einer automatisch erzeugten Aufgabe.....	101
Abbildung 38. EventHandler-Klasse, gebunden an eine Dokumentbibliothek.....	104
Abbildung 39. Klasse ModelLibEvtHndler.EventHandler.....	105
Abbildung 40. Eigenschaften der Ereignisroutine ModelLibEvtHandler.EventHandler.....	107
Abbildung 41. Klassische Portfolio-Analyse.....	109

Abkürzungen

API	Application Programming Interface
ASP	Active Server Pages
CAML	Collaborative Application Markup Language
CAS	Code Access Security
DOM	Document Object Model
GAC	Global Assembly Cache
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IE	Microsoft Internet Explorer
IIS	Internet Information Server
MS	Microsoft Corporation
MSSQL	Microsoft SQL Server
RPC	Remote Procedure Call
SPS	Microsoft Office SharePoint Portal Server
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WebDAV	Web-based Distributed Authoring and Versioning
WMSDE	Windows Microsoft SQL Desktop Engine
WSS	Microsoft Windows SharePoint Services
XML	Extensible Markup Language

1 Einleitung

1.1 Problemstellung

Die Zusammenarbeit mehrerer Personen zur Erreichung eines Ziels durchdringt neben den traditionellen Anwendungsgebieten der Wirtschaft und Industrie zunehmend den Erstellungsprozess von Produkten der Informationstechnik. Zu diesen Produkten gehören in der Wirtschaftsinformatik Modelle, in denen betriebliche Abläufe in Geschäftsprozessmodellen und Wissen in Wissensmodellen abgebildet werden.

Ist die Komplexität des abzubildenden Systems besonders hoch und wird deshalb für eine einzelne modellierende Person unbeherrschbar, besteht eine Lösung in dem auf mehrere Personen verteilten Modellieren. Beispiele für derart komplexe Probleme finden sich in den Forschungsbereichen Semantic Web und Künstliche Intelligenz der Informatik. In diesen muss umfangreiches Wissen in maschinenlesbare Form umgewandelt werden. Wissensmodelle sind die dafür geeignete formale Darstellungsform, da sie sowohl von Maschinen als auch von Menschen gelesen werden können.

Die wesentlichsten Vorteile bei der verteilten Modellierung sind die Zeitersparnis, die aus dem parallelen Vorgehen resultiert, sowie die Einbeziehung und aktive Mitwirkung von Beteiligten und Experten des zu modellierenden betrieblichen Ablaufs oder des Fachgebiets. Die Modellierung selbst ist ein kreativer Prozess. Eine Herausforderung bei der verteilten Modellierung besteht deshalb in dem Erhalt des Freiraums für den Einzelnen bei gleichzeitig zu sichernder Kooperationsfähigkeit mit anderen an einem Modellierungsprojekt beteiligten Personen [Vom Brocke 2003b:2].

Damit ein verteilt durchgeführtes Modellierungsprojekt erfolgreich durchgeführt werden kann, müssen Anforderungen erfüllt werden, die es zu ermitteln gilt. Um den dafür nötigen Überblick zu erlangen, werden die in [Vom Brocke 2003b] angeführten „Gestaltungsanforderungen an die verteilte Referenzmodellierung“ für diese Arbeit als Rahmenwerk eingeführt. [Vom Brocke 2003b:2f] unterteilt die Gestaltungsanforderungen in vier Aspekte. Hiernach kann eine Anforderung der Organisationsform, der technologischen Plattform, der Kopplung der Modelle oder der Methodik zugeordnet werden. Der Schwerpunkt des *organisationsbezogenen Aspektes* liegt auf Abstimmungsprozessen, die zur Wahrung der Flexibilität während der Modellierung bei gleichzeitigem Erhalt der Stabilität definiert werden müssen. Für die Unterstützung der Kooperations- und Koordinationsprozesse der beteiligten Modellierer ist eine technische Plattform einzusetzen.¹

¹ Siehe hierzu den Forschungsbericht 2003-2004 zu dem Thema „Verteilte Referenzmodellierung (VRM)“ des Lehrstuhls für Wirtschaftsinformatik und Controlling am Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster. Erreichbar unter <<http://www.uni->

Diesbezügliche Anforderungen werden in dem *technologiebezogenen Aspekt* zusammengefasst. Während der Gegenstand des *modellbezogenen Aspektes* darin besteht, die Wahrung der Eigenständigkeit der Modelle einerseits und die Ermöglichung ihrer Kopplung andererseits zu gewährleisten, vereint der *methodenbezogene Aspekt* die Fragen zum Vorgehen bei verteilt durchgeführten Modellierungsprojekten.

Schon heute existieren Produkte und Prinzipien, die viele diese Anforderungen bereits erfüllen. Ein Vertreter ist *SemTalk*, ein Werkzeug zur ontologiebasierten Modellierung von Geschäftsprozessen und Wissen der Sementation GmbH (Potsdam). *SemTalk* unterstützt wesentlich den Bereich des Modellbezugs, indem es mehrere Techniken anbietet, die eine Kopplung separater Modelle erlaubt.

Fragen zum Methoden- sowie dem Organisationsbezug wurden in den auf *SemTalk* bezogenen Arbeiten von [Simon 2004] und [Weichhardt 2005] an beispielhaften Szenarien erörtert. Eine wieder verwendbare Konfiguration, die diese Untersuchungen berücksichtigt, wurde bislang noch nicht umgesetzt.

Der technologiebezogene Aspekt ist – wie in Abbildung 1 veranschaulicht – bisher kaum diskutiert worden. Würde eine geeignete Plattform die Anforderungen aus diesem Bereich abdecken und damit die kooperativen Prozesse adäquat unterstützen, stünde eine umfassende Lösung für ein verteiltes Modellierungsprojekt zur Verfügung.

1.2 Anliegen

Die vorliegende Arbeit zielt darauf ab, zu untersuchen, inwieweit die Microsoft Windows SharePoint Services (WSS) geeignet sind, als technische Plattform für die verteilte Modellierung zu fungieren und so die Anforderungen aus dem technologiebezogenen Bereich abzudecken. Gleichzeitig soll mit der Einführung und Nutzung der WSS die herkömmliche Lösung für verteiltes Modellieren – das Ablegen von *SemTalk*-Modellen in freigegebenen Verzeichnissen in einem lokalen Netzwerk – abgelöst werden.

Bei den WSS handelt es sich um zusätzliche Dienste für den Microsoft Windows Server 2003, die nicht im standardmäßigen Funktionsumfang enthalten sind. Sie haben die

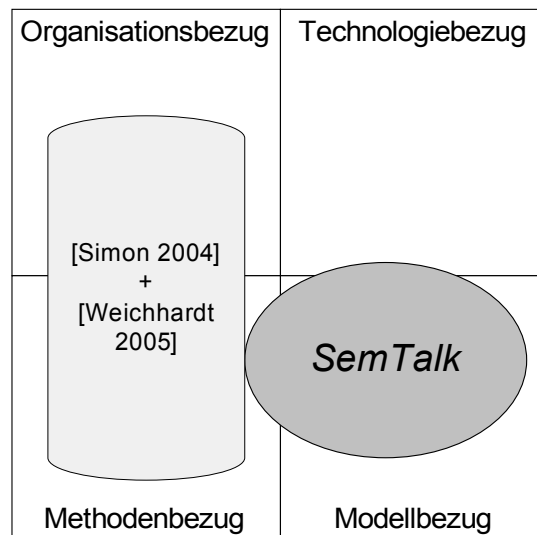


Abbildung 1. Abdeckung der vier Anforderungsbereiche

Aufgabe, gemeinsame Arbeitsbereiche für MS-Office-Dokumente zur Verfügung zu stellen. Sie zeichnen sich neben den Funktionen zur Unterstützung der Zusammenarbeit durch gute Anpassungs- und Erweiterungsmöglichkeiten aus.

Es wird gezeigt, dass die während eines verteilten Modellierungsvorhabens auftretenden kooperativen und kommunikativen Prozesse durch den Einsatz der WSS unterstützt werden können. Die Fähigkeit der WSS, ereignisgesteuert Anwendungslogik auszuführen, wird am Beispiel der automatischen Verwaltung von externen Referenzen – einem Mechanismus SemTalks zur Kopplung separater Modelle – angewandt und der damit erzeugte Nutzen für die verteilte Modellierung aufgezeigt.

Das Potential der WSS, als Plattform für die verteilte Modellierung zu dienen, wird in der abschließenden Bewertung dargelegt. Dafür werden zwei Werkzeuge zur Bewertung von IT-Vorhaben – die Portfolio- und die Nutzwertanalyse – verwendet.

1.3 Aufbau der Arbeit

Zunächst werden die für den Einstieg in die Thematik der verteilten Modellierung nötigen Grundlagen im Kapitel 2 gegeben. Den Ausgangspunkt bilden Anwendungsfälle aus allgemeiner und SemTalk-spezifischer Sicht. Nach Erläuterung der Möglichkeiten und der Begriffe der verteilten Modellierung mit SemTalk werden verschiedene Vorgehensmodelle und Organisationsstrukturen beschrieben. Dazu gehören auch SemTalk-fremde Vorhaben, deren Ziel das gemeinsame Erstellen von Modellen war. Hieran schließt sich die nähere Betrachtung der WSS an: Ihre Funktionsweise wird beschrieben und es erfolgt ihre Einordnung im Kontext des Microsoft Office Systems. Darüberhinaus wird gezeigt, dass es sich bei den WSS um eine Portalsoftware handelt, die gleichzeitig typische Funktionen einer Kollaborationstechnologie bereitstellt.

Im Kapitel 3 wird dargelegt, warum die WSS als Plattform für die Unterstützung der verteilten Modellierung geeignet sein könnten. Die an die Plattform zu stellenden Anforderungen werden (i) anhand der Resultate der im Vorfeld vorgestellten Untersuchungen, (ii) aus dem Studium der vorhandenen Dokumentationen zu SemTalk und (iii) eigenen Befragungen bei Beteiligten verteilt durchgeführter Modellierungsprojekte ermittelt. Besondere Beachtung verdienen dabei die vonseiten der Beteiligten geäußerten Anforderungen und Wünsche, sofern diese auf das verteilte Modellieren im Speziellen abzielen. Darauf folgend wird ermittelt, inwieweit die Anforderungen bereits von dem standardmäßigen Funktionsumfang der WSS abgedeckt werden.

Die noch nicht erfüllten Anforderungen bilden den Ausgangspunkt für die Erstellung eines Entwurfs, dessen Umsetzung mittels der programmiertechnischen Anpassung und Erweiterung der WSS erfolgt. Die dafür zur Verfügung stehenden Technologien sind zu untersuchen und für eine Lösung auszuwählen.

Nach Vorstellung der Lösung schließt sich ihre prototypische Umsetzung an. Sie erfolgt in Zusammenarbeit mit der Semtation GmbH, von der die benötigte Hard- und Software zur Verfügung gestellt wird. Die Bewertung der geschaffenen Lösung, bestehend aus den WSS und den eigenen Komponenten, bildet den Abschluß der Arbeit

Die für diese Arbeit relevanten Themen und die zwischen ihnen geltenden Zusammenhänge sind in dem Wissensmodell in Abbildung 2 grafisch dargestellt. Die in ovalen Elementen stehenden Begriffe sind über Beziehungen (Pfeile) miteinander verbunden. Geschlossene, unausgefüllte Pfeilspitzen stehen für eine Beziehung mit der Aussage *ist-ein* und zeigen auf einen hierarchisch übergeordneten Begriff. Die Pfeile mit den kleineren, ausgefüllten Spitzen stehen für Abhängigkeiten zwischen den Begriffen, der Kenntlichmachung von Synonymen etc.

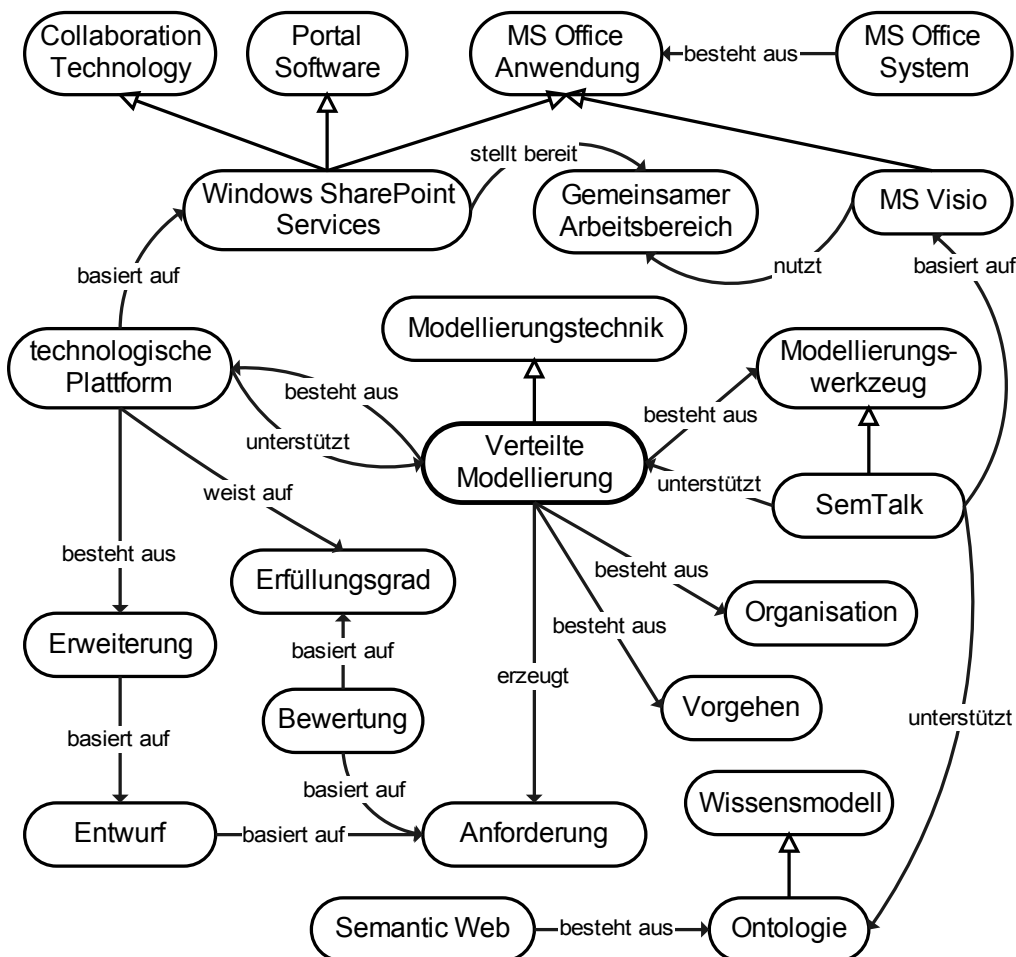


Abbildung 2. Thematische Zusammenhänge in dieser Arbeit

Zur Förderung der Lesbarkeit sind die im Folgenden beschriebenen typografischen Vereinbarungen in dieser Arbeit gültig.

Texttyp	Verwendete Schrift
Textkörper	Garamond
Hervorhebung, Fremdsprachliches	<i>Garamond, kursiv</i>
Auszüge aus dem Quellcode (<i>Listings</i>)	Courier New
Internet-Adressen (URL)	Courier New

Dem Leser werden darüberhinaus die hier fehlende, aber ansonsten weitgehend gebräuchliche, abschließende Interpunktion bei Abkürzungen auffallen. Diese Schreibweise ist kompakter und unterstreicht die Bedeutung des Punktes als das Zeichen für ein Satzende.² Werden im laufenden Text Inhalte aus fremder Literatur zitiert oder sinngemäß wiedergegeben, erfolgt die dazugehörige Quellenangabe auf die Bibliografie bezogen und in der verkürzten *Harvard-Notation*.³ Englische Begriffe werden, soweit eine Übersetzung nicht nötig scheint, in ihrer originalen Schreibweise belassen.

²Standop spricht hier von einer *fortschrittlichen Zeichensetzung* [Ewald Standop; Matthias L.G. Meyer. 2004. *Die Form der wissenschaftlichen Arbeit*. 17.Auflage Wiebelsheim: Quelle & Meyer. Seite 177].

³In eckigen Klammern folgt dem Nachnamen des Autoren (oder der Autoren) und das Erscheinungsjahr. Die Seitenzahl ist hiervon durch Doppelpunkt getrennt. Beispiel dazu: [Hillier 2005:33]. Standop; Meyer: Seite 162.

2 Grundlagen

Das Streben des Menschen ist auf die Erkenntnis von Systemen⁴ der realen oder der gedanklichen Welt ausgerichtet. Die Beschaffenheit des menschlichen Gehirns ist dafür verantwortlich. Es versucht mit Hilfe der Wahrnehmung Muster zu erkennen, um Dinge zu verstehen und Handlungen wiederholen zu können. Bei dem menschlichen Gehirn handelt es sich folglich um ein selbstorganisiertes, musterbildendes Informationssystem [De Bono 1993:25f].

Das betrachtete System kann sich auf Grund seiner Komplexität dem Verständnis einer Person entziehen [Krallmann ua 1999:28].⁵ Eine Möglichkeit ein solches System zu verstehen, ist die Schaffung von Modellen: Sie haben die wesentliche Aufgabe, ein System in abstrakter Form abzubilden [Krallmann ua 1999:29].⁶ [Gierhake 2000:15] grenzt den Modellbegriff noch stärker ein, indem er unter einem Modell die „zweckgerichtete Abbildung [...] der als wichtig erachteten Aspekte unter Vernachlässigung [...] weniger wichtig angesehener Gesichtspunkte“ versteht.

Dank der abstrakten Darstellung in einem Modell kann das originale⁷ System in vielen Fällen verstanden werden [ebenda].⁸ Und anstatt sofort ein System zu ändern, lassen sich Änderungen in dem Modell vornehmen und deren Auswirkungen messen. Die so gewonnenen Erkenntnisse fließen in die anschließende Systemänderung ein [Krallmann ua 1999:36].

In der Wirtschaftsinformatik werden Modelle für unterschiedliche Zwecke verwendet. Anders als die wesentlich feinere Kategorisierung in [Krallmann ua 1999:32] wird in [Gierhake 2000:16] von einer groben Einteilung in *Beschreibungsmodelle* und *Konstruktivistische Modelle*⁹ ausgegangen. Während den Beschreibungsmodellen Modelle von System- und Kommunikationsarchitekturen, Anwendungssystemen und Datenbanken zuzuordnen sind, handelt es sich bei Geschäftsprozess- und Wissensmodellen um Modelle zum Zweck der Erkenntnisgewinnung.

⁴Ein System zeichnet sich durch eine Menge von Elementen aus, die über Relationen miteinander verbunden sind [Krallmann ua 1999:21]. Über die Systemgrenze unterscheidet sich ein System von anderen, es umgebenden Systemen (die Umwelt) [Uni Saarland].

⁵Die Menschheit hat das System *Universum* bis heute nicht vollständig verstanden.

⁶Eine intensive Auseinandersetzung mit dem Begriff *Modell* in der Wirtschaftsinformatik führen zum Beispiel [Becker; Niehaves; Knackstedt 2004:3ff] und [Thomas 2005].

⁷Das Modell selbst ist auch wieder ein System.

⁸Das schon genannte System *Universum* kann trotz eines Modells nicht erklärt werden. Aber oft lassen sich Aussagen über das System auf ihre Richtigkeit hin überprüfen.

⁹Erkenntnisorientierte Modelle. [Gierhake 2000:16] bezeichnet sie auch als epistemologische Modelle.

Modelle werden einer *Modellierungsmethode* folgend entwickelt. Die Modellierungsmethode legt die Notation, bestehend aus den in dem Modell erlaubten Elementen und den Regeln zu ihrer Kombination, fest.¹⁰

Modellierungswerkzeuge helfen bei der Erstellung (Modellierung) von Modellen. Sie unterstützen in der Regel eine Modellierungsmethode und überwachen die Einhaltung dieser während der modellierenden Tätigkeit. Die auf dem Markt verfügbaren Modellierungswerkzeuge unterscheiden sich hauptsächlich in der Art, wie modelliert wird – entweder durch Eingabe von Text oder mittels Anordnung von grafischen Elementen.

Ist ein zu modellierendes System (i) hinreichend komplex und wird (ii) eine möglichst genaue Abbildung¹¹ des Systems angestrebt, dann ist die Aufteilung des Vorhabens in Teilaufgaben, deren Zuordnung zu einzelnen Personen und die daran anschließende parallele Bearbeitung sinnvoll. Zwei Vorgehensweisen können bei der multipersonellen Modellierung voneinander unterschieden werden: das *verteilte* und das *gemeinsame Modellieren*. Das verteilte Modellieren unterscheidet sich von dem gemeinsamen Modellieren dadurch, dass die beteiligten Personen nicht an einem (Abbildung 4), sondern an separaten aber miteinander verknüpften Modellen arbeiten, die Elemente gemeinsam nutzen – also Schnittmengen besitzen (Abbildung 3). Möchte der Betrachter eines *verteilten Modells* alle Aspekte einsehen, so muss er auf alle beteiligten Modelle zugreifen (Modelle A, B und C in Abbildung 3).

Für jedes der an einem verteilten Modellierungsprojekt beteiligten Modelle kann eine oder mehrere Personen verantwortlich zeichnen. Somit ist auch innerhalb der verteilten Modellierung ein gemeinsames Modellieren an den beteiligten Modellen denkbar.

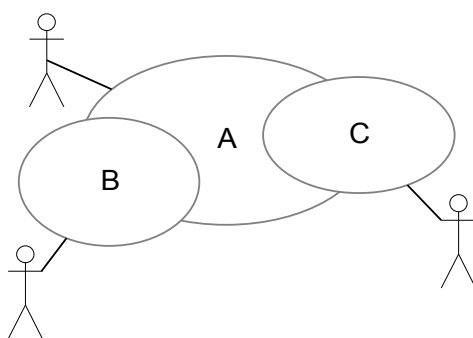


Abbildung 3. Verteiltes Modellieren

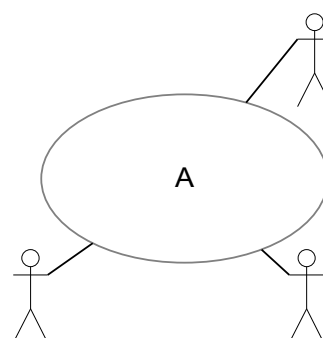


Abbildung 4. Gemeinsames Modellieren

Der Nutzen der verteilten Modellierung liegt in der parallelen Bearbeitung einer Aufgabe, was die *Effizienz* eines Projekts erhöht und bei der *Wiederverwendung* bereits modellierter Sachverhalte.

¹⁰ [Stahlknecht; Hasenkamp 2005:212],[Gierhake 2000:17]

¹¹ [Krallmann ua 1999:29] spricht von der Isomorphie (Strukturgleichheit) zwischen dem Modell und dem System.

Wie schon angeführt wurde, ist eine verteilte Modellierung dann angebracht, wenn das abzubildende System sehr komplex bzw. der zeitliche Rahmen für ein Modellierungsvorhaben stark begrenzt ist. Beispielhaft wird anhand von Ontologien und Geschäftsprozessmodellen gezeigt, dass das verteilte Modellieren sinnvoll sein kann.

(a) Ontologien. Werden in der Informatik Modelle mit dem Zweck der Darstellung eines Wissensbereichs erstellt und sollen diese anschließend von Software verwendet werden, wird häufig von *Ontologien* gesprochen [Gruber].¹² Ontologien sind stark formalisierte Modelle, die in einer aufsteigenden Reihe der Aussagekraft, beginnend mit der *Taxonomie*, über den *Thesaurus* und anschließend der *Topic Map*¹³, die höchste Stufe einnehmen [ontoprise 2003:2]. Während die Taxonomie lediglich eine Hierarchie von Begriffen ist,¹⁴ erweitert der Thesaurus diese um zwei Beziehungstypen – die Ähnlichkeits- und die Synonymbeziehung – die zwischen den Begriffen der Hierarchie eingesetzt werden können [ontoprise 2003:3f]. *Topic Maps* schließlich beinhalten wie die beiden vorher genannten Varianten Informationen *über* Informationen, ohne mit ihnen direkt verbunden zu sein. *Topics* in einer *Topic Map* sind Gegenstände des Interesses (Subjekte¹⁵), die elektronisch erfasst worden sind [empolis 2003:11].¹⁶ Die *Topics* haben Namen, weisen eine Ausprägung (*occurrences*) in Form von Informationsquellen (zB Dokumente, Webseiten) auf und haben rollenbasierte Beziehungen¹⁷ zu anderen *Topics* [empolis 2003:12],[Ontopia].

Ontologien unterscheiden sich von den vorgestellten Formen von Begriffssystemen, in dem hier – neben den Beziehungen vom Typ der Generalisierung (*ist-ein*) und der Abhängigkeit (*hat, besteht aus*) – die darin befindlichen Objekte Eigenschaftswerte erhalten können und dass für die Beziehungen untereinander Bedingungen definiert werden können [ontoprise 2003:6].¹⁸

Den Bedarf an Ontologien erkennen [Uschold; Gruninger 1996:8] bei der Kommunikation die zwischen Menschen, Organisationen und Softwaresystemen stattfindet: Da diese Subjekte eine unterschiedliche Sichtweise auf ein und denselben Gegenstand haben, aber oft miteinander kommunizieren müssen, wird ein Modell des gemeinsamen

¹² Definition nach von Foerster: „Ontologie ist [...] die Wissenschaft, die Theorie oder die Untersuchung des Seins, bzw. die Erforschung dessen, was ist, *wie es ist* usw“. Diese und weitere Definitionen des Begriffs Ontologie finden sich gut sortiert und in Beziehung zu anderen Begriffen in *Beats Bibliothek* unter <<http://beat.doebe.li/bibliothek/w00085.html>> (14.3.2006)

¹³ Topic Maps sind im ISO/IEC Standard 13250:2003 „Information technology – SGML applications – Topic Maps“ standardisiert. Verfügbar im Online-Katalog der ISO unter <<http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38068&ICS1=35&ICS2=240&ICS3=30>> (15.3.2006)

¹⁴ Diese Form der Generalisierung und Spezialisierung findet man häufig in der Biologie bei der Klassifizierung der Flora und Fauna [ontoprise 2003:3]

¹⁵ [empolis 2003:10]

¹⁶ In einem XML-Dialekt.

¹⁷ Die Beziehung Student zu Dozent weist die Rolle *ist Prüfling von* auf.

¹⁸ Ein Beispiel einer einfachen Ontologie befindet sich im Anschluss der Einleitung.

Verständnisses benötigt. Die *Künstliche Intelligenz* – ein Teilgebiet der Informatik, in dem Uschold und Gruninger 1996 wirkten – war für die Informatik Vorreiter bei der Erstellung und Nutzung von Ontologien. Dahinter stand die Idee, spezifiziertes Wissen in maschinenlesbarer Form zur Verarbeitung durch Software bereitzustellen.

Dieses Konzept übernahm einige Zeit später Tim Berners-Lee bei der gedanklichen Ausarbeitung des *Semantic Webs*, das er im Jahr 2001 als Erweiterung des bestehenden World Wide Webs formulierte [SemanticWeb]. Das erklärte Ziel des *Semantic Web* besteht darin, die derzeit noch unbeschriebenen Hypertext-Dokumente des Internets zu beschreiben und ihnen so Sinn zu verleihen.¹⁹ Informationen könnten damit, im Gegensatz zu den herkömmlichen Technologien, zielgerichteter aufgefunden werden. Menschen, die im Internet recherchieren, kennen das Problem der Suchmaschinen: Der Recherchierende erhält Fundstellen, die zwar das gesuchte Wort enthalten, aber das Wort wird in dem gefundenen Dokument in einem anderen als dem gewollten Zusammenhang verwendet. Denn es kann, das liegt in der Natur der menschlichen Sprache, synonym für verschiedene Dinge (Konzepte) benutzt werden.²⁰ Andersherum kommt es häufig vor, dass Dokumente in den Suchergebnissen nicht angezeigt werden, die uns aber interessieren könnten. In diesen nicht angezeigten Dokumenten kommt zwar das gesuchte Konzept vor, aber es ist dort anders bezeichnet (Homonyme).²¹ Fest steht, dass die heutigen Suchmaschinen oft unbefriedigende Ergebnismengen bereitstellen.

In der Vision des *Semantic Webs* sorgen Ontologien dafür, dass die *Konzepte* (siehe oben) in den Dokumenten wiedergefunden werden können: Werden in zwei Dokumenten unterschiedliche Begriffe im gleichen Kontext verwendet,²² ermöglichen Ontologien das maschinelle Erkennen der zugrundeliegenden identischen Bedeutung.²³ Die naheliegendste Anwendung von Ontologien ist also die Erweiterung der Funktionen von Suchmaschinen [McGuinness 2004:10].

Neben Maschinen, können auch Menschen Ontologien nutzen [Uschold; Gruninger 1996:7]. [McGuinness 2004:10] führt hierzu das Bilden von gemeinsamen Vokabularien und die Erschließung von Sinn an; [Uschold; Gruninger 1996:7] das Spezifizieren im Rahmen des *Systems Engineering*. Ersteres findet meist im Zusammenhang der Einführung eines Wissensmanagement in Unternehmen statt, bei dem das bei den Beteiligten (Wissensträger) schon vorhandene Wissen festgehalten werden soll.

Einen weiteren für Menschen erkennbaren Nutzen bringen Ontologien für das Surfen im Netz durch Verbinden von *tags*²⁴ [McGuinness 2004:10], mit denen Dokumente oder

¹⁹ Eine Software kann den Sinn von reinem Text in einem HTML-Dokuments nicht erfassen.

²⁰ Beispielsweise kann das Wort *Bank* als Sitzgelegenheit, Geldinstitut oder seichte Wasserstelle verstanden werden.

²¹ Der Suchbegriff ist *Bank*. Aber in dem gesuchten Hypertext-Dokument kommt nur *Geldinstitut* vor. Das dahinterstehende Konzept – ein Institut zur Aufbewahrung von Kapital – ist das gleiche.

²² Es steht das selbe Konzept hinter den Begriffen.

²³ Wenn in der Ontologie definiert ist, dass eine Bank ein Geldinstitut ist.

²⁴ *tag*: eine Auszeichnung bzw Kennzeichnung.

Daten ausgezeichnet wurden.²⁵ Außerdem lassen sich Prozesse fördern, die der gemeinsamen Festlegung über einen Sachverhalt dienlich sind.²⁶

Folgende Gründe sprechen für ein *verteilt*es Modellieren von Ontologien: (i) Können Ontologien sehr komplex werden, was ihre Modellierung durch eine einzelne Person erschwert, (ii) sollte eine Ontologie unter Beteiligung der späteren Nutzer modelliert werden, damit ihre Akzeptanz erhöht und ein möglichst hoher Grad an Übereinstimmung (*ontological commitment*) erreicht werden kann²⁷ und (iii) befinden sich die an dem Modellierungsvorhaben beteiligten Personen selten an einem Ort bzw arbeiten selten zur selben Zeit²⁸. Komplexe Produkte bestehen häufig aus Teilprodukten und analog dazu sind Verfeinerungen für bestimmte Teile der gemeinsamen Ontologie nötig. Ontologien können demnach aus Teilmodellen bestehen, die sich auf die übergeordnete Ontologie beziehen und Teile dieser detaillierter darstellen. Unterstellt man, dass Modell A in Abbildung 3 die gemeinsame Ontologie ist, dann greifen die Modelle B und C auf Inhalte der gemeinsamen Ontologie zu. Da die Teilmodelle (B) und (C) nicht am selben Ort wie die gemeinsame Ontologie (A) sein müssen, sondern (A) lediglich physikalisch erreichbar sein muss, entstehen *verteilte Modelle*.

(b) Geschäftsprozesse. Das verteilte Modellieren bietet sich auch für die Modellierung von *Geschäftsprozessen*²⁹ an: Wenn in einer kurzen Zeitspanne eine große Zahl von Geschäftsprozessen modelliert werden soll, kann ein verteiltes Vorgehen zweckmäßig sein.³⁰ Die Vorteile liegen zum Beispiel in der gemeinsamen Erstellung und Nutzung eines zentralen Vokabulars oder Organigramms des untersuchten Unternehmens (siehe hierzu 3.2.2 Eigene Untersuchungen).

Damit der mögliche Nutzen eintreten kann, muss die verteilte Modellierung von geeigneten Werkzeugen, einem gut ausgearbeiteten Vorgehen, einer wohl überlegten Organisationsstruktur sowie einer technischen Plattform untermauert werden [Vom Brocke

²⁵ Hochaktuell sind hier die Internetangebote von www.flickr.com, www.technorati.com oder www.yahoo.com. Wesentlich mehr Mächtigkeit bietet Beats Bibliothek
<<http://beat.doebe.li/bibliothek/index.html>>.

²⁶ [Gruber] spricht hier von dem „ontological commitment“.

²⁷ Die Ontologie ist ein von Menschen erzeugtes Modell und somit nicht objektiv sondern Resultat subjektiver Wahrnehmung.

²⁸ In einigen Fällen sind die beteiligten Modellierer nicht einmal Mitarbeiter desselben Unternehmens sondern in einem virtuellen Unternehmensverbund zusammengeschlossen.

²⁹ [Krallmann ua 1999:212], [Gierhake 2000:18], [Stahlknecht; Hasenkamp 2005:206] sind sich einig: ein Geschäftsprozess ist eine genau abgegrenzte, wiederholbare Abfolge von Aktivitäten, die einen Beitrag zur Wertschöpfung des Unternehmens leistet und an den (unternehmensinternen oder -externen) Kunden orientiert ist.

³⁰ In Ermangelung an Beispielen aus der Praxis sei auf zwei in den Jahren 2004 und 2005 verteilt durchgeführte Modellierungsprojekte der Wirtschaftsinformatik an der FH Brandenburg hingewiesen. Mehr dazu in [Simon 2004].

2003b:2].³¹ In erster Linie haben die Modelle in elektronischer Form vorzuliegen³² und sind für den Zugriff durch die beteiligten Modellierer zentral zur Verfügung zu stellen. Das eingesetzte Modellierungswerkzeug muss dazu geeignet sein, die Modelle miteinander zu verknüpfen, damit die Wiederverwendung von Elementen aus fremden Modellen möglich wird. Diese Forderung ordnet [Vom Brocke 2003b:3] dem *modellbezogenen Aspekt* der Gestaltungsanforderungen zu. Die Möglichkeiten, die SemTalk zur Kopplung von Modellen bereitstellt, werden in 2.1 *Verteilte Modellierung mit SemTalk* erläutert.

Hinter dem [Vom Brocke 2003b:3] angeführten *methodenbezogenen Aspekt* verbirgt sich das Vorgehensmodell, nach dem ein Projekt zur verteilten Modellierung durchzuführen ist. Mit dem definierten Vorgehen ist der Ablauf der durchzuführenden Aktivitäten formalisiert. Dabei sind Abläufe zu entwerfen, welche die Zielerreichung des Projekts sicherstellen und, da es sich bei der Modellierung um einen kreativen Vorgang handelt, gleichzeitig Freiheiten gewähren.

Für ein Projekt der verteilten Modellierung ist eine Organisationsform zu wählen, die, ähnlich dem methodenbezogenen Aspekt, die Erreichung des Projektziels dienlich ist. [Vom Brocke 2003b:2] bezeichnet diese Kategorie als den *organisationsbezogenen Aspekt*. Hier ist darauf zu achten, dass die kreativen Prozesse bei der Modellierung berücksichtigt werden und nicht in einem Korsett von Hierarchien der Zuständigkeiten und Genehmigungsverfahren unmöglich werden [Vom Brocke 2003b:2].

In [Krallmann ua 1999:26] werden typische Organisationsstrukturen verglichen, die in Abbildung 5 und Abbildung 6 wiedergegeben sind. Sie verdeutlichen die Unterschiede hinsichtlich der Komplexität bei der Team- sowie der Stablinienorganisation mit jeweils einem Vorgesetzten (grauer Punkt). Auf der einen Seite finden wir die Kommunikationswege in einem Team von miteinander arbeitenden Forschern und auf der anderen die Kommunikation zwischen einem Vorgesetzten und einer Anzahl von Sekretärinnen. Im ersten Fall, der sich mit der Organisation eines verteilten Modellierungsprojekts vergleichen lässt, kann ein wesentlich stärkerer Bedarf an Kommunikation zur Abstimmung festgestellt werden, als in dem zweiten Beispiel. Die zu wählende Organisationsform hat diesem Aspekt Rechnung zu tragen.

³¹ In [Vom Brocke 2003b] wird die *verteilte Referenzmodellierung* thematisiert. Trotzdem können die dort formulierten Gestaltungsanforderungen auf die verteilte Modellierung allgemein angewandt werden.

³² [Gierhake 2000:175] formuliert die „Eliminierung der papierbasierten Verarbeitung“ als Voraussetzung für die parallele Ausführung von Prozessen.

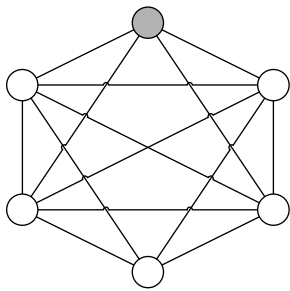


Abbildung 5. Team-Organisation

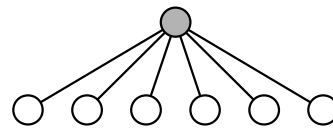


Abbildung 6.
Stablinienorganisation

Eine Auswahl an Vorgehensmodellen und Organisationsformen bei Projekten der verteilten Modellierung werden in 2.2 *Methoden und Organisation* vorgestellt.

Wesentlicher Bestandteil bei der verteilten Modellierung sind die „Austausch- und Diskursprozesse“, die zwischen den Modellierenden stattfinden. Diese sind, so [Vom Brocke 2003b:3], von einer zweckmäßigen technischen Plattform zu unterstützen. In 2.4 *Die Windows SharePoint Services* wird eine technische Plattform für die Zusammenarbeit vorgestellt. Inwieweit die WSS als Plattform für die verteilte Modellierung dienen können, ist Gegenstand der Untersuchung in dieser Arbeit.

2.1 Verteilte Modellierung mit SemTalk

SemTalk nutzt und erweitert Microsoft Office Visio, eine Software zur Visualisierung von Informationen,³³ um eigene Funktionen. Sollen mit SemTalk nur Modelle auf dem eigenen Rechner erstellt werden, kommt man ohne einen Server aus.

SemTalk unterstützt mehrere Modellierungsmethoden: Für die Wissensmodellierung steht die SemTalk-eigene Notation von Ontologien und für die Geschäftsprozessmodellierung stehen die Kommunikationsstrukturanalyse (KSA)³⁴, die PROMET-Methode³⁵ sowie die erweiterten Ereignis-gesteuerten Prozessketten (eEPK)³⁶ zur Verfügung.³⁷

³³ Mehr Informationen über das Produkt MS Office Visio können unter <http://www.microsoft.com/germany/office/visio/leistungsmerkmale/default.mspx> (3.4.2006) abgerufen werden.

³⁴ Die KSA ist in den späten 80er Jahren an der TU in Berlin von Prof Krallmann uA entwickelt worden [Semtation 2005:5].

³⁵ Prof Österle hat mit Unterstützung der IMG an der Schweizer Universität St. Gallen die PROMET-Methode entworfen. Mehr zu PROMET unter <http://www.semtalk.de/Promet.htm> (14.3.2006)

³⁶ Von August-Wilhelm Scheer entwickelte Modellierungsmethode, die von Petri-Netzen abgeleitet ist. Zu den Grundlagen von EPK mehr in G. Keller, M. Nüttgens, A.-W. Scheer. 1992. „Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK)“. In: Scheer, A.-W. (Hg.). Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes. Heft 89. <http://www.iwi.uni-sb.de/nuettgens/Veroeff/Artikel/heft089/heft089.pdf> (14.3.2006)

³⁷ Einen Überblick über alle unterstützten Modellierungsmethoden gibt die Auflistung der Anleitungen, die unter <http://www.semtalk.com/manuals.htm> (13.3.2006) verfügbar sind.

Außerdem lassen sich neue Modellierungsmethoden hinzufügen, die sich dann in SemTalk verwenden lassen. Zwei Faktoren machen dies möglich: (i) die komfortable Erzeugung von neuen Darstellungselementen in MS Visio³⁸ und (ii) das objektorientierte Konzept, auf dem SemTalk aufbaut.

Die Unterstützung einer Modellierungsmethode ist in SemTalk mittels der Interpretation eines Meta-Modells realisiert. Das Meta-Modell definiert die für die jeweilige Modellierungsmethode geltenden Regeln (Syntax und Semantik).³⁹ Soll ein Modell einer bestimmten Modellierungsmethode folgend erstellt werden, muss eine MS-Visio-Vorlage ausgewählt werden, die dann die für die Methode vorgesehenen Darstellungselemente und das dazugehörige Meta-Modell lädt.

Nach [Vom Brocke 2003b:3] sollen verteilte Modelle unabhängig voneinander sein, aber gleichzeitig gekoppelt werden können. Im Folgenden wird dargelegt, mit welchen Verfahren separate Modelle in SemTalk miteinander gekoppelt werden können. Es wird gezeigt, dass der modellbezogene Aspekt der Anforderungen an eine Umgebung zur verteilten Modellierung von dem Modellierungswerkzeug SemTalk abgedeckt wird.

Für die verteilte Modellierung mit SemTalk ist die objektorientierte Vorgehensweise von außerordentlicher Bedeutung. Das Paradigma der Objektorientierung geht davon aus, dass alle Dinge Objekte sind, die ein Verhalten (Methoden) aufweisen und Eigenschaften (Attribute) besitzen. Infolgedessen werden in SemTalk Klassen erzeugt, deren Objekte (Instanzen) in den Prozessmodellen verwendet werden. Die Klassen stellen die „begriffliche und semantische Basis der [...] Modelle“ her [Semtation 2005:26].⁴⁰ Somit wird die Konsistenz der im Modell verwendeten Begriffe erhalten. Das ist für umfangreiche Projekte mit mehreren Prozessen sinnvoll und nötig. Da SemTalk die Verwaltung der Verbindung zwischen Klassen und ihren Objekt-Instanzen übernimmt, werden außerdem umfangreiche Änderungen an Klassen effizient an alle davon instantiierten Objekte weitergegeben.

Sind (i) Modelle für andere Modelle physikalisch erreichbar, ist (ii) bei der Modellierung der Modelle objektorientiert vorgegangen worden und folgen (iii) die Modelle der selben Modellierungsmethode, dann lassen sich mit SemTalk verteilte Modelle herstellen. Bevor jedoch auf die Varianten verteilt zu Modellieren genauer eingegangen werden kann, müssen wichtige Begriffe erläutert werden.

(1) Betrachtetes Modell. Im Kontext dieser Arbeit handelt es sich bei dem betrachteten Modell um das SemTalk-Modell, das zur Zeit bearbeitet wird. Bei verteilten Modellen

³⁸ Dort als *Schablonen* (eng *shapes*) bezeichnet.

³⁹ Mehr zu Meta-Modellen und deren Modellierung in Dimitris Karagiannis und Harald Kühn. „Metamodelling Platforms“. In: Bauknecht, K; Min Tjoa, A; Quirchmayer, G (Hgg.). *Proceedings of the Third International Conference EC-Web 2002*. Berlin, Heidelberg: Springer. S. 182-196.
<http://www.dke.univie.ac.at/mmp/FullVersion_MMP_DexaECWeb2002.pdf> (10.3.2006)

⁴⁰ Da sich zwischen den Klassen Beziehungen (Vererbungen, Abhängigkeiten usw) herstellen lassen, kann in diesem Zusammenhang von *Ontologie-basierter Geschäftsprozessmodellierung* gesprochen werden.

ändert sich mit dem Standpunkt des Betrachters (= Modellierer) auch das betrachtete Modell.

(2) **Externes Modell.** Alle anderen außer dem betrachteten Modell sind externe Modelle. Wann ein Modell extern ist, hängt also vom Standpunkt der Betrachtung ab.

(3) **Externes Objekt.** Alle Objekte in einem externen Modell werden als externe Objekte bezeichnet.

(4) **Externe Referenz.** Zwischen einem Objekt eines betrachteten Modells und einem externen Objekt lässt sich ein Verweis – eine externe Referenz – herstellen. Auf die externe Referenz wird weiter unten genauer eingegangen.

(5) **Referenzobjekt.** Das von einem Objekt des betrachteten Modells mittels einer externen Referenz referenzierte externe Objekt wird als Referenzobjekt bezeichnet.

(6) **Verfeinerung.** Eine Verfeinerung ist ein Diagramm, welches ein einzelnes Objekt eines gegebenen Diagramms detaillierter darstellt. Es können zwei Arten von Verfeinerungen in SemTalk hergestellt werden: Die *interne Verfeinerung*, die sich im selben Modell befindet und die *externe Verfeinerung*. Letztere wurde für die verteilte Modellierung eingeführt und verbindet Objekte des betrachteten Modells mit Diagrammen externer Modelle. Die externe Verfeinerung wird im Folgenden genauer besprochen.

2.1.1 Verfahren zur verteilten Modellierung mit SemTalk

(1) **Die externe Verfeinerung.** Sind umfangreiche Sachverhalte zu abzubilden, bietet es sich an, erst auf einer abstrakten Ebene zu modellieren und anschließend die Details in einem separaten Modell wiederzugeben.⁴¹ Für Prozessmodelle lässt sich das mit der Verfeinerung umsetzen. In SemTalk wird üblicherweise der gesamte Geschäftsprozess auf einem hohen Level der Abstraktion mitsamt seiner Verfeinerungen in einer Datei gespeichert. Möglich macht das MS Visio, das erlaubt, mehrere Zeichenblätter in einer Datei abzuspeichern.

⁴¹Die Verfeinerung eines Prozesses ist eine Ausprägung des Top-Down-Ansatzes (vom Abstrakten hin zum Konkreten) [Krallmann ua 1999:36f].

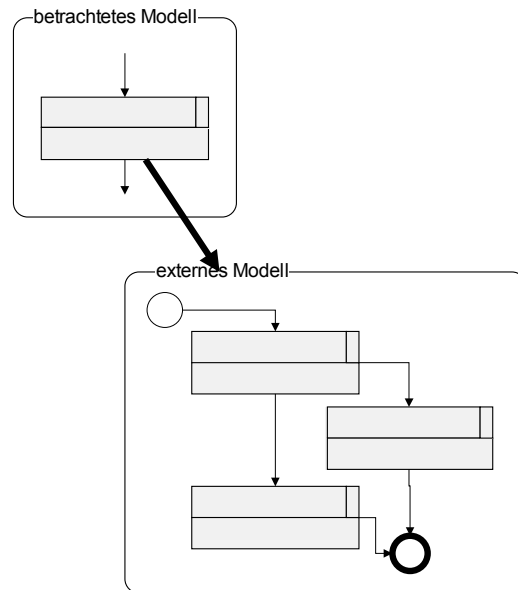


Abbildung 7. Externe Verfeinerung einer Aktivität in der KSA-Methode

Anders verhält es sich bei externen Verfeinerungen. Dabei wird eine Verknüpfung zwischen dem Prozesselement des betrachteten Modells und einem Diagramm eines externen Modells angelegt (siehe Abbildung 7). Die Information über die externe Verfeinerung ist in dem *übergeordneten* Modell gespeichert. Das externe Modell, welches in diesem Zusammenhang das verfeinernde Diagramm beinhaltet, hat keine Kenntnis davon, dass ein Objekt eines anderen Modells auf es verweist. Daher kann diese Art der Kopplung als lose bezeichnet werden und erfüllt so die Forderung in [Vom Brocke 2003b:3] nach Kopplungsfähigkeit bei gleichzeitiger Eigenständigkeit von verteilten Modellen.

Bei der externen Verfeinerung finden im Gegensatz zur internen Verfeinerung keinerlei Konsistenz-sichernde Maßnahmen statt: Wird in der KSA-Modellierungsmethode eine *interne Verfeinerung* angelegt, werden die dem verfeinerten Prozesselement direkt vor- und nachgelagerten Prozesselemente aus dem übergeordneten Diagramm in das verfeinernde Diagramm als Prozessstart bzw Prozessende übernommen. Ein solcher Mechanismus findet für die externe Verfeinerung nicht statt. Somit lassen sich auch SemTalk-fremde Dokumente als externe Verfeinerung an ein Objekt binden. Wie die Kopplung innerhalb des XML-Codes aussieht, zeigt Listing 1: Das Attribut *refinement* hat als Wert den Pfad zu dem externen Modell (*wertschöpfungskette* ohne die MS-Visio-Dateierweiterung .vsd) und dem verfeinernden Diagramm (*Einkauf*).

```
<Instance name="Ac#Domänenontologie erstellen.1493" ID="1493"
class="Ac#Domänenontologie erstellen"
refinement="file:///D:/wertschöpfungskette/Einkauf"
extrefapp="C:\Programme\Microsoft Office\Visio11\VISIO.EXE">
</Instance>
```

Listing 1

Die externe Verfeinerung erlaubt es, dass sich aus mehreren Modellen auf ein Diagramm bezogen wird. Einmal modellierte Geschäftsprozesse können so leicht wiederverwendet werden, was deren Gebrauchswert erhöht. Mit externen Verfeinerungen ließen sich auch in SemTalk modellierte Branchenmodelle an die jeweiligen (Prozess-)Ausprägungen in einem Unternehmen koppeln, indem die konkreten Prozessmodelle den Branchenmodellen als externe Verfeinerung zugeordnet werden.

In dem zweistufigen Dialog zur Herstellung einer externen Verfeinerung lässt sich nach der Auswahl der Modell-Datei ein darin befindliches Diagramm angeben. Der Betrachter kann ein verfeinertes Prozesselement an der Unterstreichung seines Objekt-namens erkennen. Die externe Verfeinerung kann über das Kontextmenü des verfeinerten Prozesselements erreicht werden.

(2) Die externe Referenz. Im SemTalk-Begriffsraum ist eine externe Referenz eine Verknüpfung zwischen zwei Objekten, wobei sich die Objekte in separaten SemTalk-Modellen befinden. Mit einer externen Referenz wird einem Objekt eines betrachteten Modells mitgeteilt, dass es sich um einen *Spezialfall* des referenzierten, externen Objekts handelt. Diese Maßnahme dient der Herstellung von Konsistenz über eine Menge von mehreren Modellen: Objekte die den gleichen Namen besitzen, aber in separaten Modellen auftauchen, sind auf ein gemeinsames Konzept hin zu prüfen. Damit wird die parallele Modellierung gleicher Dinge vermieden und die Wiederverwendung gefördert.⁴² Ist ein Konzept mehrfach vorhanden, muss entschieden werden, wer das originale Konzept (Objekt) pflegt und wessen Objekt sich auf dieses Original zu beziehen hat. Nach der Herstellung der externen Referenz gleicht sich das referenzierende Objekt mit allen Methoden, Attributen und Zuständen des Referenzobjekts ab.

Zu den replizierten Methoden, Attributen und Zuständen lassen sich in das referenzierende Objekt eigene Methoden, Attribute und Zustände einfügen. Diese werden nicht von dem Abgleich mit dem referenzierten Objekt beeinflusst und können als *lokale* Methoden, Attribute und Zustände bezeichnet werden.⁴³ In dem nachfolgenden Vergleich wird der Unterschied zwischen replizierten und lokalen Aspekten deutlich: Neben der replizierten Methode *verzinken*, hat das referenzierende Objekt die Attribute *verzinkt* und *ge-*

⁴² Anwendungsbeispiel: Nutzung von Referenzmodellen für die Erstellung konkreter Anwendungsmodelle.

⁴³ Von dem Referenzobjekt replizierte Aspekte stellen eine Teilmenge der Aspekte des referenzierenden Objekts dar.

schweißt von dem Referenzobjekt übernommen. Daneben besitzt es eine eigene Methode *lackieren* und das Attribut *lackiert*.

Referenzobjekt

Karosserie
verzinkt
geschweißt
verzinken

Referenzierendes Objekt

Karosserie
verzinkt
geschweißt
lackiert
verzinken
lackieren

Die Funktion zum Anlegen der externen Referenz ist über das Kontextmenü jedes Objekts in SemTalk möglich. Das externe Modell mit dem Referenzobjekt muss allerdings im XML-Format⁴⁴ vorliegen, das SemTalk standardmäßig neben der MS-Visio-Datei speichert. Ist das Modell lediglich im proprietären Visio-Dateiformat verfügbar, kann es nicht – ohne von MS Visio geladen zu sein – von außen gelesen oder beschrieben werden.

Bei der ausgewählten XML-Datei muss es sich um eine gültige Repräsentation eines SemTalk-Modells handeln und es muss ein Objekt in dem externen Modell vorhanden sein, das die gleiche Bezeichnung hat, wie das Objekt des betrachteten Modells. Diese Prüfungen sind Bedingung für eine Konsistenz zwischen den nun miteinander verbundenen Objekten und entspricht der Philosophie des *semantic webs*, wonach sich gleiche Konzepte⁴⁵ über Namen wiederfinden.⁴⁶

Bei dem Abgleich⁴⁷ zwischen dem Referenzobjekt und dem referenzierenden Objekt werden zunächst der Name, die Methoden, Attribute und Zustände des Referenzobjekts ermittelt. Anschließend werden die ermittelten Aspekte mit denen des referenzierenden Objekts verglichen. Stellt SemTalk dabei eine Ungleichheit fest, wird das referenzierende Objekt aktualisiert: Der Name wird an den möglicherweise geänderten Namen des Referenzobjekts angepasst sowie die fehlenden Methoden, Attribute und Zustände hinzugefügt.

Das Wissen über eine externe Referenz besteht nur auf der Seite des referenzierenden Objekts. Das hat den Vorteil der Sparsamkeit hinsichtlich der zu speichernden Informationen. Aber gleichzeitig folgt daraus ein elementares Problem: Wird das Referenzobjekt oder dessen Modell *gelöscht*, so besteht weiterhin die externe Referenz in dem referenzierenden Objekt. Das Referenzobjekt hat schließlich keine Kenntnis von der externen Referenz und kann somit das referenzierende Objekt (bzw das beinhaltende Modell) von dem neuen Sachverhalt *nicht* informieren. Die externe Referenz entspricht dann einem nicht-aktuellen Zustand, da das Referenzobjekt physikalisch nicht mehr erreichbar

⁴⁴ Weiterführendes zu der Auszeichnungssprache XML in den Teilen 3.3 und 3.5.2

⁴⁵ Entspricht den Objekten in einem SemTalk-Modell.

⁴⁶ „Identifying Things on the Semantic Web.“ <<http://www.w3.org/2001/03/identification-problem/si.html>> (14.1.2006)

⁴⁷ SemTalk-Funktion *Aktualisieren*

ist. Die Folge ist die Inkonsistenz des referenzierenden Modells: Die ursprüngliche Festlegung, dass es sich bei dem lokalen, referenzierenden Objekt um eine Spezialisierung des Referenzobjekts handelt, ist weiterhin im Modell des referenzierenden Objekts definiert, aber nicht mehr gültig.

Ähnlich verhielt es sich, wenn das Referenzobjekt lediglich *umbenannt* wurde: Da das Referenzobjekt in dem externen Modell über den Namen gesucht wurde (siehe oben), konnte SemTalk es nach dessen Umbenennung nicht wiederfinden.⁴⁸ Da es sich aber weiterhin um das gleiche Konzept handelt, das lediglich eine neue Bezeichnung erhielt und darum nicht mehr wiedergefunden werden konnte, war eine Lösung zu finden, die der Idee des *semantic webs* widerspricht⁴⁹ – die Objekt-ID wurde eingeführt. Mit ihr sind Objekte eines Modells eindeutig identifizierbar. Ein Objekt, das sich nun auf ein externes Objekt mittels einer externen Referenz bezieht, besitzt in seinen Eigenschaften die Objekt-ID des Referenzobjekts.⁵⁰ Eine Umbenennung des Referenzobjekts führt seit dem nicht mehr zu inkonsistenten externen Referenzen. Bei der Aktualisierung des referenzierenden Objekts wird nun auch der Name des Referenzobjekts übernommen. Im Listing 2 ist die Objekt-ID des Referenzobjekts im Attribut *SrcID* des referenzierenden Objekts zu finden. Das *Location*-Element beinhaltet den Pfad zu dem externen Modell.

```
<Class name="Objekt" ID="1550" SrcID="1552">
  <Location name="file:///Z:/externesModell.xml" />
</Class>
```

Listing 2

SemTalk stellt Funktionen bereit, die dabei helfen, Objekte externer Modelle *wiederverwenden*. Das Resultat sind immer Objekte, die mit einer externen Referenz versehen sind. Nachfolgend die Funktionen:

(a) Anzeige externer Modelle und Import externer Objekte. Ist SemTalk geöffnet, können mit der *Anzeige*-Funktion Objekte eines externen Modells angezeigt und, sofern gewünscht, einzeln in das aktuell geöffnete Modell importiert werden. Jedes auf diese Weise importierte Objekt besitzt automatisch eine externe Referenz auf das angezeigte externe Modell.

⁴⁸ SemTalk sieht – richtiger Weise – nicht vor, in diesem Augenblick die externe Referenz automatisch zu entfernen: Ein Objekt eines auf dem lokalen PC gespeicherten Modells referenziert ein Objekt eines im Internet erreichbaren Modells. Kann SemTalk das Referenzobjekt auf Grund einer eingeschränkten Internetverbindung nicht erreichen, ist die externe Referenz nicht sofort zu entfernen.

⁴⁹ Häufig tritt dieses Problem bei der (verteilten) Modellierung von Geschäftsprozessen auf. Oftmals wird etwas konzeptionell identisches in zwei Modellen unterschiedlich benannt (Synonym).

⁵⁰ Die sogenannte *SrcId* (*Source-ID*).

(b) **Der SemTalk-Import.** Diese Funktion erlaubt dem Benutzer ein beschleunigtes Vorgehen, was aber ansonsten dem Vorgehen zur Anzeige und Import einzelner Objekte ähnelt. Der Unterschied liegt im Grad des Komforts: Zunächst kann sich der Modellierer das komplette externe Modell anzeigen lassen. In diesem Dialog ließe sich das gesamte externe Modell (oder eines der darin befindlichen Diagramme mit allen darin befindlichen Objekten) in einem Vorgang in das eigene Modell importieren.

(3) **Die externe Verfeinerung von.** Dem Diagramm eines SemTalk-Modells kann mitgeteilt werden, dass es sich um die Verfeinerung eines oder mehrerer externer Objekte⁵¹ handelt. Es ist dann die *externe Verfeinerung von* dem Prozesselement *x*. Abbildung 8 verdeutlicht den Sachverhalt an einem Beispiel: Auf der höchsten Unternehmensebene von McDonalds wurde die global gültige *McDonalds Wertschöpfungskette* modelliert. Dieses Modell enthält ein Diagramm *Einkauf*, das den weltweit standardisierten Geschäftsprozess des Einkaufs von Rohstoffen abbildet. Da dieser als unternehmensinterner Standardprozess genutzt werden soll, ist er den Prozesselementen *Einkauf* der lokalen Wertschöpfungsmodelle (zB *McDonalds Wertschöpfung USA*) als externe Verfeinerung zuzuordnen.

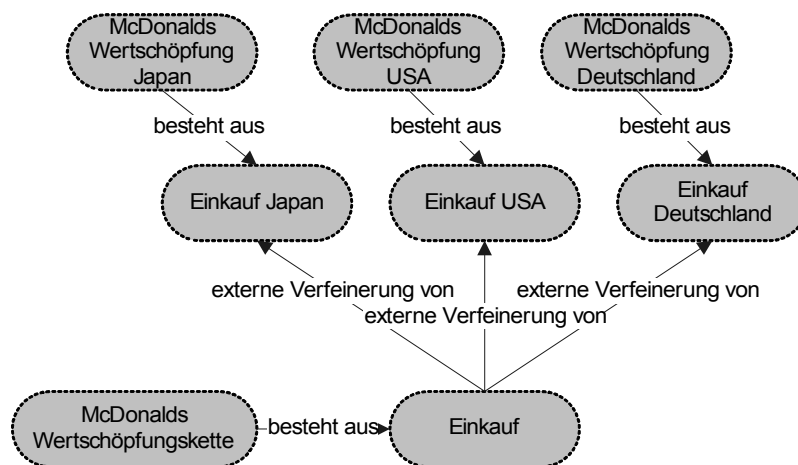


Abbildung 8. Der globale Geschäftsprozess Einkauf als externe Verfeinerung der lokalen Prozesse

Von den Prozesselementen (zB *Einkauf Deutschland*), denen von Außen eine externe Verfeinerung zugewiesen wurde, kann zu der externen Verfeinerung navigiert werden. So kann mit diesem Mechanismus die Wiederverwendung von bereits modellierten Teilprozessen gewährleistet werden.

Die Zuweisung eines Prozessdiagramms zu mehreren Prozessobjekten geschieht in den Eigenschaften eines Diagramms. Auf der Registerkarte *Ext Refinement Of* wird erst ein externes Modell geöffnet und anschließend ein darin befindliches Prozessobjekt ausgewählt. Listing 3 zeigt, wie diese Zuordnung innerhalb des Codes der Modell-Datei

⁵¹ Prozesselemente externer Modelle.

realisiert ist: Das Diagramm *Rohstoff einkaufen* ist den zwei externen Prozesselementen *Rohstoff einkaufen* der Modelle *Einkauf.Deutschland* sowie *Einkauf.Japan* zugewiesen.

```
<Diagram name="Rohstoff einkaufen" class="Process" id="4"
type="Instance" stencil="bonproc.vss" root="ProcessElement">
  <InvRefinement
    name="D:\Einkauf.Deutschland.xml#1493"
    ID="1493"
    file="D:\Einkauf.Deutschland.xml"
    title="1 Rohstoff einkaufen
(D:\Einkauf.Deutschland.xml)" />
  <InvRefinement
    name="D:\Einkauf.Japan.xml#2401"
    ID="2401"
    file="D:\Einkauf.Japan.xml"
    title="1 Rohstoff einkaufen
(D:\Einkauf.Japan.xml)" />
</Diagram>
```

Listing 3

Ist ein Diagramm als eine externe Verfeinerung definiert worden, dann erscheint in dem externen Modell des *verfeinerten* Prozesselements ein Hinweis auf die externe Verfeinerung (Abbildung 9). Dieser als *SmartTag* realisierte Knopf macht das Navigieren zu dem externen Modell möglich, das die externe Verfeinerung beinhaltet.

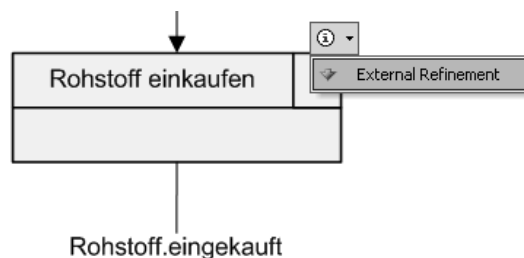


Abbildung 9. Hinweis auf eine externe Verfeinerung

(4) Der SemTalk-Assistent Merlin und der Modell-Index. Soll über eine Menge von SemTalk-Modellen begriffliche Konsistenz erreicht werden,⁵² bietet sich die Verwendung des SemTalk-Assistenten *Merlin* an. Er kann Hinweise auf (i) die andernorts vorliegende Verwendung von Klassennamen, (ii) die synonyme sowie homonyme Verwendung eines Klassennamens in dem betrachteten Modell sowie (iii) die bereits bestehende Verwendung⁵³ eines gerade eingegebenen Objektnames geben. Der Modellierer kann entscheiden, wie er auf den Hinweis reagieren möchte. Merlin schlägt ihm hierfür eine von

⁵² Objekte, meist Informationsklassen, können in mehreren Modellen vorkommen aber nur an einer Stelle wird das genuine Objekt gepflegt.

⁵³ Dabei kann es sich um den Namen einer in dem selben Modell bereits bestehenden Klasse, einer Methode, eines Attributs oder eines Zustandes handeln.

dem Ereignis abhängige Aktion vor. Beispielhaft sei hier die Möglichkeit erwähnt, von Merlin automatisch eine externe Referenz erstellen zu lassen, wenn der Modellierer ein Objekt nach einem externen Objekt benannt hat. In Abbildung 10 ist dieser Fall beispielhaft dargestellt: Eine gerade neu erzeugte Informationsklasse wurde *Baum* genannt. Der SemTalk-Assistent Merlin stellt fest, dass es bereits ein externes Objekt mit der Bezeichnung *Baum* in einem anderen Modell gibt. Auf die daraufhin mögliche Aktion eine externe Referenz zu dem externen Objekt herzustellen, wurde bereits eingegangen.

Die oftmals als unliebsam empfundenen MS-Office-Assistenten⁵⁴ werden von den Benutzern häufig ausgeblendet.⁵⁵ Für die adäquate Unterstützung des Modellierers ist die Anzeige des SemTalk-Assistenten jedoch nötig.

Merlin greift für seine unterstützende Arbeit auf den *Modell-Index* zu. Dieser wird von Merlin gepflegt und beinhaltet Informationen über alle Objekte, die sich in der Menge der zu indizierenden Modelle befinden. Die Voraussetzung für einen möglichst umfangreichen Modell-Index ist die Registrierung aller erreichbaren Modelle.

In dem Modell-Index werden die Namen aller Objekte zusammen mit dem Namen des Modells, das sie beinhaltet, ihrem Objekt-Typ⁵⁶, ihrer System-Klasse⁵⁷ sowie ihrem Namensraum gespeichert. Da in dem Begriffsraum von SemTalk alle Elemente eines Modells Objekte sind, werden neben Klassen und Instanzen auch Diagramme mit indiziert.

Die derzeitige Unterstützung der verteilten Modellierung erlaubt es, über eine einfache Einstellung die Modell-Indexdatei (*modelindex.xml*) auf einen anderen, als den standardmäßig eingestellten Speicherort zu legen. Dabei könnte es sich beispielsweise um ein freigegebenes Verzeichnis in einem lokalen Netzwerk handeln. Wäre ein Modellierungsprojekt derart geplant, dass die beteiligten Modellierer ihre Modelle auf dem freigegebenen Netzwerkverzeichnis zu speichern haben, dann könnten die Modellierer ihre SemTalk-Anwendung so konfigurieren, dass diese sich auf den dort ebenfalls abgelegten zentralen Modell-Index beziehen. Damit wäre es dem einzelnen Beteiligten möglich, sich über Objekte in den Modellen der anderen Personen informiert zu lassen und gegebenenfalls Referenzen herzustellen. Ein in einem separaten Dokument festgehaltenes Glossar

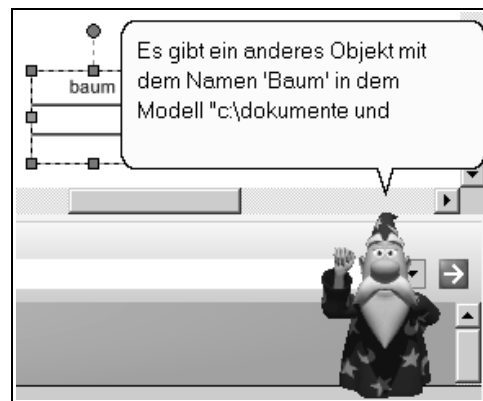


Abbildung 10. Merlin mit Hinweis über ein vorhandenes externes Objekt.

⁵⁴ Bei Merlin handelt es sich um die Verwendung und Erweiterung eines MS Office-Assistenten.

⁵⁵ Auszüge im Internet: <<http://lustich.de/lustich/bilderdb-bilder-4-7.html>>, <<http://www.heise.de/newsticker/meldung/17028>> und <http://www.vnr.de/vnr/computeredv/office/praxistipp_15110.html>

⁵⁶ Objekte können zB Klassen, Instanzen, Prozess- oder Informationsmodelle sein.

⁵⁷ Die in einer Modellierungsmethode möglichen System-Klassen entstammen dem der Methode zugrundeliegendem Metamodell.

aller Objekte in den vorhandenen Modellen wäre nicht nötig und es könnte sich ein Netz von externen Referenzen zwischen den Modellen aller Beteiligten ergeben. Während der Erstellung dieser Arbeit betreute der Autor eine Studiengruppe bei einem derartig realisierten Modellierungsprojekt und konnte dabei die Effektivität dieses Vorgehens feststellen.⁵⁸

2.1.2 Exkurs: Referenzmodelle – Modellierung und Nutzung mit SemTalk

Mit den hier vorgestellten Möglichkeiten von SemTalk, verteilt zu modellieren, ließen sich mit SemTalk Referenzmodelle erstellen und in Anwendungsmodellen nutzen. Unter Referenzmodellen verstehen [Becker; Niehaves; Knackstedt 2004:1] „[...] Modelle, welche in anderen Anwendungskontexten Verwendung finden sollen“ und vom Brocke Modelle, „deren Inhalte in Konstruktionsprozessen anderer [...] Modelle wieder verwendet werden“ [Vom Brocke 2003a:34],[Vom Brocke 2003b:1],[Vom Brocke; Buddendick 2004:19]. [Becker; Niehaves; Knackstedt 2004:1] definieren *Verwendung* als Referenzierung auf die Inhalte von Referenzmodellen aus Anwendungsmodellen heraus. Beispiele für Referenzmodelle finden sich zum Beispiel in Vorgehensmodellen,⁵⁹ die ein Vorgehen standardisiert beschreiben, und in Branchenmodellen⁶⁰. In jedem Fall wird von der konkreten Ausprägung des Geschäftsprozesses – ob Vorgehens- oder Branchenmodell – abstrahiert und ein allgemeingültiges Modell erstellt. Damit kann in Hinblick auf die Reorganisation von Geschäftsprozessen „die Erarbeitung der Soll-Abläufe beschleunigt“ werden [Krallmann ua 1999:228].

Für die Modellierung und Nutzung von Referenzmodellen halten [Vom Brocke; Buddendick 2004:22] neben der *Konfiguration* die Techniken der *Instantiierung*⁶¹, der *Spezialisierung*⁶², *Aggregation*⁶³ sowie *Analogie*⁶⁴ für notwendig. Im Folgenden wird gezeigt, wie mit den in SemTalk vorhandenen Funktionen und Mechanismen die in [Vom Brocke; Buddendick 2004] eingeführten Konstruktionstechniken angewendet werden können.

(1) Instantiierung eines Referenzmodells. Ein Versandhaus entschließt sich, seine Geschäftsprozesse zu optimieren. Dazu wird die Bearbeitung des Posteingangs unternehmensweit einheitlich in einem Prozessmodell mit SemTalk abgebildet. Dies ist das nun verfügbare Referenzmodell *Posteingang*. Jede Abteilung hat ihre Geschäftsprozesse

⁵⁸ Lehrveranstaltung „Design für Kooperation“ im Schwerpunkt Kooperative Systeme des Studiengangs Wirtschaftsinformatik an der FH Brandenburg im Jahreswechsel 2005/2006.

⁵⁹ Das Vorgehen bei dem Qualitätsmanagement-Audit ist beispielsweise standardisiert.

⁶⁰ In diesen werden die Geschäftsprozesse abgebildet, die für eine Branche typisch sind.

⁶¹ Modellierung von Sachverhalten, die von mehreren Anwendungsmodellen genutzt werden.

⁶² Übernahme des kompletten Inhalts eines Referenzmodells in ein Anwendungsmodell und die Möglichkeit die übernommenen Inhalte zu erweitern.

⁶³ Import von Inhalten mehrerer externer Modelle in ein Anwendungsmodell und Herstellung von Beziehungen zwischen den importierten Inhalten.

⁶⁴ Nutzung von Diagrammen externer Modelle mittels der *externen Verfeinerung*. Da hier nur Ähnlichkeit der Inhalte gefordert ist, dürfen keine Konsistenz überprüfende Maßnahmen stattfinden.

selbst mit SemTalk zu modellieren. Mindestens alle Abteilungen, die Kundenkontakt haben, müssen in ihrem Modell einen Prozess *Posteingang bearbeiten* angelegt haben. Das Referenzmodell mit dem Prozess *Posteingang* kann nun jedem Prozess *Posteingang bearbeiten* in den Prozessmodellen der Abteilungen mit Hilfe der SemTalk-Technik *externen Verfeinerung von* zugewiesen werden. Die Nutzer der Abteilungsmodelle können von dem Prozess *Posteingang bearbeiten* zu dem Referenzprozess *Posteingang* navigieren und sich während der täglichen Arbeit an dem dort abgebildeten Prozess orientieren.⁶⁵

(2) Analogiekonstruktion. Bleibt man bei dem oben genannten Beispiel, so könnten die Modellierer der Abteilungsmodelle das global verfügbare Referenzmodell *Posteingang* mit der *externen Verfeinerung* in ihre Geschäftsprozessmodelle einbinden. Ähnlich wie bei *Instantiierung* kann bei der *Analogiekonstruktion* der Referenzprozess geändert werden, ohne die Abteilungsmodelle aktualisieren zu müssen.

(3) Spezialisierung und Aggregation. Diese beiden Varianten der Nutzung von Referenzmodellen können mit der Import-Funktion und den dabei von SemTalk automatisch angelegten externen Referenzen realisiert werden. Neben dem Referenzmodell *Posteingang* könnten auch Teile eines Referenzmodells *Reklamationsgespräch* in die Modelle der Abteilungen des Versandhauses importiert werden. Die importierten Prozesse und Modellobjekte können in einem neuen Geschäftsprozessmodell kombiniert werden (= Aggregation). Da den importierten Prozessobjekten Klassen gegenüberstehen, wurden diese mit in die Abteilungsmodelle importiert. Die Klassen können nun, ohne dabei das Referenzmodell zu beeinträchtigen, um eigene Methoden, Attribute und Zustände erweitert werden (= Spezialisierung).

Entschließt sich ein Unternehmen, mit dem Ziel Schwachstellen aufzudecken und Prozesse zu verbessern dazu, ein Branchenmodell einzuführen, lassen sich die oben ausgeführten Konstruktionstechniken der Referenz- und Anwendungsmodellierung mit SemTalk nutzen. So können die in der Praxis immer auftretenden Abweichungen zwischen dem Referenzmodell und den konkreten Prozessen im Unternehmen abgebildet werden, ohne dass das Nutzenpotential des Referenzmodells abnimmt.⁶⁶

Die hier gezeigten Anwendungsbeispiele zur Referenzmodellierung mit SemTalk ließen sich dank der Verknüpfungen mit externen Modellen auch verteilt durchführen. Während die Einführung eines Referenzmodells immer eine strategische Entscheidung der Unternehmensleitung ist, kann deren Nutzung und Erweiterung an die Fachabtei-

⁶⁵ Hat das Versandhaus vor, ein Qualitätsmanagement aufzubauen, sind standardisierte Vorgänge für alle Mitarbeiter bereitzustellen.

⁶⁶ Oft gibt es branchentypische Prozesse, die in ihrer konkreten Ausprägung im Unternehmen die Alleinstellungsmerkmale des Unternehmens ausmachen. Deren Beseitigung hätte negative Auswirkungen auf die Wettbewerbsfähigkeit des Unternehmens.

lungen delegiert werden. Die Referenzmodellierung könnte so effizient und auf mehrere Personen verteilt durchgeführt werden. Dies macht wiederum den Einsatz einer Umgebung zur verteilten Modellierung notwendig.

2.1.3 Exkurs: Zyklisch referenzierte Objekte und verteilte Datenbanken

Im Zuge der Auseinandersetzung mit der Referenzierung zwischen zwei Objekten separater Modelle (externe Referenz) ist Folgendes aufgefallen: Bei herkömmlich referenzierenden Objekten handelt es sich um Objekte, welche mit Hilfe von externen Referenzen Objekte aus externen Modellen referenzieren und sich mit diesen hinsichtlich der Objektaspekte⁶⁷ abgleichen. Ein zyklisch gerichteter Graph entsteht dann, wenn das soeben zum Referenzobjekt gemachte Objekt ein Objekt in einem dritten Modell referenziert – also selbst zu einem referenzierenden Objekt wird – und das neue Referenzobjekt aus dem dritten Modell das ursprüngliche Objekt in dem ersten Modell referenziert. In wird der Sachverhalt grafisch verdeutlicht. Dabei existiert die Informationsklasse *Automobil* in drei separaten Modellen, die in einem Zyklus referenziert wird.

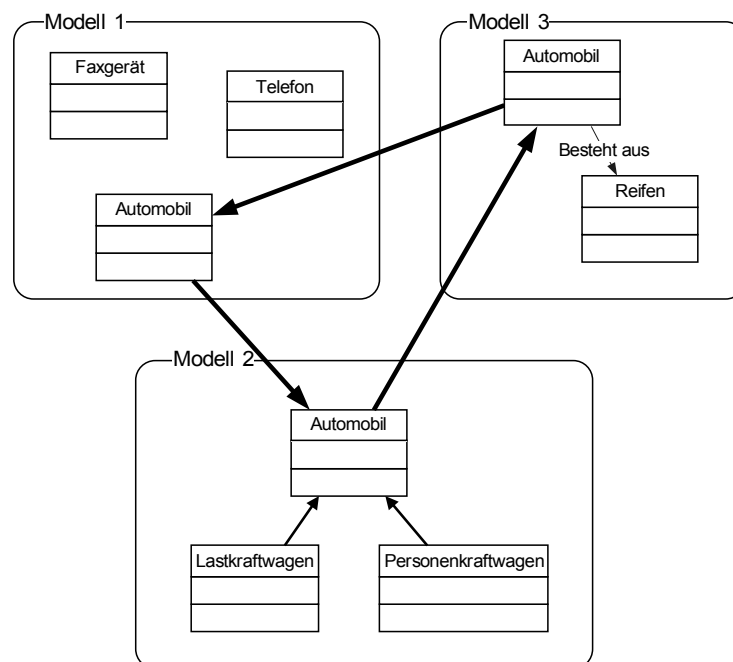


Abbildung 11. Zyklisch gerichteter Graph von externen Referenzen in SemTalk

Mit der Herstellung dieses zyklisch gerichteten Graphen von externen Referenzen ließe sich gemeinsam an einem Objekt – es kann nun von einem *zyklisch referenzierten Objekt* gesprochen werden – in separaten Modellen arbeiten. Bei einer regelmäßigen Aktualisierung würden Ergänzungen an dem zyklisch referenzierten Objekt, gleich in welchem Modell vorgenommen, auf alle beteiligten Modelle übertragen werden.

⁶⁷ Objektname, Methoden, Attribute und Zustände.

Stelle man sich vor, dass anstelle des zyklisch referenzierten Objekts eine Datenbank-Tabelle stünde und sich anstatt der beteiligten Modelle separate Datenbanken befänden, könnte über die zyklische Referenzierung und den damit möglichen Aktualisierungen eine verteilte Datenbank aufgebaut werden. Dieses Prinzip liegt den verteilten Datenbanken in heutigen Peer-2-Peer-Netzwerken zugrunde (*distributed hashtables*, DHT).⁶⁸ Darüberhinaus gibt es Bestrebungen, das Domain Name System⁶⁹ (DNS) dezentral und verteilt bereitzustellen.⁷⁰

In der aktuellen Version von SemTalk wird lediglich das Ergänzen von referenzierten Objektaspekten unterstützt. Sind Aspekte des Referenzobjekts nach Herstellung der externen Referenz entfernt oder umbenannt worden, werden diese Änderungen bei der Aktualisierung des referenzierenden Objekts nicht berücksichtigt. Der Grund dafür ist, dass SemTalk die einmal von einem Referenzobjekt replizierten Aspekte nicht als solche markiert. Würde (i) die Funktion des Markierens und Überprüfens replizierter Objektaspekte hinzukommen, wären (ii) alle beteiligten Modelle physikalisch erreichbar und würden (iii) die nötigen Aktualisierungen der referenzierenden Objekte automatisiert vorgenommen, so wäre ein annähernd konsistenter Zustand eines zyklisch referenzierten Objekts in allen Modellen zu einem bestimmten Zeitpunkt gewährleistet.

Derzeit füllt sich das zyklisch referenzierte Objekt lediglich mit den aus dem Referenzobjekt ermittelten Objektaspekten auf, ohne dass die für die Konsistenz nötige Wartung durchgeführt wird. Auch müssen noch Fälle berücksichtigt werden, in denen eine Unterbrechung des Zyklus vorliegt.⁷¹ Würden jedoch die zyklisch referenzierten Objekte korrekt gewartet, bestünden Auswirkungen hinsichtlich der Prozessmodellierung, wenn Prozesselemente die von Referenzobjekten replizierten Methoden verwenden.⁷²

2.2 Methoden und Organisation

Der in [Vom Brocke 2003:3] eingeführte methodenbezogene Aspekt betont den Bedarf an einem konkreten Vorgehen⁷³ und einem Regelwerk für die Modellierung unter gleichzeitiger Wahrung der persönlichen Ausdrucksmöglichkeiten. Ähnliches trifft auf den organisationsbezogenen Aspekt zu: Gegenstand sind hier die zu regelnden Abstimmungsprozesse und die Organisationsstruktur. Beide müssen so gestaltet sein, dass die Ziele des

⁶⁸ Mehr dazu unter <http://de.wikipedia.org/wiki/Distributed_Hashtable> (28.4.2006).

⁶⁹ Verwaltung des Namensraums im Internet. DNS übersetzt die öffentlichen IP-Adressen in Domännennamen. Zum Beispiel: IP-Adresse 217.12.3.11 zu www.yahoo.de.

⁷⁰ Das Projekt *Distributed DNS* ist unter <<http://distributeddns.sourceforge.net>> (28.4.2006) erreichbar.

⁷¹ Ein Modell ist entfernt oder nicht erreichbar. Auch möglich: Das zyklisch referenzierte Objekt ist aus einem beteiligten Modell entfernt.

⁷² Eine Methode des Referenzobjekts wird umbenannt und bei dem referenzierenden Objekt aktualisiert. Prozesselemente, die diese Methode verwenden würden automatisch umbenannt werden, was unter Umständen von dem Modellierer nicht gewollt ist.

⁷³ „zeitlich-sachlogische Abfolge von Funktionen im Konstruktionsprozess“.

Modellierungsvorhabens erreicht, aber die Beteiligten nicht in ein „Korsett“ von Regularien eingengt werden. Wie dieser Spagat zwischen Regeln einerseits und dem Erhalt der für die Modellierung nötigen Flexibilität andererseits in Modellierungsvorhaben gemeistert wurde, soll im Folgenden anhand von dokumentierten Vorgehensmodellen dargestellt werden. Diese sind aus der Praxis heraus entstanden und weitestgehend evaluiert.

Abschließend werden die als ähnlich angesehenen methodischen Schritte zusammengefasst. Die für diese Schritte maßgeblichen Aufgaben, bestimmen die Anforderungen an eine Plattform für die verteilte Modellierung.

2.2.1 Vorgehensmodelle nach Sure

Für diesen Teil ist es hilfreich, die im Anhang I befindlichen Prozessmodelle parallel zu betrachten. Das hier beschriebene Vorgehen ist dort in der KSA-Methode modelliert.

Die im Zusammenhang mit der Erstellung des „SEmantic portAL“ (SEAL) für das Institut AIFB der Universität Karlsruhe unter Verwendung des Modellierungswerkzeugs OntoEdit gemachten Erfahrungen werden in [Sure ua 2002] wiedergegeben.⁷⁴ Das Hauptaugenmerk in [Sure ua 2002] liegt auf der verteilten Erstellung von Ontologien.

Für [Sure ua 2002] ergab sich im Projekt SEAL eine Dreiteilung des Vorgehens in die Phasen (i) Anforderungsspezifikation, (ii) Verfeinerung und (iii) Evaluation. In dem ersten Schritt des als „Ontology Engineering“ bezeichneten Vorgehensmodells – der *Phase der Anforderungsspezifikation* – schließen sich Experten im Modellieren von Ontologien und Experten der abzubildenden Domäne⁷⁵ in Gruppen zusammen [Sure ua 2002:222ff]. Mit Unterstützung durch die Ontologiekonstrukteure definieren die Wissensträger das Ziel der zu erstellenden Ontologie, erstellen allgemeine Richtlinien zur Modellierung, erschließen schon vorhandenes Wissen, identifizieren potenzielle Nutzer, Anwendungsfälle sowie Anwendungen, welche später auf die Ontologien zugreifen sollen [Sure ua 2002:223]. Das Ergebnis ist ein Dokument, in dem alle ermittelten Anforderungen zusammengefasst werden. Daran schließt sich eine erste Beschreibung der abzubildenden Domäne in Form einer semi-formalen Ontologie an.

Der Anforderungsspezifikation folgt die Phase der Verfeinerung. Bei der Verfeinerung werden die informellen Erkenntnisse der ersten Phase in mehreren Iterationen mittels einer geeigneten Sprache formalisiert. An dieser Aktivität sind hauptsächlich die Ontologiekonstrukteure beteiligt. [Sure ua 2002] nutzten für die Verfeinerung bzw. Formalisierung der Ontologie das Werkzeug OntoEdit. Dieser Editor wurde für die Unterstützung einiger Aufgaben um Eigenimplementationen⁷⁶ erweitert. Das Ergebnis der zweiten Phase ist die fertige Ontologie.

⁷⁴SEAL im Internet: <<http://www.aifb.uni-karlsruhe.de/about.html>> (30.7.2005)

⁷⁵ Unter den vielen Bedeutungen des Wortes *Domäne* erscheint in dem vorliegenden Fall die Analogie zu dem Term *Fachgebiet* naheliegend [Wortschatz].

⁷⁶ Darunter Mind2Onto (eine Erweiterung zur Übernahme von Inhalten aus dem MindManager der Firma Mindjet) und OntoKick (Werkzeug zur gezielten Aufnahme der Anforderungen).

In der abschließenden Evaluationsphase untersuchen die Ontologiekonstrukteure in Zusammenarbeit mit den Wissensträgern und den späteren Nutzern die Ontologie auf ihre Brauchbarkeit und ihre Vollständigkeit im Sinne der Anforderungsspezifikation aus Phase Eins [Sure ua 2002:234]. Dabei werden Test-Ausgaben (*test sets*) aus der fertigen Ontologie erzeugt, die auf ihre Konsistenz hin überprüft werden.⁷⁷ Um die Ontologie auf semantische Fehler hin zu untersuchen, werden Aussagen, die aus der Ontologie generiert werden, auf inhaltliche Fehler untersucht. Den Abschluss findet die Evaluationsphase in der Beurteilung der Güte der Ontologie. Können mit der Ontologie qualifizierte Fragen beantwortet werden,⁷⁸ ist die Ontologie erfolgreich evaluiert und das Projekt beendet.

In einem zwei bis drei Jahre dauernden Prozess der Weiterentwicklung, bei dem York Sure (wahrscheinlich maßgeblich [Anm des Autoren]) beteiligt war, entfernte man sich von dem Vorgehen aus [Sure ua 2002]. Dieses hatte ursprünglich den Sinn, die Ontologie in einen möglichst perfekten Anfangszustand zu bringen, noch bevor ihre tatsächliche Nutzung stattfand.⁷⁹ Da eine Ontologie die Repräsentation von Wissen darstellt, und sich Wissen evolutionär weiterentwickelt und keineswegs absolute Gültigkeit besitzt,⁸⁰ musste ein Ansatz entwickelt werden, der diesem Umstand Rechnung trägt: das DILIGENT⁸¹-Vorgehensmodell.⁸² Es berücksichtigt die Tatsache, dass die Nutzer der Ontologie eine eigene Sichtweise auf die in der gemeinsamen Ontologie wiedergegebenen Inhalte haben und dass die gemeinsame Ontologie in regelmäßigen Abständen an die neuesten Erkenntnisse angepasst werden muss.

Der DILIGENT-Prozess gliedert sich in fünf Abschnitte: Erstellung, lokale Übernahme, Analyse, Revision und lokale Aktualisierung. Die Rollen definieren [Sure ua 2005:86f] so: Es gibt die Nutzer der Ontologie⁸³ und ein zentrales Gremium⁸⁴, das die Qualität der von allen genutzten (Kern-)Ontologie überwacht, diese wartet und aktualisiert.

⁷⁷ Prüfung auf syntaktische Korrektheit.

⁷⁸ Sogenannte *Competency Questions*, welche die Ontologie beantworten können sollte und wodurch eine Aussage über die Güte der Ontologie gemacht werden kann [Sure ua 2002:225].

⁷⁹ Ähnlich einem Produkt oder Werk, das konzeptioniert, entwickelt, dann genutzt und während der wird.

⁸⁰ Zustände und Regeln verändern sich mit der Zeit. Wissen, dh Sachverhalte und Ontologien ist nicht in einen fortwährend gültigen Zustand – das Wissen *lebt* also.

⁸¹ DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT).

⁸² Ein Vorgehens-(Referenz)Modell, dass [Sure ua 2005:86] als *process template* bezeichnet.

⁸³ Wohingegen vormals die Wissensträger gleichzeitig die Ontologie-Nutzer waren, wird hier in Wissensträger (am Erstellungsprozess beteiligt) und Ontologie-Benutzer (nicht am Erstellungsprozess beteiligt) unterschieden.

⁸⁴ Das zentrale Gremium (*central board* [Sure ua 2005:86]) besteht einerseits aus qualifizierten [im Sinne der Modellierung von Ontologien] Ontologie- und Wissenskonstrukteuren und andererseits aus Ontologie-Nutzern und Wissensträgern. Mit der guten Verteilung der Sitze wird die Akzeptanz der Benutzer erhöht.

Nachdem in der ersten, als *Build* bezeichneten Phase ein Anfangsstadium⁸⁵ der Kern-Ontologie erstellt wurde, wird diese in der Phase der lokalen Übernahme (*local adaptation*) allen Benutzern zugänglich gemacht. Die Nutzer übernehmen die Inhalte der Kern-Ontologie in ihre lokalen Kopien und passen diese ihren eigenen Bedürfnissen, sofern erforderlich, an.⁸⁶ Nutzer können keine Änderungen an der Kern-Ontologie machen. Diesbezügliche Anträge nimmt das zentrale Gremium entgegen.

Das Gremium untersucht in der Analysephase die lokalen Ontologien der Ontologie-Benutzer. Dabei versucht das Gremium Ähnlichkeiten zwischen den lokalen Ontologien zu erkennen. Mit den Eingaben der Benutzer und den Erkenntnissen aus der Analyse der lokalen Ontologien wird ein Konsens über die Änderungen der Kern-Ontologie für deren neue Version geschlossen.

Die Revision⁸⁷ dient dazu, die lokalen Ontologien hinsichtlich ihrer inhaltlichen Nähe zu der gemeinsamen Kern-Ontologie zu prüfen.⁸⁸ Die gemeinsame Ontologie ist von den Wissensträgern inhaltlich und von Ontologiekonstrukteuren aus technischer Sicht zu evaluieren.

Das DILIGENT-Vorgehen ist mit der Aktualisierung der lokalen Ontologien durch die Ontologie-Benutzer abgeschlossen. Das sollte immer dann passieren, wenn das Gremium eine neue Version der gemeinsamen Kern-Ontologie freigegeben hat.

2.2.2 Vorgehensmodell nach Simon

[Simon 2004] konzentriert sich auf methodische und organisatorische Schwerpunkte der verteilten Modellierung von *Geschäftsprozessen* mit SemTalk. Er reflektiert die Erfahrungen, die er bei der Konzeption und der Umsetzung eines verteilt durchgeführten Modellierungsprojekts im Rahmen der Lehrveranstaltung Systemanalyse II des Studiengangs Wirtschaftsinformatik der Fachhochschule Brandenburg im Jahr 2004 sammeln konnte. Da es sich um ein Projekt innerhalb der Lehre handelte, konnte ein idealtypisches Vorgehen für die verteilte Modellierung realisiert werden.⁸⁹

Die für das Projekt festgelegte Organisationsstruktur entsprach einem Einliniensystem, dem ein Projektleiter in oberster Ebene (Professor, Dozent der Lehrveranstaltung) vor-

⁸⁵ Das Ziel ist hierbei, in einem möglichst klein gehaltenen Personenkreis die Übereinstimmung über eine anfängliche Ontologie herzustellen. Offensichtlich ist das ein Ergebnis des schwierigen Vorgehens, das in [Sure ua 2002] beschrieben ist – bereits von Anfang an alle Beteiligten in den konsensuellen Erstellungsprozess der Ontologie mit einzubeziehen.

⁸⁶ Die Phase der lokalen Übernahme (Übernahme der gemeinsamen Ontologie und deren lokale Anpassung) reflektiert die Erkenntnis [Sure ua 2005:89], dass es keine von allen akzeptierte, stets gültige Ontologie über ein Fachgebiet geben kann: Jeder Mensch hat auf den Gegenstand der Ontologie eine eigene Sicht. Im Gegensatz dazu wurde in [Sure ua 2002] noch versucht, einen globalen Konsens zwischen allen Beteiligten (Ontologiekonstrukteure und den Wissensträgern) zu erreichen.

⁸⁷ Ehemals die Phase der Evaluation.

⁸⁸ Die inhaltliche Nähe zwischen lokaler Ontologien und der gemeinsamen Ontologie soll gewahrt werden. [Sure ua 2005:87] führt die höhere Akzeptanz auf Seite der Nutzer und den geringeren Wildwuchs zwischen lokalen Ontologien als Gründe für diese Maßnahme an.

⁸⁹ Eine exakte Adaption für die freie Wirtschaft verhindern die unterschiedlichen Randbedingungen, wie die verfügbare Zeit, IT-Infrastruktur und Anzahl der beteiligten Personen.

steht. Jede der drei Studiengruppen wurde in Teams unterteilt, wobei jedes Team eine der drei Rollen *Koordination*, *Qualitätssicherung* oder *Modellierung* übernahm. Während die Modellierungsteams hauptsächlich die aufgenommenen Prozesse zu modellieren hatten, nahmen die Teams der beiden anderen Rollen keine modellierenden Tätigkeiten wahr. Vertreter der Koordinierungsteams der drei Studiengruppen hatten sich mit dem Projektleiter (Dozent) abzustimmen. Dabei festgelegte Aufgaben und Termine wurden anschließend an die untergeordneten Modellierungs- und Qualitätssicherungsteams weitergegeben. Weiterhin hatten die Koordinierungsteams die gemeinsame Ontologie aus den von den Modellierungsteams eingereichten Prozess-Glossaren zu bilden. Folglich waren die Koordinierungsteams für die begriffliche Konsistenz über das gesamte Projekt verantwortlich (zentrales Vokabular).

Die mit der Qualitätssicherung beauftragten Teams hatten die Aufgabe, zu bestimmten Meilensteinen die Qualität der von den Modellierern zur Prüfung eingereichten Prozess-Modelle festzustellen. Wurde ein Modell beanstandet, erfolgte dessen Rückführung an das Modellierungsteam mit der Bitte um Überarbeitung.

Die fertiggestellten Modelle bildeten einer Hierarchie von Geschäftsprozessen. Die abschließende Maßnahme bestand demzufolge im rekursiven Importieren der Prozessmodelle in ein globales Modell: Modelle der unteren Detaillierungsebene mussten in Modelle der nächsthöheren Abstraktionsebene importiert werden. In dem zusammengeführten Modell, waren alle modellierten Prozesse über interne Verfeinerungen eingebunden.

2.2.3 Vorgehensmodell nach Weichhardt

Der *Redaktionsprozess für verteilte Modellierung* [Weichhardt 2005] beschreibt das Vorgehen bei verteilter Modellierung aus Sicht des Herstellers von SemTalk und ist größtenteils in der Praxis entstanden. Dabei ist eine Ähnlichkeit zu dem von [Simon 2004] erarbeiteten Vorgehen feststellbar, obwohl in [Simon 2004] ein stärkerer Detaillierungsgrad vorliegt und die Qualitätssicherung ein viel größeres Gewicht hat.⁹⁰ Des Weiteren ist die Organisationsstruktur in [Weichhardt 2005:1] auf die Rolle des Modellkoordinators und des Modellierers verschlankt.⁹¹

2.2.4 Muster in den Vorgehensmodellen

Über die hier vorgestellten Vorgehensmodelle zur verteilten Modellierung sind in den Jahren 2002 (hier [Sure ua 2002]) bis 2005 ([Weichhardt 2005], [Sure ua 2005]) entstanden. Dabei haben sich viele Gemeinsamkeiten in dem Vorgehen gezeigt, die als wiederkehrende Muster bezeichnet werden können.

⁹⁰ Hier heißt es anstatt *Domänenontologie veröffentlichen* und *Domänenontologie importieren* [Simon 2004]: *zentrales Informationsmodell bereitstellen* und *zentrales Objekt importieren* [Weichhardt 2005:1].

⁹¹ Analog zu [Sure ua 2005].

(1) Anforderungsspezifikation. In den Vorgehensmodellen von [Sure ua 2002] und [Simon 2004] beginnen sofort nach der Festlegung der Teams Tätigkeiten zur Anforderungsdefinition hinsichtlich der Durchführung des Modellierungsprojekts. Dabei werden Vorgaben wie Designrichtlinien, Teilaufgaben, Termine und Ziele festgelegt. Mit dieser Tätigkeit ist im Wesentlichen die Projektleitung betraut. Diese hat sich über die festzulegenden Aspekte abzustimmen und anschließend allen Beteiligten zur Verfügung zu stellen.

In den später entstandenen Vorgehensmodellen ist der Vorgang zur Spezifikation der Anforderungen weggefallen (siehe [Sure ua 2005] und [Weichhardt 2005]). Es ist zu vermuten, dass in der Praxis zuviel Zeit in der Anfangsphase des Projekts verstrichen ist, bis erste Ergebnisse der Modellierung sichtbar wurden.⁹²

(2) Semi-formale Version der Ontologie. Das Erstellen und Veröffentlichen einer ersten (*semi-formalen* [Sure ua 2002]) Version der gemeinsamen Ontologie ist durchweg anerkannt. Das ist wichtig, damit die Modellierer bzw. Ontologiekonstrukteure diese für ihre Tätigkeiten – die Modellierung von Teilaufgaben – nutzen können. Diese erste Version hilft einen gemeinsamen Sprachgebrauch zu Beginn des Modellierungsprojekts herzustellen.⁹³

Für das gemeinsam zu nutzende Modell, das im weiteren Verlauf eines Projektes weiterentwickelt wird, ist immer eine Organisationseinheit verantwortlich zu machen.⁹⁴ Nur diese darf die gemeinsame Ontologie warten, eine neue Version erzeugen und bereitstellen.

(3) Übernahme der Ontologie. Nach der Veröffentlichung der ersten Version der gemeinsamen Ontologie wird diese von den Nutzern übernommen.⁹⁵ Nutzer können im Fall der verteilten Modellierung von Geschäftsprozessen die Modellierer der Teilprozesse oder, im Fall der Erstellung einer Ontologie, die Ontologie-Nutzer im Unternehmen sein [Sure ua 2005]. Für die lokale Übernahme des gemeinsamen Modells müssen die Nutzer über Zugriffsrechte auf das gemeinsame Modell verfügen.

(4) Nutzung, Anpassung und Erweiterung. Es schließt sich die Phase der Bearbeitung der Teilaufgaben an.⁹⁶ Die Ontologie-Nutzer, Modellierer oder Ontologiekonstrukteure

⁹² Das Ziel lautet nun möglichst früh eine erste Version der späteren Ontologie zu haben. Diese wird in einem evolutionären Prozess weiter entwickelt, verfeinert und formalisiert.

⁹³ Oft auch als Glossar bezeichnet.

⁹⁴ In [Sure ua 2002] nicht konkret benannt aber aus dem Kreis der *Ontologiekonstrukteure* gebildet; bei [Simon 2004] und in [Weichhardt 2005] ist es das Team der *Koordinatoren*; In [Sure ua 2005] wird für die gemeinsame Ontologie das *zentrale Gremium* aus Vertretern der beteiligten Fraktionen gebildet.

⁹⁵ In SemTalk hieße der Vorgang *Import* mit Erzeugung einer *externen Referenz*.

⁹⁶ Modellierer in [Weichhardt 2005] und [Simon 2004] beginnen, die ihnen übertragenen Geschäftsprozesse zu modellieren. Dagegen fangen die Ontologie-Nutzer in [Sure ua 2005] an ihre lokale Kopie der gemeinsamen Ontologie an die eigenen Bedürfnisse anzupassen.

können ihre lokalen Modelle frei bearbeiten. Werden dabei Erkenntnisse gemacht, welche in das gemeinsame Modell oder die gemeinsame Ontologie übernommen werden sollten, muss ein Antrag an die verantwortliche Organisationseinheit gestellt werden. Arbeiten mehrere Personen an einem Teilmodell, müssen diese sich hinsichtlich ihrer Entscheidungen bezüglich der Modellierung abstimmen. Es kommt vor, dass Fehlentscheidungen rückgängig gemacht werden müssen.

(5) Aktualisierung, Revision und Evaluation. Nach Beendigung der modellierenden Tätigkeiten setzt zum Einen die *Qualitätssicherung* der lokalen Modelle und zum Anderen der gemeinsamen Ontologie oder des gemeinsamen Modells ein. Dabei werden die im vorherigen Schritt eingebrachten Anträge von der verantwortlichen Organisationseinheit im Turnus geprüft und gegebenenfalls in das gemeinsame Modell eingearbeitet. Parallel dazu findet die Prüfung der lokalen Modelle bzw. Ontologien durch die kompetente Organisationseinheit statt. Nach der Veröffentlichung der neuen Version sind die Neuerungen des gemeinsamen Modells von den Modellierern, Ontologie-Nutzern oder Ontologiekonstrukteuren in ihre lokalen Modelle zu übernehmen.

Zwischen den Phasen (4) und (5) besteht bei allen Vorgehensmodellen, bis auf [Weichhardt 2005], ein Zyklus. Damit wird der iterative Prozess zur kontinuierlichen Verbesserung des gemeinsamen Modells hergestellt.

2.3 Plattformen für die gemeinsame Erstellung von Ontologien

Das verteilte Erstellen von Ontologien ist in verschiedenen Vorhaben bereits durchgeführt worden. Dabei mussten sich die Verantwortlichen bezüglich der Gestaltung der Zusammenarbeit Gedanken machen.

(1) Agentcities. Das erklärte Ziel des von der EU finanzierten Projekts Agentcities.RTD war „the dynamic, intelligent and autonomous composition of Web services to achieve user and business goals“ [Ceccaroni; Kendall 2002:1]. Hierfür mussten gemeinsam aber ortsunabhängig Ontologien hergestellt werden. Diese sollten anschließend von *agent-based services* – also Software-Programmen – genutzt werden [Ceccaroni; Kendall 2002:1].

Die während der Wartung und Anpassung entstandenen Verwirrungen und Ungenauigkeiten zwischen den Beteiligten der Ontologierstellung [Ceccaroni; Kendall 2002:3] konnten nur mittels Auswahl und Einführung eines grafischen Modellierungswerkzeugs beseitigt werden. Man entschied sich für den *Medius Visual Ontology Modeler* (VOM)⁹⁷. Er

⁹⁷Zusatzmodul für das UML-Modellierungswerkzeug Rational Rose (IBM) der Sandpiper Software.
<<http://www.sandsoft.com/products.html>> (11.1.2006)

erfüllte die von [Ceccaroni; Kendall 2002] definierten fünf grundlegenden Anforderungen an ein Modellierungswerkzeug für die gemeinsame Erstellung von Ontologien: (i) einfache Benutzerführung und leichte Erlernbarkeit, (ii) eine grafische Repräsentation der Modellierungsergebnisse,⁹⁸ (iii) Funktionen des Konfigurationsmanagements,⁹⁹ (iv) integrierte Überprüfung von Konsistenz und Vollständigkeit des Modells sowie (v) die Möglichkeit zum Export des Modells in das Format DAML+OIL¹⁰⁰ [Ceccaroni; Kendall 2002:3]. Weiterführende Funktionen, wie das (i) Editieren in Echtzeit, (ii) eine Multiuser-Entwicklungsumgebung, (iii) Möglichkeiten zum Betrachten, Analysieren, Vergleichen und Zusammenführen mehrerer Modelle und (iv) der Import von DAML+OIL, OWL¹⁰¹, RDFS¹⁰² und älteren Speicherformaten für Ontologien sind bis dahin noch nicht umgesetzt – könnten aber nötig sein [ebenda]. Außerdem wurde festgestellt, dass die vom VOM bereitgestellten Assistenten-Programme und die damit mögliche automatische Erzeugung von Modellelementen und -diagrammen zwar nicht unbedingt erforderlich waren, jedoch bei der Modellierung halfen.

Den Bedarf an diesem umfangreichen Anforderungskatalog meinen [Ceccaroni; Kendall 2002:3] vor allem bei großen Projekten zu erkennen, deren Kernaufgabe die gemeinsame Erstellung von Ontologien ist.

(2) Die Semantic Collaborative Environment Architecture. Das von der Firma TopQuadrant durchgeführte Projekt mit dem Thema „Semantic Technology and Model-Based Life-Cycle Support for NASA Space Vehicle Engineering“ [Hodgson 2005] hatte als vorrangiges Ziel, die Ideen des *Semantic Web* in die Entwicklung und den Bau von Raumfahrzeugen einzubetten. Da Raumfahrzeuge komplexe Produkte sind, deren Herstellung nur durch Mitwirkung mehrerer Unternehmen möglich und darüberhinaus zahlreichen Risiken ausgesetzt ist, konnte hier der Bedarf an Ontologien ausgemacht werden.

Hodgson, Mitarbeiter des von der NASA beauftragten Beratungsunternehmens TopQuadrant, stellt in [Hodgson 2005:29] die *Semantic Collaborative Environment Architecture* (SCEA) – eine Referenzarchitektur für eine Plattform zur Unterstützung der Zusammenarbeit – vor. Sinn und Zweck der SCEA ist die Unterstützung der sechs am Lebenszyklus eines Raumfahrzeugs beteiligten Arbeitsgruppen.¹⁰³

⁹⁸ Mindestens in Form von HTML.

⁹⁹ Gemeint ist eine Versionsverwaltung für Modelle, die das Wiederherstellen alter Versionen der Ontologie gewährleistet.

¹⁰⁰ Historischer Ursprung der OWL, mit dem sich Ontologien beschreiben ließen. Aus heutiger Sicht erscheint der Modellexport in die Formate DAML+OIL überholt. Das vom W3C als Nachfolger ins Leben gerufene OWL-Format sollte hier zum Standard werden.

¹⁰¹ Die Web Ontology Language, ein XML-Dialekt der technisch auf dem RDFS beruht und dieses aber an Ausdrucksstärke übertrifft, wird mit dem Akronym OWL abgekürzt. Mit ihr können, wie zuvor mit DAML+OIL und RDFS, Ontologien für das Semantic Web erzeugt und bereitgestellt werden.

¹⁰² Resource Description Framework Schema ist eine Syntax-Empfehlung des W3C für die Beschreibung einer Domäne mittels XML. Das bedeutet, mit RDFS können Ontologien erstellt werden.

¹⁰³ Siehe hierzu [Hodgson 2005:28]. Darunter die Teams der Systemingenieure und Missionsplaner aber auch der IT-Verwaltung.

In Abbildung 12 ist der Aufbau der SCEA dargestellt. In der SCEA kann die Gruppierung von Funktionen und Werkzeugen erkannt werden.¹⁰⁴ In der vorliegenden Abbildung sind die in den Bereichen *Graphics*, *Semantic Infrastructure*¹⁰⁵ und *Platform Infrastructure* eingesetzten Technologien ausgeblendet. Denn im Gegensatz zu den anderen Anforderungsbereichen sind in [Hodgson 2005] dazu konkrete Systeme benannt.¹⁰⁶

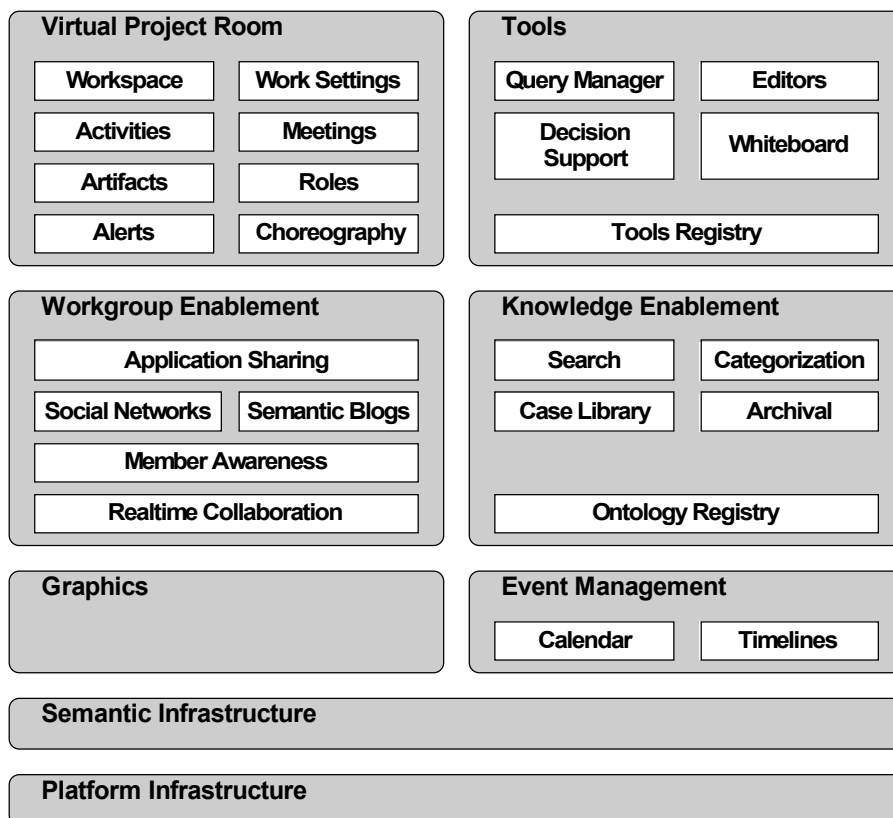


Abbildung 12. *Semantic Collaborative Environment Architecture*. Quelle: [Hodgson 2005:29]

Ein identifizierter Anwendungsfall der SCEA ist besonders interessant: Die Unterstützung des *Ontology Lifecycle Management Environment* mittels der SCEA [Hodgson 2005:28]. Hodgson erkannte den dringenden Bedarf an einer Plattform für die Erstellung und Wartung von komplexen Modellen [wie den angesprochenen Ontologien]. Grund dafür ist das gemeinsame Vorgehen zu deren Erstellung einerseits und die nötige Wartung und das Management von Modellen andererseits [Hodgson 2005:26]. Die Notwendigkeit einer Architektur während des Erstellungsprozesses begründet Hodgson mit den dabei

¹⁰⁴ Es ist nicht bekannt, woran sich der Bedarf der einzelnen Werkzeuge orientiert.

¹⁰⁵ TopQuadrant versteht unter der Semantic Infrastructure Semantische Webdienste, kontext- und situationsbezogene Datenverarbeitung, ortsungebundene Datenverarbeitung sowie ontologiebasierte Denkmachines (Mills Davis, Dean Allemang, Robert Coyne. 2004. WONDERWEB DELIVERABLE D25: Evaluation and Market Report. <<http://wonderweb.semanticweb.org/deliverables/documents/D25.pdf>> (31.3.2006)). Werkzeuge dafür sind: Editoren, Vermittler, Übersetzer, automatisierte Denkmachines und Verifikatoren.

¹⁰⁶ Anstelle hier wie in den anderen Anforderungsdimensionen abstrakte Klassen zu definieren, sind die konkret genannten Systeme (Instanzen) für eine allgemeine Betrachtung ungeeignet.

anfallenden organisatorische Aufgaben: Schaffung von Namensräumen, Änderungsmitteilungen, die Auftrennung und Zusammenfügung von Modellen [ebenda].

2.4 Die Windows SharePoint Services

In dem Gefüge der Microsoft-Office-Produkte nehmen die Windows SharePoint Services (WSS) die Aufgabe der Bereitstellung *gemeinsamer Arbeitsbereiche*¹⁰⁷ wahr. Microsoft beschreibt die WSS als ein *integriertes Portfolio von Diensten zur Unterstützung von Kooperation und Kommunikation mit dem Ziel, Menschen, Informationen, Prozesse und Systeme miteinander zu verbinden*.¹⁰⁸ Tafel 1 zeigt das *Microsoft Office System* mit den dazugehörigen Produkten im Überblick.

Microsoft Office System		
<i>Clientsseitige Anwendungen</i>		<i>Server und Services</i>
<i>Standard-Anwendungen</i>	<i>Zusatz-Anwendungen</i>	
Word, Excel, Access, Outlook, PowerPoint	Project, Visio, InfoPath, FrontPage, OneNote, Communicator und Publisher	Live Communication Server, Project Server, SharePoint Portal Server, Windows SharePoint Services, Exchange Server und Live Meeting

Tafel 1

Technisch betrachtet, handelt es sich bei den WSS um eine hochgradig skalierbare Erweiterung für den Microsoft Windows Server 2003. Sie werden von Microsoft kostenfrei zum Herunterladen angeboten.

Der gemeinsame Arbeitsbereich ist der Ort für die kooperative Zusammenarbeit eines Teams, in dem gemeinsame Dokumente, Aufgaben und Termine verwaltet werden. Er lässt sich an die Bedürfnisse der Benutzer und an das gemeinsam zu erreichende Ziel anpassen.

Technisch gesehen ist der gemeinsame Arbeitsbereich ein in den WSS logisch abgegrenzter Teil, der über eine Einstiegsseite – ein *Portal* – erreicht wird.¹⁰⁹ Portale selbst werden als Internet-Präsentation eines Unternehmens definiert, die darüberhinaus in der

¹⁰⁷ Unter gemeinsamen Arbeitsbereichen versteht Microsoft Orte, an denen Ideen, Informationen und Dokumente ausgetauscht und gemeinsam bearbeitet werden: <<http://www.microsoft.com/windowsserver2003/techinfo/sharepoint/overview.mspx>> (8.1.2006)

¹⁰⁸ Sinngemäß übersetzt von <<http://www.microsoft.com/windowsserver2003/technologies/sharepoint/default.mspx>> (8.1.2006)

¹⁰⁹ Im Internet erreichbare Portale verbergen hinter der Einstiegsseite – dem Portal – das gesamte Onlineangebot eines Anbieters.

Lage sind, personalisierte Inhalte anzeigen zu können sowie bedarfsgerechte Prozesse anzubieten [Kirchhof ua 2004:5]. In Verbindung mit der ständigen Verfügbarkeit sowie der Orts- und Betriebssystemunabhängigkeit ergeben mit der Einführung eines Portals deutliche Vorteile für ein Unternehmen.

Ein Portal hat einen der Zwiebel ähnlichen Aufbau: Die äußeren für die Öffentlichkeit frei zugänglichen Portalbereiche können als ein Business-2-Consumer-Angebot (B2C) verstanden werden [Bauer 2001:37]. Davon abgegrenzte, zugangsbeschränkte Seiten stehen nur Geschäftspartnern offen und haben damit einen Business-2-Business-Charakter (B2B) [ebenda]. Hier wird ein Großteil der Geschäftsprozesse, die mit Geschäftspartnern im Zusammenhang stehen, abgedeckt. Das Mitarbeiterportal (Business-2-Employee, B2E) könnte ein dritter, nichtöffentlicher Bereich in einem Portalangebot sein, indem unternehmensinterne Informationen (zB interne Stellenausschreibungen, Betriebsrat) bereitgestellt werden.

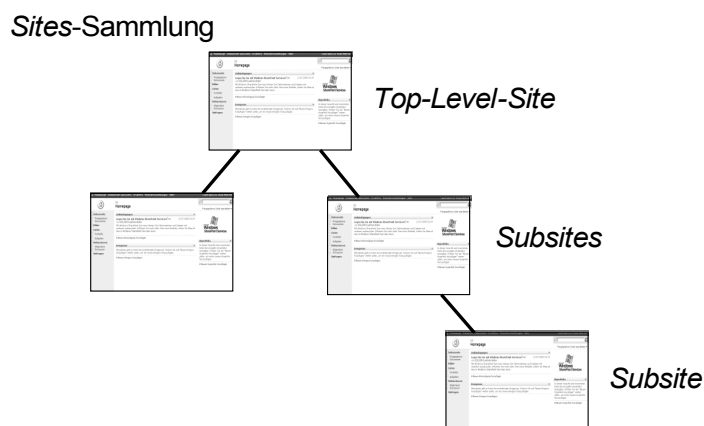


Abbildung 13. Hierarchie von Sites in den WSS. Quelle: [Microsoft 2003c].

In einer WSS-Installation können beliebig viele gemeinsame Arbeitsbereiche erzeugt werden, wobei lediglich die von den WSS verwendete Datenbank Grenzen setzt. Damit kann jede Organisationseinheit in einem Unternehmen über einen eigenen gemeinsamen Arbeitsbereich verfügen. Die im Unternehmen vorhandene Organisationsstruktur lässt sich über eine Hierarchie von Arbeitsbereichen nachbilden. Abbildung 13 verdeutlicht dies. Dabei entspricht jede in der Abbildung gezeigte *Site* einem gemeinsamen Arbeitsbereich.

Portale werden mit Hilfe einer Portalsoftware realisiert. Die WSS, mit denen sich gemeinsame Arbeitsbereiche bereitstellen lassen, sind eine Art Portalsoftware. Im Folgenden wird der Frage nachgegangen, was ein Portal ausmacht¹¹⁰ und welche Ausprägungen diese Eigenschaften im Fall der WSS haben.

¹¹⁰ Erste Bemühungen zur Beantwortung dieser Frage finden sich in [Bauer 2001:38ff], zeitlich gefolgt von [Gurzki 2002:4ff], [Gurzki; Hinderer 2003] sowie in [Kirchhof ua 2004].

Die Philosophie von Portalen basiert mehr auf den Funktionen als auf der dem Portal zugrundeliegenden Technik – der Portalsoftware. In [Bauer 2001] werden erstmals die fünf Basisfunktionalitäten eines Portals eingeführt. Diese sind:

1. Personalisierung,
2. Benutzer- und Rechteverwaltung,
3. Dynamische Inhalte und Webpublishing,
4. Externe Webanwendungen sowie
5. Integration von Unternehmensanwendungen.

[Kirchhof ua 2004:8] sind mit dieser Liste von [Bauer 2001] im Wesentlichen konform, erweitern sie jedoch um die Aspekte *Suche*, *Single-Sign-On* und *erweiterte Portalmodule*. Hierfür entwickelten [Gurzki; Hinderer 2003:2] die ”Referenzarchitektur für Portal-Software“¹¹¹. Die von [Bauer 2001:40f] eingeführten Merkmale der *Dynamischen Inhalte* und *Webpublishing* werden in [Gurzki; Hinderer 2003:2] sowie in [Kirchhof ua 2004:9] durch die Begriffe *Content-*, *Layout-* sowie *Strukturmanagement* ersetzt. Im Anhang II ist die Referenzarchitektur für Portal-Software nach [Gurzki; Hinderer 2003] abgebildet.

Die für Portalsoftware charakterlichen Aspekte werden nun unter Berücksichtigung ihrer Umsetzungen in den WSS näher erläutert. Dabei geben die von [Bauer 2001] eingeführten Basisfunktionalitäten die schwerpunktmäßigen Schritte vor. Nach der einleitenden Besprechung der *Personalisierung* werden die *Benutzerverwaltung* und daraufhin die *dynamischen Inhalte* und *Webpublishing* vorgestellt. Die Einbindung *externer Webanwendungen* und die *Integration von Unternehmensanwendungen* bilden den Abschluss der Einführung in Portalsoftware.

¹¹¹ Abweichend der sich in [Gurzki; Hinderer 2003] abwechselnden Schreibweisen ”Portal Software“ und ”Portal-Software“ wird in dieser Arbeit durchgängig die Schreibweise *Portalsoftware* verwendet.

2.4.1 Personalisierung

Ist es dem Benutzer eines Portals möglich, die Darstellung von Seiteninhalt und Aussehen einer Portalseite individuell an seine Bedürfnisse anzupassen, wird von *Personalisierung*

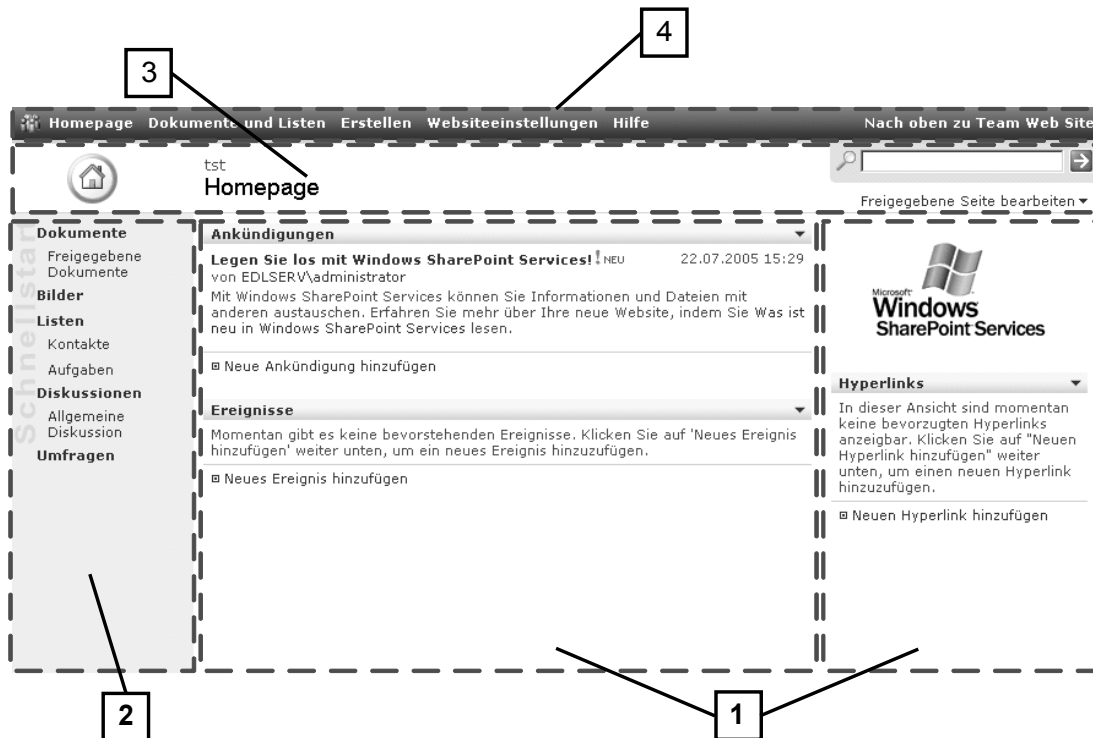


Abbildung 14. Eine typische Webpartseite.

gesprächen [Bauer 2001:38f],[Kirchhof ua 2004:3, 8]. Den grundlegenden Aufbau einer durch die WSS bereitgestellten Seite macht Abbildung 14 deutlich. Das für Webseiten der WSS charakterliche Drei-Spalten-Layout [Bauer 2001:66], bestehend aus [1] den zwei Spalten für Portalelemente und [2] dem Schnellstartbereich, ist leicht erkennbar. Bei [3] befindet sich ein variabler Seitenkopf mit dem Titel der Webseite, dem Eingabefeld für die Suche und dem Menüknopf für die Anpassung der Webseite. Der feste Seitenkopf mit Navigations- und Administrationsmenüs ist bei [4] zu finden. Dieser ist typisch für Portale [Bauer 2001:58].

Die WSS erlauben es Benutzern die über administrative Rechte verfügen, ihre Oberfläche nach den eigenen Bedürfnissen anzupassen. Im Begriffsraum der SharePoint-Produkte wird diese als *personal view* bezeichnet. Der Benutzer der über die Rechte verfügt, die *personal view* anzupassen, darf auch die Sicht derer modifizieren, die in dem gemeinsamen Arbeitsbereich lediglich über Leser-Rechte verfügen (*shared view*).

(1) **Webparts.** Eine normale WSS-Seite ist ähnlich dem Produkt eines Baukastensystems aus einzelnen Elementen zusammengestellt. Die Oberflächenbausteine eines Portals – allgemein als Portlets bezeichnet [Gurzki 2002:5] – heißen hier *Webparts*. Diese lassen sich einfach per Drag&Drop-Mechanismus aus einem Webpart-Katalog in die zu verändernde Webseite einbinden. Ebenso einfach ist das Verschieben und das Entfernen

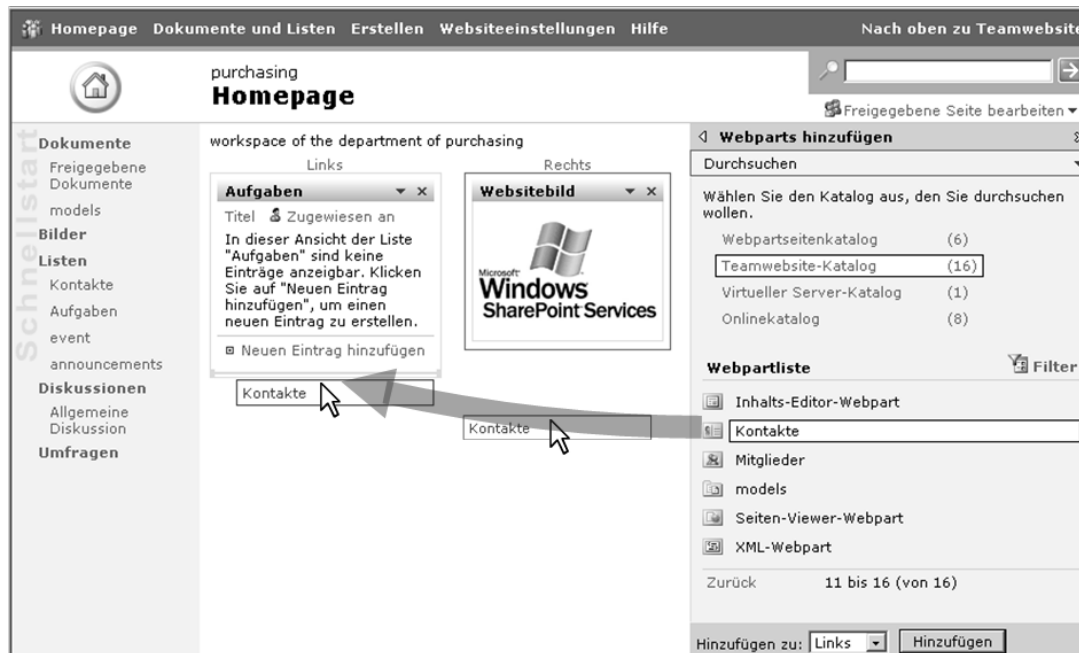


Abbildung 15. Hinzufügen eines Webparts auf eine Webpart-Seite

nicht erwünschter Webparts. Befindet sich der Benutzer im Modus der Anpassung, kann der Webpartkatalog im rechten Bereich der anzupassenden Webseite eingeblendet werden. Abbildung 15 zeigt, wie das Webpart *Kontakte* aus dem Webpartkatalog mit der Maus in die linke Spalte der Webpart-Seite hineingezogen wird.

Der Aufbau eines Webparts entspricht in den Grundzügen den von [Bauer 2001:67] eingeführten Komponenten "Titelleiste mit Funktionsicons" und "Inhalt". [Bauer 2001:ebenda] unterteilt Portlets zudem in die Komponenten "Titelbanner", "Kopf" und "Fußbereich". Da diese Elemente aber weder genau voneinander abgegrenzt werden können, noch – im Fall von Webparts – immer anzutreffen sind, erscheinen sie für eine allgemeingültige Definition nicht geeignet.

In den Webparts kann Geschäftsprozesslogik implementiert werden. [Bauer 2001:59] bezeichnet dies als eine für Portlets übliche Funktionalität. Für die Realisierung der Logik können alle Hochsprachen des Microsoft .NET-Framework verwendet werden.¹¹²

Nach Installation der WSS ist nur ein begrenzter Katalog von Webparts verfügbar. Eine Schnittstelle an den Online-Webpartkatalog von Microsoft erlaubt die Einbindung

¹¹² Im Rahmen dieser Arbeit konnten jedoch nur Beispiele ermittelt werden, bei deren Realisierung entweder Programmiersprache C#.NET oder VB.NET zum Einsatz kam.

weniger zusätzlicher Webparts. Diese von Microsoft zur Verfügung gestellten Webparts implementieren meist einen Zugriff auf Informationen, die von *Webdiensten* angeboten werden. So lassen sich beispielsweise Wetterinformationen und Börsenkurse mittels Webparts in den gemeinsamen Arbeitsbereich einbinden.

Das Aussehen der standardmäßig mitgelieferten Webparts kann bis zu einem gewissen Grad angepasst werden: Bis auf wenige für alle Webparts gleichermaßen verfügbare Funktionen, wie dem Minimieren und Entfernen, können Webpart-abhängige Anpassungen durchgeführt werden. Das Anpassen eines Webparts erfolgt in einem Konfigurationsfenster, das im rechten Bereich einer WSS-Webseite eingeblendet wird. Dabei wird die ursprünglich besuchte Webseite nicht verlassen. Ist die Anpassung des Webparts vollendet und das Konfigurationsfenster geschlossen, erscheint die Webseite mit dem angepassten Webpart.

Neben den, zugegebenermaßen begrenzten, standardmäßig verfügbaren Webparts, lassen sich die Webpart-Kataloge um eigen entwickelte Webparts erweitern (siehe oben). Die für die Einbindung der Eigenentwicklungen verfügbaren Dokumentationen und Mechanismen sind leider äußerst mangelhaft und schwer verständlich; die dafür nötigen Erweiterungen existierender Entwicklungswerkzeuge von Microsoft wären überhaupt nicht vorhanden, wenn nicht die Webparts auch die grundlegenden Oberflächenelemente des kommerziell vertriebenen SharePoint Portal Servers (SPS) wären.

Hier scheint ein Konflikt beim Hersteller vorzuliegen, der auf der einen Seite nicht will, dass ein kostenlos angebotenes Produkt (WSS) beliebig erweitert werden kann und andererseits den Käufern des SPS die Möglichkeiten zu dessen Anpassung nicht versagen kann. Glücklicherweise unterstützen zahlreichen private Internetseiten und Diskussionsgruppen im deutsch- sowie englischsprachigem Usenet die Gemeinschaft der WSS-Anwender.¹¹³

(2) Listen. Die zweite Möglichkeit, eigene Informationen auf die Oberfläche einer WSS-Webseite zu bringen, steht mit den *Listen* zur Verfügung. Dies sind in ASP.NET entwickelte Oberflächenbausteine, deren Erscheinungsbild Tabellen ähneln. Listen sind aber keine physikalischen Tabellen im Sinne einer Datenbanktabelle oder einer Tabelle eines Tabellenkalkulationsprogramms. Vielmehr handelt es sich bei Listen um virtuelle Tabellen, die zum Zeitpunkt des Aufrufs, dh dynamisch, aufgebaut werden.

Die WSS bieten eine Anzahl vordefinierter Listentypen für die Darstellung von Ankündigungen, Ereignissen, Kontakten, Umfragen, Diskussionen, Aufgaben und Hyperlinks an. Hinzu kommen Listen, die der Anwender selbst entwerfen und mit Inhalten füllen kann. Die Eigenschaften einer Liste, zB die Spaltenüberschriften und Datentypen

¹¹³ Eine etwas allgemeine Newsgroup, in der aber deutsch gesprochen wird <microsoft.public.de.german.sharepoint.windowsservices>; umfangreicher in Englisch unter <microsoft.public.sharepoint.windowsservices> und eine für Entwickler unerlässliche Informationsquelle befindet sich in <microsoft.public.sharepoint.windowsservices.development>.

für den Zelleninhalt, können auf einer dazugehörigen Konfigurations-Webseite verändert werden.

Mit der *Dokumentbibliothek* steht eine spezielle Art von Liste zur Verfügung: In ihr werden gemeinsame Dokumente gespeichert. Für die Dokumentbibliothek stehen zusätzliche Funktionen zur Verfügung. So können Dokumente hochgeladen, eine neue Version erzeugt, für den exklusiven Zugriff durch einen Benutzer gesperrt und wieder freigegeben werden. Des Weiteren lassen sich in Dokumentenbibliotheken (virtuelle) Ordner anlegen.

Für jede Art von Liste und Listeneintrag kann sich ein Benutzer für eine Mitteilung anmelden, die ihn über ein darin stattgefundenes Ereignis¹¹⁴ informiert. Diese Mitteilung wird ihm von den WSS per Email zugesandt.

Eine WSS-Webseite kann nur unter dem Betriebssystem Microsoft Windows und unter Verwendung des Internet Explorer ab der Version 5.5 vollständig genutzt werden. Das betrifft die Darstellung, die in Browsern anderer Hersteller teilweise anders wiedergegeben wird, aber auch die Möglichkeiten der Interaktion. Dies fällt dann auf, wenn beispielsweise die Webseite im Rahmen der *Personalisierung* angepasst werden soll: Die weiter oben erwähnte Drag&Drop-Funktion zum Einfügen, Verschieben und Entfernen von Webparts funktioniert in einem anderen Browser als dem Internet Explorer meist nicht.¹¹⁵ Vermutlich haben diese Schwierigkeiten ihren Ursprung in der Verwendung von JavaScript-Funktionen, die nicht dem industriellen Standard entsprechen.

2.4.2 Benutzerverwaltung und Rechte

Werden in einer kooperativen Umgebung die WSS eingesetzt, liegt eine enge Verzahnung von dem zugrundeliegenden Betriebssystem (Microsoft Windows Server 2003) mit dem Webserver (Internet Information Server 6.0 (IIS)) und den WSS als Portalsoftware vor. Für die WSS ist es so möglich, die schon im Betriebssystem eingepflegten Benutzerkonten zu nutzen: Dem IIS wird mitgeteilt, dass sich nur Benutzer des Windows Betriebssystems an dieser Website anmelden dürfen. Dies erfolgt durch die Aktivierung von *Windows Authentication* in der Website-Konfiguration. Versucht nun eine Clientanwendung – üblicherweise ein Webbrowser – eine von den WSS verwaltete Webseite anzufordern, öffnet sich ein Dialog, der die Windows-Benutzerdaten des Anwenders erfragt. Werden die WSS in einer durch den *Active Directory*-Verzeichnisdienst¹¹⁶ verwalteten Intranet-Domäne eingesetzt, sind die im *Active Directory* angelegten Benutzer für den Zugriff auf Webseiten der WSS zugelassen. Eine nur durch die WSS verwaltete Benutzerführung ist auf Grund der

¹¹⁴Unter Ereignissen werden Aktionen verstanden, die eine Veränderung des Inhalts herbeiführen.

¹¹⁵Eine Übersicht der Unterstützung unterschiedlicher Browser ist unter <http://msdn.microsoft.com/library/en-us/spptsdk/html/smpxClients_SV01030793.asp> abrufbar.

¹¹⁶Verzeichnisdienst von MS. Verzeichnisdienste sind Datenbanken in einem lokalen Netzwerk, die Benutzerdaten und Anwendungen der Domäne verwalten.

Verflechtung mit dem Betriebssystem nicht möglich. Damit spart sich Microsoft die Umsetzung einer doppelten Benutzerverwaltung für die WSS.

Die Benutzerrechte innerhalb der WSS sind trotz der Zuhilfenahme des Betriebssystems von diesem inhaltlich losgelöst: Benutzer, die im Windows Server eingerichtet sind, haben nicht per se Zugriff auf die WSS. Sie müssen erst in den WSS als Benutzer hinzugefügt werden. Der Windows Server-Administrator hat zwar dann immer auch Administratorenrechte innerhalb aller WSS-Bereiche, aber Administratorenrechte für die WSS können auch einem einfachen Windows-Benutzer zugesichert werden. Damit ist die logische Trennung zwischen dem Benutzerkonto der Domäne (Windows Server) und dem WSS-Benutzerkonto vollzogen.

Die Benutzerrechte für die WSS werden entweder auf eine Rolle bezogen oder einzeln vergeben. Neben den vordefinierten Rollen lassen sich neue Rollen mit genau definierten Rechten einrichten. Unabhängig der Rechte, die ein Benutzer in dem gemeinsamen Arbeitsbereich besitzt, können seine Rechte auf Webseiten explizit erweitert oder beschränkt werden. Dies ermöglicht Benutzern, die in dem gemeinsamen Arbeitsbereich lediglich über lesende Rechte verfügen, für eine darin befindliche Webseite schreibenden Zugriff zu gestatten. Ein typisches Anwendungsbeispiel sind Webseiten, auf denen Diskussionen zwischen Lesern erlaubt sind. Ebenso ist denkbar, dass Mitglieder einer Gruppe in ihrem Gruppenarbeitsbereich über umfangreichere Rechte verfügen können, als jene, die sie in dem gemeinsamen Arbeitsbereich haben.

Das Definieren spezieller Zugriffsrechte beschränkt sich allerdings auf eine Webseite. Für einzelne Webparts oder Listeneinträge können die Rechte der Benutzer *nicht* angepasst werden. Das hat zur Folge, dass sich die Zugriffsrechte für einzelne Dokumente einer Dokumentbibliothek nicht speziell erweitern oder beschränken lassen, da die Dokumente als Einträge in einer Liste (Dokumentbibliothek) realisiert sind.

2.4.3 Dynamische Inhalte und Webpublishing

Der IIS ist in der Lage, mehrere Webserver-Anwendungen parallel zu verwalten. Der Zugriff auf die einzelnen Webserver-Anwendungen erfolgt über sogenannte *Websites*, wobei jede Website einem *Anwendungspool* zugeordnet ist. Die Websites des IIS werden auch als *virtuelle Server* bezeichnet. Die virtuellen Server (bzw Websites) definieren sich über den Rechnernamen oder die IP-Adresse und die Anschlussnummer (eng Port), die der Website bei ihrer Einrichtung mitgeteilt wird. Die Installationsroutine der WSS richtet automatisch zwei solcher virtuellen Server mit den zugehörigen Anwendungspools ein.¹¹⁷

Alle in einem IIS vorbereiteten virtuellen Server können mit der WSS-Anwendung erweitert werden. Dies geschieht auf der Webseite der *SharePoint-Zentraladministration*. Auf diese Weise können viele WSS-Top-Level-Wurzelseiten (siehe 2.4) auf einem einzigen

¹¹⁷ Ein virtueller Server wird für die SharePoint-Zentraladministration auf oberster Ebene benötigt und ist von allen anderen WSS-Erweiterungen losgelöst. Die WSS erweitern automatisch die immer vorhandene IIS-Website, die über Anschlussnummer 80 verfügbar ist.

realen Windows Server eingerichtet werden.¹¹⁸ Grenzen setzt die Anzahl der verfügbaren Anschlussnummern.¹¹⁹

Jedem von den WSS verwalteten virtuellen Servern ist eine Inhaltsdatenbank zugeordnet. In ihr sind alle Daten der gemeinsamen Arbeitsbereiche, wie die gemeinsamen Dokumente, Aufgaben, Diskussionen und Umfrageergebnisse gespeichert. Die Zuordnung der Inhaltsdatenbanken zu den virtuellen Servern nimmt die nur einmal vorhandene Konfigurationsdatenbank vor.

Ruft ein anfragender Browser eine WSS-Webseite auf, wird Aufbau und Inhalt der Webseite dynamisch erzeugt und als fertige ASP.NET-Webseite an den Browser zurückgeschickt. Der Aufbau der Seite wird aus dem vordefinierten Gitter (hier: Zonen) der WSS-Webseite und den darin platzierten Webparts (siehe 2.4.1 Personalisierung) oder Listen zusammengestellt.

Die Inhalte der Webparts und Listen werden aus der Inhaltsdatenbank ausgelesen und, sofern erforderlich, die Inhalte errechnet. Eine von Microsoft entwickelte, von XML abgeleitete Sprache – die *Collaborative Application Markup Language* (CAML) – ermöglicht die Beeinflussung des Layouts von Webparts und Listen. Die CAML wird zum Zeitpunkt

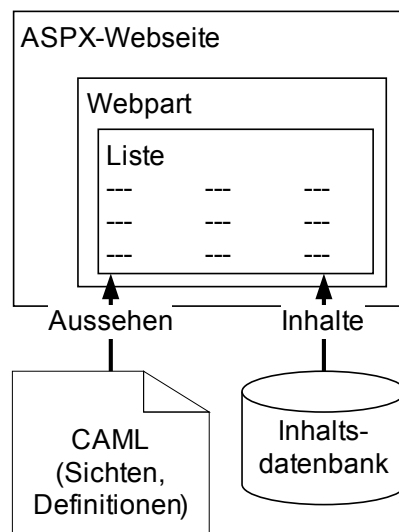


Abbildung 16. Webpart mit Liste auf einer ASPX-Webseite der WSS

des Aufrufs von den WSS interpretiert. Microsoft spricht bei diesem Vorgang vom *Rendern* des Webparts.

¹¹⁸Pro virtuellem Server existiert eine Top-Level-Wurzelseite – zB <http://meinServer:8080>. Weitere Top-Level-Seiten werden darunter im Verzeichnis *sites* angeordnet – zB <http://meinServer:8080/sites/Verkauf>. Diese wiederum können Subsites (gemeinsame Arbeitsbereiche, Dokumentarbeitsbereiche) besitzen.

¹¹⁹Die Zahl der verfügbaren Anschlussnummern liegt bei 65536 (2^{16}). Einige Dienste, auch des Windows Servers, benötigen einen ganz bestimmten oder sogar eine Bandbreite von Anschlussnummern: Der Dienst zum Empfangen von Emails – POP3 – braucht beispielsweise im Normalfall den Anschluss 110.

Alle Elemente der Webseite werden in der Sprache angezeigt, die bei der Einrichtung des gemeinsamen Arbeitsbereich angegeben wurde.¹²⁰ Hierfür sorgen die Vorlagen der WSS-Webseiten, die in sprachabhängigen Verzeichnissen gespeichert sind.¹²¹ Als Datenbanksystem für die Konfigurations- und die Inhaltsdatenbank(en) kann entweder die kostenfrei mitgelieferte Windows Microsoft SQL Desktop Engine (WMSDE) oder der zusätzlich zu erwerbende MS SQL Server 2000 verwendet werden.

Das Veröffentlichen von gemeinsamen Dokumenten im Internet versteht [Bauer 2001:96] als *Webpublishing*. Während das traditionelle *Webpublishing* hauptsächlich gerichtet stattfindet [ebenda] – von dem Administrator einer Webseite zu dem Besucher – erhält dieser Begriff dank heutiger Portale einen bidirektionalen Charakter. Mit der Einführung von Portalen sind nun auch die Benutzer in der Lage, Dokumente zu veröffentlichen. Das WebDAV-Protokoll ist eine der Technologien, die das Veröffentlichen im Internet (*Webpublishing*) ermöglichen. Mit WebDAV können Dokumente von dem Windows Explorer eines Windows-PC direkt in der Dokumentbibliothek eines gemeinsamen Arbeitsbereichs gespeichert werden.¹²²

2.4.4 Webanwendungen

Ein Portal kann Webanwendungen bereitstellen. [Bauer 2001:115] versteht unter Webanwendungen Programme, mit denen der Benutzer über den Webbrowser interagiert und das von dem Application Server ausgeführt wird. Dieser steuert neben der Ausführung der Anwendung auch den Zugriff auf Datenbanken, während der Webserver die Interaktion mit dem Benutzer übernimmt [Bauer 2001:123].

Die Architektur eines Application Server skizziert [Bauer 2001:125] so: Der Application Server besteht im Wesentlichen aus drei Komponenten: Programmobjekte (*Components*) werden in einer Laufzeit ausgeführt (*Container*) und sind über Schnittstellen (*Connectors*) in der Lage, mit anderen Anwendungen, Benutzungsoberflächen oder Datenbanken zu kommunizieren.

Weit verbreitete Application-Server-Systeme verwenden entweder die J2EE-Spezifikation (Java) oder Microsofts .NET-Technologie. Als ein Produkt von Microsoft basieren die WSS auf dem .NET-Framework. Konkret wird hierfür ein Prozess von dem Windows Server gestartet, der die Laufzeit bereitstellt und zum gegebenen Zeitpunkt (zB Aufruf einer Webseite) auf dem Server bereitgestellte Programme ausführt. In dem noch folgenden Teil 3.3.1 *Verfügbare Technologien* werden die Möglichkeiten, Programmcode mit den WSS und dem Application Server auszuführen, vertieft.

¹²⁰Der Vorgang der Anpassung nennt sich Lokalisierung, wobei ein verfügbarer Kulturkreis (gemeint sind Übersetzungen, zB Deutsch, Englisch oder Französisch) ausgewählt wird.

¹²¹Die Vorlage der deutschsprachigen Liste für Ankündigungen findet sich im Verzeichnis:
<Laufwerk:>\Programme\Gemeinsame Dateien\Microsoft Shared\web server extensions\60\TEMPLATE\1031\STS\LISTS\ANNOUNCE

¹²²Mehr zu WebDAV im Entwurfs-Teil dieser Arbeit (siehe WebDAV in 3.3.1)

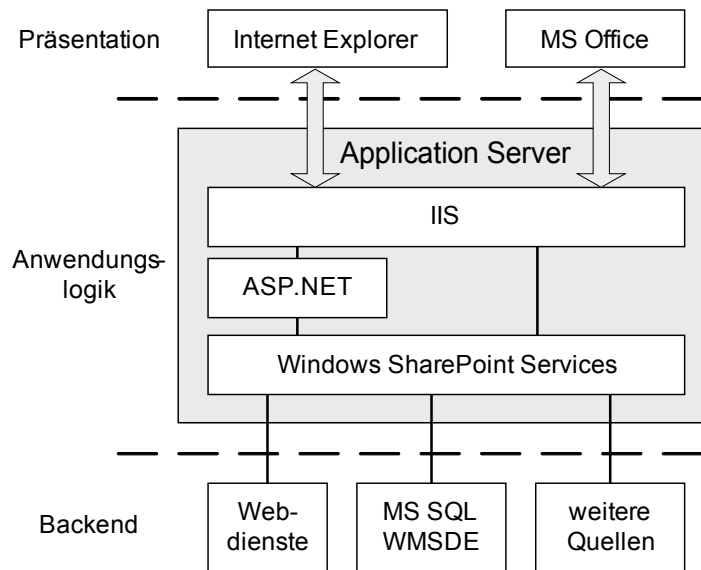


Abbildung 17. Architektur der WSS

2.4.5 Microsoft Office System Integration

Die Integration von Unternehmensanwendungen in [Bauer 2001:127] ist im Verlauf der letzten fünf Jahre in den Begriff der Enterprise Application Integration (EAI) aufgegangen. Ziel der EAI ist die Integration der im Unternehmen existierenden Anwendungssysteme in eine Integrationsplattform.¹²³ Die zwischen der integrierenden Plattform und den Anwendungssystemen geschalteten *Adapter* sorgen für die Vereinheitlichung des Datenaustausches. Die Aufgabe der Portalsoftware ist es, den Benutzern eine homogene Oberfläche für die verschiedenen im Unternehmen eingesetzten Anwendungssysteme bereitzustellen.

Die WSS weisen eine schwache EAI-Funktionalität auf [Hillier 2005:277]. Ist eine Integration von Unternehmensanwendungen dennoch beabsichtigt, muss diese mühevoll selbst entwickelt werden.

¹²³ <http://de.wikipedia.org/wiki/Enterprise_Application_Integration> (30.3.2006)

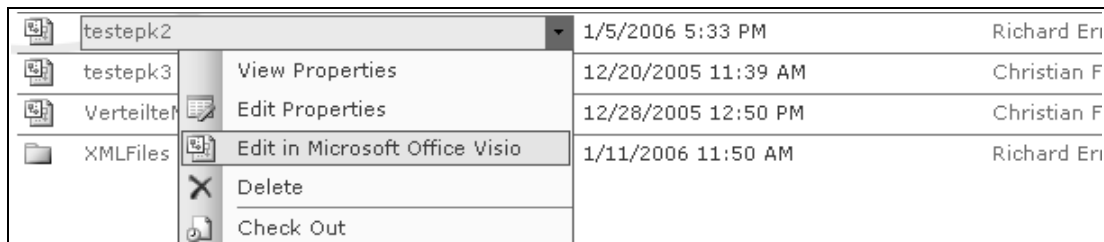


Abbildung 18. Kontextmenü einer MS Visio-Datei in einer Dokumentbibliothek.

Im Gegensatz dazu wurde eine starke Integration mit MS-Office-Produkten unter dem Begriff *Microsoft Office System Integration* erreicht. Microsoft spricht hier von einer engen *Verzahnung* [Dowler 2003:1] zwischen den clientseitigen MS-Office-Anwendungen und den WSS. So lässt sich ein in einem gemeinsamen Arbeitsbereich gespeichertes Office-Dokument über dessen Kontextmenü öffnen.¹²⁴ Dabei wird die mit dem Dokumententyp verknüpfte MS-Office-Anwendung auf dem anfragenden Client gestartet und das Dokument zum weiteren Bearbeiten geöffnet. In Abbildung 18 ist dargestellt, wie sich das Kontextmenü bei einer MS-Office-Visio-Datei verhält.

Wesentlich tiefer ist die *Microsoft Office System Integration* in den clientseitigen MS-Office-Produkten eingebaut. Hier werden die von den WSS angebotenen Dienste über eine zusätzlich eingeblendete Bedienungsfläche¹²⁵, den *Freigegebenen Arbeitsbereich*, direkt nutzbar gemacht. Der freigegebene Arbeitsbereich (siehe Abbildung 19) wird synonym zum Begriff des gemeinsamen Arbeitsbereichs verwendet. Er wird automatisch eingeblendet, sobald die MS-Office-Anwendung festgestellt hat, dass das soeben geöffnete Dokument aus einem gemeinsamen Arbeitsbereich stammt. Für die



Abbildung 19. Freigegebener Arbeitsbereich, eingeblendet in MS Visio

Benutzung WSS-spezifischer Funktionen zur Verwaltung von Dokumenten wird dank

¹²⁴Die WSS verfügen über die Informationen, welcher Dokumententyp zu welcher Office-Anwendung gehört.

¹²⁵Als *Sidebar* bezeichnet, da sie sich frei an einem der Seitenränder von MS Visio andocken lässt.

der eingeblendeten Oberfläche kein Browser mehr benötigt. Der Zugriff und dessen Sperrung, die Speicherung, Löschung und die Versionierung gemeinsamer Dokumente erfolgt über die Bedienelemente des Freigegebenen Arbeitsbereichs. Neben den Funktionen zur Verwaltung der Dokumente kann auf die Aufgabenliste des gemeinsamen Arbeitsbereichs zugegriffen oder die Kommunikation mit den anderen Gruppenmitgliedern initialisiert werden.¹²⁶ Des Weiteren ist die Verwaltung von Benutzern des gemeinsamen Arbeitsbereichs möglich. Eine Übersicht der in den einzelnen MS Office-Anwendungen verfügbaren Funktionen für die Arbeit mit einem gemeinsamen Arbeitsbereich ist in Tafel 2 zu finden.

Funktion	clientseitige MS Office-Anwendung
Öffnen und Speichern über das Menü <i>Datei</i>	Excel, FrontPage, InfoPath, Microsoft Project, Onenote, Outlook, PowerPoint, Publisher, Visio, Word
Aufgabenbereich <i>Freigegebener Arbeitsbereich</i>	Excel, Microsoft Project, OneNote, PowerPoint, Visio, Word
Erstellen einer freigegebenen Anlage	Outlook
Synchronisieren von Kalender- und Kontaktlisten	Outlook
Dokumentaktualisierungen für freigegebene Anlagen	Excel, PowerPoint, Visio, Word
Automatisches Sammeln von Metadaten	Word, Excel, PowerPoint
Ein- und Auschecken von Dokumenten	Word, Excel, PowerPoint, Visio
Versionskontrolle	Word, Excel, PowerPoint
Speichern von Inlinediskussionen auf dem Server	Word, Excel, PowerPoint
Importieren von Sharepoint-Listen als Daten	Excel, Access
Exportieren von Daten als SharePoint-Liste	Excel, Access

Tafel 2. *Quelle: [Dowler 2003].*

Vereinfacht wird die webbasierte Zusammenarbeit an Dokumenten dann, wenn die Dokumente nicht heruntergeladen, bearbeitet, auf dem lokalen Rechner gesichert und anschließend wieder auf den Server hochgeladen werden müssen. Im Zusammenspiel zwischen den WSS und MS-Office-Produkten können Dokumente *direkt* in dem gemeinsamen Arbeitsbereich bearbeitet werden. Die Speicherung des Dokuments erfolgt immer

¹²⁶ Es werden die Emailadressen der Team-Mitglieder angezeigt sowie deren Online-Status, sofern sie den Microsoft-eigenen Instant Messenger verwenden.

auf der Dokumentbibliothek und nicht auf dem lokalen Rechner.¹²⁷ Das heißt, es existiert – zumindest für den Anwender – keine Kopie des Dokuments auf dem zugreifenden Client-PC.

Eine vollständige Ausführung aller Integrationsszenarien ist an dieser Stelle nicht möglich. Darum sei auf den „Integrationsleitfaden für Microsoft Office 2003 und Windows SharePoint Services“ von Anthony Dowler, Juli 2003, verwiesen. Der Leitfaden ist unter <http://www.microsoft.com/germany/technet/datenbank/articles/600279.msp> erreichbar (letzter Zugriff: 14.3.2006).

Bei allen Integrationsvorhaben ist stets zu berücksichtigen, dass für einen produktiven Einsatz der WSS im Zusammenspiel mit anderen MS-Office-Anwendungen deren Einbettung in eine vernetzte Umgebung nötig ist. Die Infrastruktur muss über zeitgemäße Hard- und Software verfügen, will man die gemeinsame Arbeit mit den WSS ohne Nachteile gewährleisten. Dies betrifft hauptsächlich die Datenübertragungsgeschwindigkeit im lokalen Netzwerk, aber auch der Verbindungsgeschwindigkeit zu dem Internet. Versuchsaufbauten im September 2005 in einem Labor der FHB zeigten, dass unzureichende Ressourcen in der Infrastruktur die Arbeit mit den WSS unmöglich machten.¹²⁸

2.4.6 Unterstützung von Zusammenarbeit

Sollte in den 80er und 90er Jahren eine Technologie eingeordnet werden, deren Zweck darin lag, die Zusammenarbeit von Gruppen informationstechnisch zu unterstützen, wurden häufig die Begriffe CSCW- und Groupware-Systeme gebraucht. [Munkvold 2003:7f] fasst die Gesamtheit derartiger Systeme unter *Collaboration Technologies* zusammen. Hierunter fallen auch Dokumenten- und Wissensmanagementsysteme. Auf Grund des ständig wachsenden Umfangs an Unterstützungssystemen, ist diese Wortschöpfung (also *Collaboration Technology*) geeignet, eine übergeordnete Kategorie zu bilden.

Erste Bestrebungen, Technologien zur Unterstützung von Zusammenarbeit genauer zu klassifizieren, wurden 1987 unternommen. Eines der ersten Ergebnisse war die Raum-Zeit-Matrix, in der Systeme nach ihrer Ausprägung hinsichtlich der gleichzeitig (*synchronen*) oder zeitlich versetzt (*asynchronen*) stattfindenden Kommunikation bzw der am selben oder an verschiedenen Orten stattfindenden Kommunikation eingeordnet werden [Munkvold 2003:9].

¹²⁷Tatsächlich sichert MS Visio die Verbindungsdaten der WSS-Sitzung in der Windows-Registrierungsdatenbank und speichert durchaus das Dokument auf der Festplatte des Windows-PC zwischen. Auf die Sitzungsdaten wird bei dem Vorgang der Speicherung zurückgegriffen. Für den Anwender ist dieser Vorgang transparent – er hat den Eindruck immer *auf* dem Server zu arbeiten.

¹²⁸Das betreffende Labor war mit mehrere Jahre alten Computer-Arbeitsplätzen ausgestattet.

Raum/Zeit	gleiche Zeit (synchron)	verschiedene Zeit (asynchron)
gleicher Ort	Sitzung mit persönlicher Anwesenheit	<i>schwarzes Brett</i>
verschiedener Ort	Video-/Audio-Konferenz, Instant Messenger, Telefon	Email, Newsgroups

Tafel 3. Raum-Zeit-Matrix mit Beispielen

Im Jahre 1997 veröffentlichten Grudin und Poltrock in dem von M. Zelkowitz herausgegebenen Sammelwerk *Advances in Computers* ihre Erkenntnisse über die unzureichende Aussagekraft der Raum-Zeit-Matrix. Sie schlugen ein neues Klassifizierungssystem vor. In diesem lässt sich jedes CSCW-System seiner funktionalen Ausprägung folgend im Bereich der Kommunikations-, der Koordinations- sowie Kooperationsunterstützung differenzieren – das 3-K-Modell.¹²⁹ Den *Kommunikationstechnologien* werden Email, Instant Messaging, Audio- und Videokonferenzsysteme zugeordnet, wohingegen Dokumentenmanagementsysteme, webbasierte Gruppenräume, elektronische Foren und Mehrbenutzereditoren¹³⁰ zu den *Kooperationstechnologien* gezählt werden. Technologien zur Unterstützung der *Koordination* haben ihren funktionalen Schwerpunkt im Workflowmanagement sowie der Pflege gemeinsamer Kalender.

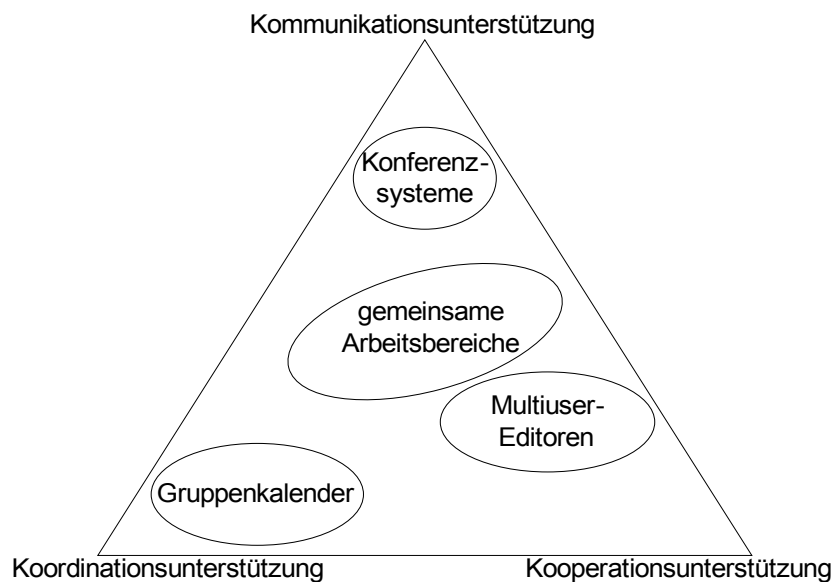


Abbildung 20. Das 3-K-Modell. Quelle: [Borghoff; Schlichter 1998:128]

Da es kaum noch bedeutende Systeme der *Collaboration Technology* auf dem Markt gibt, die nicht mehrere Aspekte des 3-K-Modells unterstützen, können sie alle mehr oder weniger als *integrated products* bezeichnet werden [Munkvold 2003:10]. [Munkvold 2003:ebenda] un-

¹²⁹ Nach S. Teufel, C. Sauter, T. Mühlherr, K. Bauknecht. 1995. *Computergestützte Gruppenarbeit*. Bonn: Addison-Wesley.

¹³⁰ <<http://de.wikipedia.org/wiki/CSCW>> (22.2.2006)

terteilt die integrierten Produkte weiterführend in *collaboration product suits*¹³¹, *integrated team support technologies*¹³² sowie *e-learning technologies*¹³³. Als dominierende Beispiele für *collaboration product suits* führt Munkvold Lotus Notes/Domino sowie Microsoft Exchange/Outlook an.

Um die WSS in dem Klassifizierungssystem von [Munkvold 2003] einordnen zu können, erscheint eine Betrachtung der von ihm vorgeschlagenen *integrated team support technologies* sinnvoll [Munkvold 2003:24]. Systeme dieser Kategorie unterstützen vornehmlich die Arbeit im Team und bieten diverse synchrone und asynchrone Kommunikationsdienste in verschiedenen Ausprägungsstufen an. Eine *integrated team support technology* kann entweder die asynchrone oder die synchrone Kommunikation verstärkt unterstützen. Werden im Wesentlichen asynchrone Kommunikationsdienste mit gemeinsamer Dokumentenverwaltung, Terminplanung sowie Diskussionsgruppen angeboten, dann spricht [Munkvold 2003:24] von *teamrooms*. Dagegen werden unter *desktop conferencing systems* Systeme mit synchronen Kommunikationsdiensten, wie Konferenz-Systeme oder Chaträume verstanden [ebenda]. Vergleicht man die Dienste der von Munkvold vorgeschlagenen *teamrooms* mit denen der WSS, lässt sich zwischen diesen beiden – im Gegensatz zu allen anderen Kategorien der *Collaboration Technologies* – die größtmögliche Übereinstimmung herleiten.

¹³¹ Vereinen Funktionen einer Groupware mit denen eines Dokumentenmanagementsystems und haben die Möglichkeit Workflows zu unterstützen.

¹³² Produkte dieser Art unterstützen die Arbeit von Teams indem sie eine Umgebung für die Gruppenarbeit aus synchronen und asynchronen Diensten bereitstellen (sinngemäß übersetzt aus [Munkvold 2003:24]).

¹³³ E-Learning-Systeme ermöglichen verschiedene Lehr- und Lernformen, bei denen Lehrende mit Schülern und Schüler mit anderen Schülern kommunizieren können. [Munkvold 2003:25] unterteilt die verfügbaren Systeme in Systeme zur Unterstützung des synchronen gemeinsamen Lernens, des asynchronen gemeinsamen Lernens und des eigenständigen Lernens.

3 Idee, Anforderungen und Entwurf

3.1 Windows SharePoint Services als Plattform für die verteilte Modellierung

In [Vom Brocke 2003b:3] wurde im Zusammenhang mit dem technologiebezogenen Aspekt der Bedarf an einer technischen Plattform zur Unterstützung der verteilten Modellierung formuliert. Für die Eignung der WSS als Plattform zur verteilten Modellierung mit SemTalk kommen im Wesentlichen drei Gründe in Betracht.

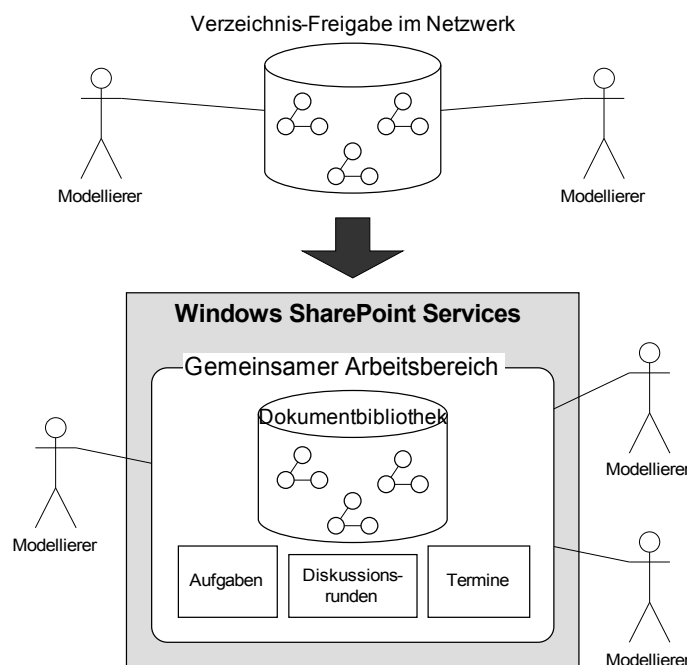


Abbildung 21. Die WSS als Plattform für die verteilte Modellierung

(1) Umfangreiche Technologie. Bei den WSS handelt es sich um eine sehr komplexe Software. Ihren Namen verdankt sie der um noch einige Funktionen umfassenderen Servervariante – dem *Microsoft Office SharePoint Portal Server*. Obwohl die WSS nicht über den Funktionsumfang des SharePoint Portal Server (SPS) verfügen, muss erwähnt werden, dass der SPS technisch auf den WSS basiert und diese lediglich erweitert. Tafel 4 stellt die Funktionen beider Produkte gegenüber.

Die bei der Realisierung der WSS eingesetzten Technologien lassen eine gute Erweiterung bzw. Nutzung vorhandener Funktionen zu. Grund dafür ist ihre Kopplung an den Windows Server 2003 und dessen Möglichkeiten als Application Server. Die Technologien werden im Teil 3.3.1 *Verfügbare Technologien* ausführlich dargestellt.

Funktion	Windows SharePoint Services	SharePoint Portal Server 2003
Benachrichtigungen	Ja	Ja
Anpassung via Browser	Ja	Ja
Diskussionsgruppen	Ja	Ja
Dokumentbibliotheken	Ja	Ja
Dokumentarbeitsbereiche	Ja	Ja
Besprechungsarbeitsbereiche	Ja	Ja
Listen	Ja	Ja
Integration von Microsoft BizTalk	Nein	Ja
Integration von Microsoft FrontPage 2003	Ja	Ja
Integration von Microsoft InfoPath 2003	Ja	Ja
Umfrage	Ja	Ja
Vorlagen	Ja	Ja
Webpartseiten	Ja	Ja
Automatische Kategorisierung	Nein	Ja
Leserschaft	Nein	Ja
Themenbereiche	Nein	Ja
Nachrichten	Nein	Ja
Persönliche Seiten	Nein	Ja
Gemeinsame Dienste	Nein	Ja
Single Sign-On	Nein	Ja
Webseitenverzeichnis	Nein	Ja
Benutzerprofile	Nein	Ja

Tafel 4. Quelle: [Microsoft 2003a]

(2) Microsoft Office System Integration. Die WSS und MS Visio, auf dem SemTalk basiert, gehören beide zu der Microsoft Office-Produktpalette. Dank der unter *Microsoft Office System Integration* zusammengefassten clientseitigen Erweiterungen der Office-Produkte ist das kooperative Arbeiten mit den serverbasierten WSS aus den gewohnten Office-Produkten möglich.¹³⁴ Das bedeutet, dass sich mit der Nutzung der WSS automatisch Mehrwerte ergeben, die bei Verwendung einer anderen Anwendung unter Umständen nachgebildet werden müssten.

(3) Geringer Investitionsaufwand. Die geringen Anschaffungskosten machen das dritte Auswahlkriterium aus. Denn nur, wenn die Kosten einer Umgebung für verteilte Modellierung in Grenzen gehalten werden können, ist mit der Durchdringung der verteil-

¹³⁴ <<http://www.microsoft.com/germany/office/wss/leistungsmerkmale/default.mspx>> (8.1.2006)

ten Modellierung bei den Anwendern zu rechnen. Des Weiteren wird bei Einführung einer Standardsoftware von einem „schnelleren ROI“¹³⁵ gesprochen.¹³⁶

3.2 Anforderungen an eine Plattform zur verteilten Modellierung

„Das Ziel der Analyse ist es, die Wünsche und Anforderungen [...] an ein neues Softwaresystem zu ermitteln und zu beschreiben [...]. Es ist wichtig, dass bei der Modellbildung in der (System-)Analyse alle Aspekte der Implementierung bewusst ausgeklammert werden.“ [Balzert 2005:9]

Die Anforderungen teilen sich in zwei Bereiche auf: Jene, welche von den WSS hinsichtlich ihrer Aufgabe als Plattform zur gemeinsamen Arbeit erfüllt werden müssen und diejenigen, die sich auf das Zusammenspiel zwischen SemTalk und den WSS hinsichtlich verteilter Modelle beziehen. Für den ersten der beiden Bereiche werden die Anforderungen der im Grundlagenteil vorgestellten Arbeiten mit den noch zu zeigenden eigenen Untersuchungsergebnissen zusammengefasst. Diese, an eine Plattform zur Zusammenarbeit zu stellenden, Anforderungen werden mit dem standardmäßig verfügbaren Funktionsumfang der WSS verglichen. Die dabei nicht abgedeckten Anforderungen bilden die Grundlage für die spätere Anpassung und Erweiterung der WSS. Nach der Spezifikation der umzusetzenden Anforderungen folgt der Entwurf.

Damit die bereits vorhandenen Techniken zur verteilten Modellierung mit SemTalk in der Kombination mit den WSS unterstützt werden, sind die Anforderungen an die Anpassung von SemTalk zu ermitteln und zu spezifizieren. Denn werden technische Lösungen geplant, die mit den WSS zusammenwirken sollen, müssen die sich daraus ergebenden Vorgaben eingehalten werden.

Alle in den Phasen der Anforderungsdefinition¹³⁷ und des Entwurfs¹³⁸ abgebildeten Modelle folgen der Notation der UML. Ihr liegt das Paradigma der Objektorientierten Systementwicklung zugrunde.¹³⁹ In der UML werden zwei grundlegende Konzepte unterschieden – das statische und das dynamische Modell. Während das statische Modell die dem Entwicklungsziel zugrundeliegenden Klassen, deren Beziehungen untereinander sowie ihre Gruppierung definiert, dient das dynamische Modell der Darstellung der Kommunikation, der Aktivitäten und des Verhaltens der aus den Klassen abgeleiteten Objekte während der Laufzeit des Programms [Balzert 2005:11].

¹³⁵ Der Return-on-Investment (ROI) ist ein häufig gebrauchtes Kennzahlensystem in der Investitionsrechnung. Es handelt sich um den Quotienten aus Periodengewinn geteilt durch das Gesamtkapital, und gibt an, wie schnell das investierte Kapital in das Unternehmen zurückfließt.

¹³⁶ <<http://de.wikipedia.org/wiki/Standardsoftware>> (23.1.2006)

¹³⁷ Die Objektorientierte Analyse (OOA)

¹³⁸ Das Objektorientierte Design (OOD)

¹³⁹ Das Konzept der Objektorientierung beruht darauf, dass alles ein Objekt ist. Ein Objekt ist immer ein reelles Exemplar einer Klasse von Objekten. Softwaresysteme bestehen aus miteinander kommunizierenden Objekten.

Die der Objektorientierten Analyse (OOA) zuzuordnenden UML-Anwendungsfall-Diagramme dienen der Beschreibung der von dem System zu lösenden Aufgaben auf einem sehr hohen Abstraktionsniveau. In der Phase der Anforderungsanalyse findet keine Berücksichtigung der technischen Umsetzung statt. Der Fokus liegt auf der Ermittlung und Spezifikation der Leistungen, die von dem umzusetzenden System zu erfüllen sind.

UML-Klassen-Diagramme dienen in der anschließenden Entwurfsphase (Objektorientiertes Design, OOD) zur Darstellung des statischen Verhaltens der umzusetzenden Komponenten. Der konkrete Ablauf der Kommunikation zwischen den Objekten wird in den UML-Sequenzdiagrammen abgebildet.¹⁴⁰

3.2.1 Anforderungen aus bisherigen Untersuchungen

Aus den im Grundlagen-Teil dieser Arbeit vorgestellten Untersuchungen¹⁴¹ ergeben sich die Kriterien, die für den Erfolg einer Plattform zur Unterstützung der verteilten Modellierung maßgeblich sind. Mögliche Hürden bei der später erwünschten Akzeptanz können, sofern diese Anforderungen von der Plattform unterstützt werden, schon vor deren eigentlichen Einführung genommen werden.

Die Gliederung der im Folgenden aufgeführten Anforderungen orientiert sich an den Mustern und Gemeinsamkeiten, die in den verschiedenen Vorgehensmodellen festgestellt werden konnten (siehe 2.2.4). Diese Einordnung ist insofern sinnvoll, als der Bedarf einer Funktion oder eines Werkzeugs besonders dann nachvollzogen werden kann, wenn es gebraucht wird. Zunächst werden jedoch einige Anforderungen definiert, die sich nicht speziellen Phasen zuordnen lassen oder die über die gesamte Dauer eines verteilten Modellierungsprojekts benötigt werden.

Projektübergreifende Anforderungen

(1) Client-Server-Architektur. Die verteilte Modellierung kann nur mit einer Client-Server-Architektur umgesetzt werden (siehe Kapitel 2 Grundlagen). Für ihre Realisierung sind Internettechnologien einzusetzen, die den weltweiten Zugriff auf die Modelle zu jeder Tageszeit ermöglichen.

(2) Synchroner Kommunikation. Wird zur selben Zeit aber an unterschiedlichen Orten an Modellen gearbeitet, müssen die dabei stattfindenden Diskurse und Abstimmungsprozesse unterstützt werden. Üblicherweise werden dafür sogenannte *Instant Messenger*¹⁴²,

¹⁴⁰ Auf eine Einführung in die UML wird an dieser Stelle verzichtet. Es sei auf die einschlägige Literatur verwiesen. Zum Beispiel [Balzert 2005] oder Bernd Oestereich. 2004. *Objektorientierte Softwareentwicklung. Analyse und Design mit UML 2.0*. 7. Aufl. Heidelberg: Oldenbourg.

¹⁴¹ [Sure ua 2002], [Ceccaroni; Kendall 2002], [Simon 2004], [Weichhardt 2005], [Sure ua 2005] sowie [Hodgson 2005].

¹⁴² Bedeutende Vertreter sind MSN Messenger von Microsoft, der AOL Instant Messenger (AIM) und ICQ, beide von AOL. Eine sehr gute Übersicht der Funktionen findet man in dem Artikel zu Instant Messenging in der deutschen Wikipedia (<http://de.wikipedia.org/wiki/Instant_Messenger> (8.3.2006))

Software für Internettelefonie¹⁴³ oder Audio-Konferenzen (*Multi-Person Audio Chat*) eingesetzt. Video-Konferenzen bleiben auf Grund der immer noch unzureichenden Netzwerkkapazitäten die Seltenheit.

(3) Asynchrone Kommunikation. Neben dem Email-Dienst, der die direkte aber zeitlich versetzte Kommunikation unterstützt, werden Diskussionsrunden¹⁴⁴ und Foren benötigt. Letztere dienen dem zeitlich versetzten und indirekten Diskurs zwischen den Beteiligten eines Modellierungsprojekts. Auch diese Funktionen werden phasenübergreifend benötigt.

Anforderungen nach Projektphasen

(1) Die Anforderungsspezifikation. Die Projektleitung braucht für die Entscheidungsfindung *Besprechungsräume* und *Diskussionsrunden* [Sure ua 2002:227]. Um das Projekt planen zu können, sind *Meilensteine*, *Termine* und die *Aufgaben* der Modellierer oder Ontologiekonstrukteure festzulegen. Zu bestimmten Themen sind allen Beteiligten des Projekts Mitteilungen zu machen. Hierfür sollte eine Funktion vorhanden sein, welche die *Bekanntmachungen* vorbereitet und anschließend allen Empfängern mitteilt.

In der Phase der Anforderungsspezifikation werden einige Dokumente erzeugt.¹⁴⁵ Diese sind nicht von dem selben Dokumententyp, wie die Modelle von SemTalk. Darum muss die Plattform in der Lage sein, *heterogene Dateitypen*¹⁴⁶ zu verwalten. Des Weiteren sind diese Dokumente zu *versionieren*, um bei einem Fehler einen früheren Zustand wiederherstellen zu können, und mit *Sperrmechanismen* vor Inkonsistenz zu schützen.¹⁴⁷

Die Projektbeteiligten müssen der Plattform als Benutzer hinzugefügt werden. Diese müssen in der Lage sein, auf die Produkte dieser Phase zugreifen zu können.¹⁴⁸ Eine *Benutzerverwaltung*, die den Umfang der Rechte des Einzelnen bereitstellt und den *Zugriff auf Ebene der Dokumente* steuert, ist also zwingend erforderlich.

(2) Erste Version der gemeinsamen Ontologie. Die Organisationseinheit, die für das gemeinsam zu nutzende Modell verantwortlich zeichnet, hat über ausreichend *Rechte* zu verfügen, das Modell auf der Plattform zu bearbeiten und anschließend zu veröffentlichen.¹⁴⁹ Für die gemeinsame Arbeit an dem gemeinsamen Modell benötigt die Organisationseinheit die gleichen Mechanismen wie jene, die in der ersten Phase die Projektleitung

¹⁴³ Hier wird Skype häufig verwendet.

¹⁴⁴ Zum Beispiel das Usenet mit den Newsgroups.

¹⁴⁵ Zumeist Text-Dokumente wie die Anforderungsspezifikation (siehe [Sure ua 2002]).

¹⁴⁶ Unterschiedliche Dateitypen aus unterschiedlichen Anwendungen.

¹⁴⁷ Arbeiten zufällig mehrere zur Bearbeitung berechnete Personen an einem Dokument gleichzeitig, sind nur dessen Inhalte in dem Dokument sichtbar, der als letztes das Dokument gesichert hat. Es muss also das exklusive Bearbeiten eines Dokuments gewährleistet sein.

¹⁴⁸ Das sind die zentralen Dokumente der Anforderungsspezifikation.

¹⁴⁹ Dieser Vorgang ist wie eine Freigabe zu verstehen, bei dem alle Nutzer des gemeinsamen Modells dieses nun öffnen, aber nicht bearbeiten und speichern können.

bezüglich der zentralen Dokumente braucht.¹⁵⁰ Eine *Integration von SemTalk* in der Plattform ist hilfreich, so dass sich Modell-Dateien direkt aus der gemeinsamen Plattform heraus öffnen lassen. Gleiches gilt für das Speichern des Modells: Dieses sollte immer direkt in der Plattform geschehen.

[Sure ua 2002:228] betont, dass es bei der verteilten Modellierung wichtig ist, Designentscheidungen nachvollziehbar zu machen. Dies geschieht mit *Kommentaren* innerhalb des Modells oder an der Modell-Datei, jeweils auf deren Version bezogen. Die Entscheidungsfindung während der Modellierung kann mit *Diskussionsrunden* unterstützt werden. *Bekanntmachungen* werden benötigt, um die Projektbeteiligten von der Bereitstellung des gemeinsamen Modells zu informieren.

(3) Import des gemeinsamen Modells. Die Ontologie-Nutzer, Ontologiekonstrukteure und Modellierer haben *Leserechte* auf das gemeinsame Modell zu besitzen. Des Weiteren muss der verwendete Editor über eine Funktion zum *Import* von Modellen verfügen.

(4) Nutzungsphase. Zur Erledigung ihrer Aufgaben ist den Modellierern der volle Zugriff auf ihre eigene Modell-Datei zu gewähren. Daneben brauchen sie lesende Rechte für die Modelle anderer, um darin befindliche Inhalte in ihren eigenen Modellen nutzen zu können.¹⁵¹ Gleiches gilt für die zentralen Dokumente, die allen zur Verfügung stehen müssen, aber nur die autorisierte Organisationseinheit ändern darf.

Konsistenz über mehrere Modelle hinweg kann erreicht werden, wenn SemTalk über alle Objekte in externen Modellen informiert ist. Dazu müssen sämtliche Objekte des Modellierungsprojektes an einer zentralen Stelle bekanntgegeben werden.¹⁵² Diese Übersicht ist zu verwenden, wenn externe Objekte gleichen Namens gefunden werden sollen.

Sind fremde Modelle, auf die sich ein Modellierer in seinem Modell bezieht, verändert worden, muss der Modellierer darüber mit Hilfe einer *Änderungsmitteilung* in Kenntnis gesetzt werden. Ontologiekonstrukteure bzw Modellierer fremder Modelle müssen auch über veränderte Referenzobjekte informiert werden, auf die sie sich in ihren Modellen beziehen. Das ist für den Erhalt der Konsistenz in ihren Modellen nötig.¹⁵³

Für die Planung der eigenen Aufgabenerfüllung müssen die Modellierer ihre Termine planen und – sofern in Teams organisiert – Aufgaben den einzelnen Teammitgliedern zuordnen können.

[Sure ua 2005:90] schlägt ein *Argumentation Framework* für den Fall vor, dass die Modellierer Objekte identifiziert haben, die ihrer Ansicht nach in dem gemeinsamen Modell eingearbeitet werden sollen. Während in kleineren Projekten der verteilten

¹⁵⁰ Das heißt, Versionsverwaltung und Sperrung der Modell-Datei während der Bearbeitung und Vergabe von Rechten auf die Datei.

¹⁵¹ Nutzung der Import-Funktion oder Erzeugung externer Referenzen in SemTalk (siehe 2.1.1(2)).

¹⁵² Diese Aufgabe hat bisher der Modell-Index erfüllt (siehe 2.1.1(4)).

¹⁵³ Ein Objekt, das sich auf ein externes Objekt mittels einer externen Referenz bezieht, muss aktualisiert werden. Ansonsten reflektiert es nicht den aktuellen Zustand des externen Objekts.

Modellierung eine einfache Mitteilung an die Organisationseinheit als ausreichend angenommen werden kann, könnte ein auf Formularen basiertes *Argumentation Framework* in größeren Projekten von Bedeutung sein.

(5) Qualitätssicherung. Die Sitzungen der autorisierten Organisationseinheit mit dem Ziel, die Änderungsanträge der Modellierer zu besprechen, finden bestenfalls in Anwesenheit aller Mitglieder statt. Für deren Vorbereitung wären von der Plattform bereitgestellte *separate Besprechungsräume* hilfreich. Darin eingebundene *Diskussionsrunden* könnten die Mitglieder im Vorfeld der eigentlichen Sitzung mit den zu diskutierenden Anträgen vertraut machen.

Finden die Sitzungen der Organisationseinheit zu einem Zeitpunkt statt aber befinden sich die beteiligten Mitglieder an verschiedenen Orten, wird ein *Konferenz-System* benötigt (siehe 2.4.6). Hier schlägt [Sure ua 2005:97f] einen virtuellen Chat für die Entscheidungsfindung vor, der moderiert wird, ein festes Thema hat und in einem definierten Zeitraum stattfindet.¹⁵⁴

Um die Qualität der Teil-Modelle zu prüfen und die betreffenden Modellierer von Verbesserungsvorschlägen zu informieren, werden die schon vorgestellten Techniken verwendet.¹⁵⁵ Die Report-Funktion von SemTalk (*SemTalk Report Editor*) kann hier von Nutzen sein, da mit ihr Auswertungen über das aktuelle, aber auch externe Modelle erzeugt werden können.

3.2.2 Eigene Untersuchungen

Neben den verschiedenen verfügbaren Schriften und Präsentationen, die einen praxisbezogenen Hintergrund haben und demzufolge als Erfahrungsberichte eingeordnet werden können, wurden im Rahmen dieser Arbeit eigene Untersuchungen in Form von Interviews und Umfragen durchgeführt. Bei der im Juli 2005 per Fragebogen durchgeführten Umfrage nahmen acht Studenten des 2. Semesters im Studiengang Wirtschaftsinformatik an der Fachhochschule Brandenburg teil. Diese führten in der Lehrveranstaltung Systemanalyse II ein Modellierungsprojekt durch, bei dem unter anderem der Aspekt der verteilten Modellierung zum Tragen kam. Die ebenfalls im Juli 2005 fernmündlich durchgeführten Interviews mit zwei Mitarbeitern der Bayer MaterialScience AG¹⁵⁶ hatten zum Ziel, Anforderungen von Personen einzuholen, die berufsbedingt mit der Modellierung von Geschäftsprozessen zu tun haben.

¹⁵⁴Sure musste jedoch auch Probleme einräumen. So wurde während der Diskussion von dem Thema abgewichen und es bildeten sich parallele Diskussionen heraus. Über die eigentlich zu diskutierenden Anträge wurde kaum entschieden. In späteren Sitzungen formalisierten [Sure ua 2005:97f] den Ablauf, was zu besseren Ergebnissen führte. Eine zu prüfende Lösung wäre die 2002/03 an der FH Brandenburg getestete Konferenz-Software OpenSpace-Online <<http://www.openspace-online.com/>>.

¹⁵⁵Rechte zum Öffnen der Teil-Modelle für Mitglieder der autorisierten, kompetenten Organisationseinheit, Integration SemTalk's in der Plattform, Aufgabenlisten, Email und Instant Messenger.

¹⁵⁶Bayer MaterialScience AG ist ein Teilkonzern der Bayer AG. Im Internet erreichbar unter: <<http://www.bayermaterialscience.de>>

Sowohl bei den Interviews als auch bei der Umfrage kam ein Fragebogen zum Einsatz, der sich an den Vorschlägen von [Krallmann ua 1999:71] und [Gierhake 2000:301ff] zur Feststellung des Ist-Standes orientierte. Er kann im Anhang III eingesehen werden. Bei der Erstellung des Interviewleitfadens lag ein besonderes Augenmerk auf den Erfahrungen der betreffenden Person bei der verteilten Modellierung mit SemTalk. Die Erfahrungen lassen sich auch hier wieder in die vier Aspekte Organisation, Modell, Methode sowie Technologie unterteilen. Entsprechend der zu erwartenden emotional beeinflussten Antworten wurde versucht, bestimmte Fragen möglichst indirekt zu erfragen [Krallmann ua 1999:65]. Die Antworten auf die Frage der erlebten Schwierigkeiten bei der verteilten Modellierung stehen exemplarisch für diese Problematik bei der Datenerhebung zur Istanalyse.

Umfrage bei Studenten eines Projektes zur verteilten Modellierung

Ziel des Modellierungsprojektes der Lehrveranstaltung war das Erlernen von Techniken und Methoden der Geschäftsprozessmodellierung unter Leitung von Prof Wikarski. Gegenstand der Modellierung war die elektronische Gesundheitskarte (eGK), die innerhalb der nächsten Jahre in der Bundesrepublik Deutschland flächendeckend eingeführt werden soll. Die für die Modellierung benötigten Informationen erhielten die Studenten in Form von bereits modellierten Spezifikationen. Diese wurden von den Fraunhofer-Instituten für Software- und Systemtechnik (ISST), Arbeitswirtschaft und Organisations (IAO) sowie Sichere Informationstechnologie (SIT) in der Unified Modelling Language (UML) modelliert und der FH Brandenburg für die weitere Bearbeitung zur Verfügung gestellt. Das Modellierungsprojekt hatte demnach den Charakter eines Reengineering-Vorhabens.

Die Modellierung erfolgte mit SemTalk in der Version 2. Die Studenten des Studiengangs waren in drei Gruppen aufgeteilt und jede dieser Gruppen war in Teams zu zwei bis drei Studenten gegliedert. Ein Team aus jeder Gruppe hatte die Aufgabe, die Aktivitäten seiner Gruppe zu koordinieren – das *Koordinierungsteam*. Ein vorab bestimmter Teamleiter führte sein jeweiliges Modelliererteam an. Heraus kam eine Stablinienorganisation über das gesamte Modellierungsprojekt. Das angewandte Vorgehen entsprach in den Grundzügen dem von [Simon 2004] entwickelten Vorgehen. Aber es kann festgestellt werden, dass die Rolle der Qualitätsbeauftragten fehlte.

Die Umfrage fand nach Abschluss des Modellierungsvorhabens statt und sollte die von den Studenten gemachten Erfahrungen bezüglich der Zusammenarbeit festhalten. An der Umfrage nahmen ein modellierender Student, zwei Teamleiter der Modellierungsteams und vier Beteiligte der Koordinierungsteams teil.

Zu Beginn hatte das Koordinierungsteam generelle Vorgaben, wie Meilensteine und den einzuhaltenden Design- und Konventionsregeln herauszuarbeiten. Die Koordinierungsteams trafen sich wöchentlich, um die Aufgaben für die Modellierungsteams

zu besprechen und jene davon in Kenntnis zu setzen. Die Aufgaben beinhalteten auch eine Terminvorgabe, die im Abgleich mit dem Projektmanagement-Werkzeug BCS stattfand. Jedes Team hatte, unabhängig davon, ob es sich um Modellierer oder Koordinierer handelte, sein Projekt mit Hilfe des Projektmanagement-Werkzeugs BCS detailliert zu planen.

Bei den regelmäßigen Treffen der Koordinierungsteams wurde das gemeinsame Modell der globalen Informationsklassen (= Domänenontologie) besprochen. Dieses mussten die Modellierer importieren und hatten darauf aufbauend die Prozesse der eGK zu modellieren.¹⁵⁷ Während des Projekts hatten die Modellierer Verbesserungen hinsichtlich der verwendeten Objekte und des Designs umzusetzen. Die Aufforderungen dazu kamen von den Teamleitern, die diese meist in Zusammenarbeit mit dem Koordinierungsteam herausgearbeitet hatten.

Abstimmungsprozesse fanden in der Regel zwischen den benachbarten Schichten der Stablinienorganisation statt. Das heißt, Teamleiter hatten mit dem Koordinierungsteam Kontakt zu halten, wohingegen sich die Koordinatoren mit den Koordinierungsteams der anderen Gruppen sowie Prof Wikarski als Leiter des Gesamtprojekts abzustimmen hatten. Die Abstimmungsprozesse fanden in Treffen unter Anwesenheit der betreffenden Personen statt oder wurden, wenn es sich um zeitkritische Entscheidungen handelte, per Email oder Telefon durchgeführt. Die ständige Verfügbarkeit von umfangreichen *Prozessbeschreibungen, Aufgaben- und Zeitplänen* war für jeden Projektbeteiligten von Bedeutung.

Während für die Modellierung ein PC mit dem Betriebssystem MS Windows 2000, MS Visio 2003 sowie SemTalk in Version 2.0 verwendet wurde, nutzte man bei den im wöchentlichen Turnus stattfindenden Abstimmungsprozessen zumeist das Medium Papier. Das heißt, dass die Modelle ausgedruckt wurden. Für die Erstellung der zu besprechenden Arbeitspläne wurde MS Word verwendet. Wurden die Modellierer über den aktuellen Stand des globalen Informationsmodell oder auf Verbesserungen für ihre Modelle hingewiesen, kam ein Beamer zum Einsatz. Die Kommunikation zwischen den Teammitgliedern fand, unabhängig ob es sich um ein Modellierungs- oder ein Koordinierungsteam handelte, fernmündlich über Telefon oder *Skype*¹⁵⁸, mittels eines Instant Messengers, via Email oder durch persönliche Gespräche statt.

Die Modelle wurden auf einem zentralen Server gespeichert. Dazu wurde der BSCW-Server der Wirtschaftsinformatik verwendet, der ein Lesen und Schreiben der Daten sowohl im internen Netzwerk als auch von zuhause aus ermöglichte. Die Mitglieder der Modellierungsteams hatten ein prinzipielles Schreibrecht auf ihre Modelle und ihre Dokumentationen. Für die Koordinatoren genügte es, auf diese Daten lesend zuzugreifen. So konnten die Studenten auch zwischen den regelmäßigen Terminen an ihren Modellen und Dokumentationen zu arbeiten.

¹⁵⁷ Entsprechend der objektorientierten Modellierung mit SemTalk (siehe 2.1).

¹⁵⁸ Eine bekannte Anwendung zur Telefonie über das Internet.

Die Modelle der Modellierungsteams mussten von den Koordinierungsteams angeschaut werden. Das war nicht nur zum abschließenden Zusammenführen der Teilmodelle zu dem Gesamtmodell, sondern auch während der Modellierungsphase für die Überprüfung der Aufgabenerfüllung nötig. Mit Hilfe der Dokumentationen der Modellierungsteams hatten die Koordinatoren die Abschlussdokumentation zu erstellen.

Die befragten Studenten wünschten sich auf die offen gestellte Frage hin, was eine Plattform für die verteilte Modellierung leisten sollte, (i) die Bereitstellung einer Anleitung für das Werkzeug SemTalk, (ii) Tutorials und Erläuterungen zu den unterstützten Modellierungsmethoden, (iii) Beispielmodelle, auch als *best practices* oder *lessons learned* bezeichnet, (iv) eine einfache Bedienung, (v) Konsistenz überprüfende Funktionen bei verteilten Modellen, (vi) Mitteilung über die Tätigkeit der Modellierer, (vii) Sperrung von sich in Bearbeitung befindlichen Modellen, (viii) Vorschau auf Modelle, ohne dass ein Starten von SemTalk nötig ist. Diese Anforderungen fließen zusammen mit den schon ermittelten Kriterien (siehe 2.2.4) in die Zusammenfassung (siehe 3.2.3) ein und werden dort genauer erläutert.

Die Studenten äußerten über die hier genannten Anforderungen hinaus den Wunsch nach einer Funktion, die das automatische Aktualisieren des gemeinsamen Informationsmodells durchführt, sobald sich eine Veränderung an den Informationsklassen in den Teilmodellen vollzogen hat.¹⁵⁹ Das automatische Aktualisieren eines externen Modells durch ein anderes ohne Wissen des Eigentümers ist jedoch nicht gewollt. Das hat zwei Gründe: (i) Zu leicht könnte das externe Modell bei Änderungen an Referenzobjekten in Mitleidenschaft gezogen werden und (ii) verlangt der Prozess der Modellierung die Wahrnehmung und die Genehmigung von Veränderungen durch den Modell-Eigentümer. Diese Wahrnehmung wäre gefährdet, wenn die Aktualisierung von referenzierenden Objekten von den Referenzmodellen ausgelöst werden würde.

Es lassen sich folgende allgemeine Schlüsse aus der Umfrage unter den Studenten ziehen: Einige Studenten im 2. Semester sind hoch motiviert, wenn sie mit verantwortungsvollen Aufgaben im Zusammenhang mit einem Modellierungsprojekt betraut werden. Naturgemäß bestehen hinsichtlich der Fähigkeit zur Modellierung zu diesem Zeitpunkt – dem zweiten Semester – noch Erfahrungsdefizite. Auf die Fragen des Fragebogens konnten die Studenten deshalb häufig nur unpräzise Antworten geben. Einige Antworten ließen den Schluss zu, dass eine schlichte Erklärung des Begriffs *Plattform* nicht ausreicht. Denn viele Studenten gaben an, keinen Mehrwert mit der Verwendung einer webbasierten Plattform für die verteilte Modellierung erkennen zu können.

Telefoninterview bei der Bayer MaterialScience AG

Mit der Absicht, neben den Erfahrungen der Studenten des Modellierungsprojekts Anforderungen aus der Praxis ermitteln zu können, konnten die Herren Arndt und Dr. Gai-

¹⁵⁹Die Aktualisierung eines Modells mitsamt allen externen Referenzen wird bereits durch SemTalk unterstützt. Lediglich das automatisierte Aktualisieren – in wiederkehrenden Abständen oder beim Öffnen eines Modells – findet bislang nicht statt.

da von der Bayer MaterialScience AG (im Folgenden Bayer genannt) als erfahrene Modellierer aus der Industrie für zwei Telefoninterviews gewonnen werden. Bayer führt seit ca drei Jahren ein permanentes, geschäftsbegleitendes Modellierungsvorhaben durch. Ziel ist die Schaffung von Transparenz in den Geschäftsprozessen der neu gegründeten *Profit Center*¹⁶⁰ und die Ermöglichung des kontinuierlichen Verbesserungsprozesses (KVP) im Sinne eines Qualitätsmanagements in den Fachabteilungen. Demnach werden die Ergebnisse der Geschäftsprozessmodellierung der Controlling-Abteilung für die Budgetplanung zur Verfügung gestellt und mit den Leitern der jeweils betroffenen Fachabteilung hinsichtlich ihrer Korrektheit besprochen. Das bedeutet, die modellierende Organisationseinheit hat den Charakter einer Stabsabteilung. Mit dieser Querschnittsfunktion ausgestattet, gilt es, die Geschäftsprozesse der weltweit verteilten Abteilungen zu modellieren. Dabei muss berücksichtigt werden, dass der Bayer-Konzern über 15.000 Mitarbeitern hat.

Die Anzahl der Projektbeteiligten variiert zwischen einer und drei Personen, wobei mindestens eine Person Mitarbeiter von Bayer ist und die Teamleitung übernimmt. Üblicherweise ist noch ein unternehmensexterner Mitarbeiter im Projekt engagiert.

Daten, die sie für die Geschäftsprozessmodellierung benötigen, beziehen die beiden befragten Personen aus den Prozessbeschreibungen der gerade betrachteten Fachabteilung. Zudem recherchieren sie selbst und berufen Treffen ein, in denen sie die bereits modellierten Ergebnisse den Experten aus den Abteilungen präsentieren und zusätzliche Informationen erfragen, bzw Fehler aufdecken. Dem schließt sich Korrektur der Modelle sowie allgemeinen Bereinigen der schon modellierten Prozesse an.

Die Modelle werden auf einem zentralen Server gespeichert. Abstimmungsprozesse finden in Anwesenheit der beteiligten Personen statt oder werden mittels Telefon durchgeführt. Emails werden für diesen Zweck bei Bayer selten verwendet.

Für die regelmäßig stattfindenden Präsentationen der Modellierungsergebnisse werden hauptsächlich Ausdrucke auf Papier verwendet. Während eines der zyklischen Expertentreffen kommen außerdem der Flip-Chart und die Pinnwand zur Visualisierung von Diskussionsthemen sowie der Beamer für die Präsentation von Modell-Exporten im PowerPoint-Format zum Einsatz.

Dem Vorgehen bei der Modellierung kann kein bestimmter Ablauf zugeordnet werden. Das ist insofern unkritisch, als dass die Möglichkeiten SemTalks, verteilt zu modellieren, kaum angewendet werden. Verwunderlich ist nur, dass die Befragten sich intensiv mit der Modellierung von Geschäftsprozessen beschäftigen und auch einen Großteil der Möglichkeiten zur verteilten Modellierung mit SemTalk zu kennen scheinen. Und obwohl eine verteilte Modellierung für Bayer sinnvoll wäre – zum Beispiel mit der

¹⁶⁰ Methode des betrieblichen Controllings, einzelne Unternehmensbereiche in sogenannte Profit Center aufzuteilen, um deren Rentabilität besser beurteilen zu können.

Wiederverwendung von Modellen –, soll es auf absehbare Zeit auch nicht genutzt werden.

Nach Angaben der interviewten Personen stellt für sie eine Plattform für die verteilte Modellierung keinen Mehrwert dar, da vorerst noch ein hoher Nachholebedarf im Umgang der Modellierungsmethoden besteht. Ein verteiltes Modellieren ist nach Aussage von Dr. Gaida frühestens in drei bis fünf Jahren von Interesse. Für die Industrie mag diese exemplarische Feststellung richtig sein. Im Bereich der Forschung, aber auch im Dienstleistungsgewerbe können aber die Anfänge anderslautender Tendenzen vernommen werden (siehe [Ceccaroni; Kendall 2002], [Sure ua 2002], [Vom Brocke 2003b], [Hodgson 2005], [Naumenko ua 2005] und [Sure ua 2005]).

Die mit Mitarbeitern der Firma Bayer durchgeführten Interviews spiegeln den aktuellen Stand zur verteilten Modellierung in der Industrie wieder: Hier konnte weder ein Bedarf an verteilter Modellierung noch an einer Plattform artikuliert werden. Die Schwerpunkte liegen oftmals noch bei der Erlernung der Modellierungsmethode und der Bedienung des Modellierungswerkzeugs.

3.2.3 Zusammenfassung und erste Gegenüberstellung

Das Spektrum der Anforderungen, das sich aus den im Grundlagen-Teil vorgestellten Arbeiten¹⁶¹ und den eigenen Untersuchungen herausgebildet hat, sind in der Tafel 5 zusammengefasst. Die Einteilung der Tabelle orientiert sich in groben Zügen an der *Semantic Collaboration Environment Architecture* (siehe 2.3(2)), da die dortige Struktur angemessen erscheint. Auf die SCEA-Kategorien *Graphics*, *Semantic Infrastructure* und *Platform Infrastructure* wurde hier verzichtet. Denn in dieser Arbeit sind SemTalk in Verbindung mit MS Visio als grafischer Editor und die WSS, aufbauend auf dem Windows Server 2003 und .NET, als technische Plattform als gegeben vorausgesetzt. Die Semantic Infrastructure wird derzeit noch durch den SemTalk-Assistenten und dem Modell-Index vertreten.

In der Tafel 5 sind nur Funktionen zu finden, die noch nicht durch SemTalk erfüllt werden¹⁶² und die von einer gemeinsam genutzten Plattform bereitgestellt werden können¹⁶³.

¹⁶¹ [Sure ua 2002], [Ceccaroni; Kendall 2002], [Simon 2004], [Sure ua 2005], [Hodgson 2005] und [Weichhardt 2005].

¹⁶² Das Zusammenfassen mehrerer Modelle kann über die Import-Funktion in SemTalk umgesetzt werden.

¹⁶³ Es existieren ausreichend viele Anbieter mit Produkten, die sich auf Funktionen wie Email, Instant Messaging und Internettelefonie spezialisiert haben. Deren Umsetzung innerhalb einer Plattform erscheint darum nicht sinnvoll.

Virtueller Projektraum und Workgroup Funktionalität	Werkzeuge	Verteilte Modellierung	Knowledge Ausstattung	Ereignisverwaltung
Anzeige der Anwesenheit	Integration von SemTalk	Dateiberechtigungen	Suchfunktion	Kalender/Termine
Gruppenraum	Integration von MS Office	Versionsverwaltung für Modelle	Beispielbibliothek	Festlegung von Meilensteinen
Hohe Verfügbarkeit	Integration heterogener Daten	Sperrmechanismen		Bekanntmachungen
Einfache Bedienung	Benutzerverwaltung	Versionsbezogene Kommentare		
Besprechungs-räume	Vorausansicht von Modellen	Konsistenzprüfung / Suche nach externen Objekten		
Aufgabenplanung		Änderungsmitteilung für Referenzmodelle		
Diskussionsbrett		Änderungsmitteilung für Referenzobjekte		
Konferenzen		Änderungsmitteilung für externe Verfeinerung		
Argumentation Framework		Import von Modellen		

Tafel 5

Die nun ermittelten Anforderungen werden nachfolgend den Funktionen und Möglichkeiten gegenübergestellt, die in den WSS verfügbar sind. Die jeweils in der linken Spalte aufgeführte Anforderung wird dort kurz beschrieben. In der rechten Spalte erfolgt die Erläuterung der damit übereinstimmenden Funktion der WSS.

Anforderung	Funktion oder Lösung
Virtueller Projektraum	
<p><i>Anzeige der Anwesenheit</i> Für die Wahrnehmung in der Gruppe (group awareness) ist es wichtig zu sehen, ob andere Gruppenmitglieder derzeit erreichbar sind.</p>	<p>Sofern der MSN IM als Instant Messenger verwendet wird, ist im <i>Freigegebenen Arbeitsbereich</i> der MS Office-Anwendung die Anwesenheit sichtbar.</p>
<p><i>Gruppenräume</i> Arbeitsgruppen benötigen einen Raum zur Entfaltung von Gruppendynamik (Entscheidungsfindung, Diskussion).</p>	<p>Es lassen sich weitere gemeinsame Arbeitsbereiche oder Dokumentarbeitsbereiche in der Hierarchie des virtuellen Servers einrichten (siehe 2.4).</p>
<p><i>Hobe Verfügbarkeit</i> Die Erreichbarkeit der Plattform muss unabhängig von Zeit und Ort gewährleistet sein.</p>	<p>Diese ist von der Konfiguration der Infrastruktur abhängig, in welcher der Windows Server betrieben wird (Hardware, lokales Netzwerk, Kapazität der Internetanbindung, Firewall).</p>
<p><i>Einfache Bedienung</i> Der Aufwand zum Erlernen der Funktionen soll minimal sein (zB durch intuitive Benutzerführung).</p>	<p>Die webbasierte Oberfläche und die Repräsentation des gemeinsamen Arbeitsbereiches in der MS Office-Anwendung erlauben eine leichte Bedienbarkeit.</p>
<p><i>Diskussionsrunden</i> Diese werden zur Klärung von Fragen (Vorgehen, Designentscheidungen bei der Modellierung) benötigt.</p>	<p>Können mit wenigen Mausklicks in einem gemeinsamen Arbeitsbereich eingerichtet werden.</p>
<p><i>Besprechungsräume</i> Zur Vorbereitung von Treffen mit physikalischer Anwesenheit der am Modellierungsprojekt Beteiligten</p>	<p>Die vier Typen von Besprechungsarbeitsräumen (Standard, Entscheidungssitzung, Soziale Zusammenkünfte, Planung und Organisation) sind vordefinierte Arbeitsbereiche und lassen sich bei der Einrichtung eines neuen Arbeitsbereiches aus dem Katalog der Vorlagen auswählen. Sie sind allesamt ähnlich aufgebaut und bereiten eine Anzahl von Listen und Webparts vor.</p>
<p><i>Aufgabenplanung</i> Aufgaben sind nach ihrer Identifikation Personen zur Bearbeitung zugeordnet. Dabei sind Termine zu bestimmen.</p>	<p>Ein Listentyp ist die Aufgabenliste, die es erlaubt, Aufgaben zu beschreiben, dann Benutzern zuzuordnen und für die Erledigung Termine festzulegen.</p>
<p><i>Konferenzen</i> Werden für die zur selben Zeit aber an unterschiedlichen Orten stattfindenden Abstimmungsprozesse gebraucht.</p>	<p>Die WSS bieten <i>keine</i> Unterstützung der synchronen Kommunikation.</p>
<p><i>Argumentation Framework</i> Vorlagen zur Beantragung von Änderungen an dem gemeinsamen Modell.</p>	<p>Könnte mit einem zentral verfügbaren Dokument oder einer speziell angepassten Liste realisiert werden.</p>

Anforderung	Funktion oder Lösung
Werkzeuge	
<p><i>Integration von SemTalk</i> Modelle sollten sich direkt aus der Dokumentbibliothek öffnen und bearbeiten lassen.</p>	<p>SemTalk-Modelle werden im Dateiformat von MS Visio gespeichert. Die WSS erkennen den Dokumententyp von MS Visio und erlauben das sofortige Öffnen über das Kontextmenü der Datei (siehe 2.4.5). MS Visio und SemTalk müssen am zugreifenden Client-PC installiert sein. Das direkte Bearbeiten von Dokumenten im Internet Explorer ist nicht möglich.</p>
<p><i>Integration von MS Office</i> MS Word- oder MS Excel-Dokumente (zB Prozessbeschreibungen) sollten sich, wie die SemTalk-Modelle, direkt bearbeiten lassen.</p>	<p>Wie oben ausgeführt, erkennen die WSS ein MS-Office-Dokument. Das Dokument kann über dessen Kontextmenü in der damit logisch verbundenen MS-Office-Anwendung geöffnet werden.</p>
<p><i>Integration heterogener Daten</i> Oft müssen Dokumente mit gesichert werden, die nicht von einer MS Office-Anwendung stammen (zB Bilddateien).</p>	<p>Die in den WSS verankerte Verwaltung von Dokumenten steht für alle Dateiformate zur Verfügung. Das sofortige Öffnen über das Kontextmenü (siehe beide vorherigen Punkte) aber ist – ohne Anpassungen vornehmen zu müssen – MS-Office-Dokumenten vorbehalten.</p>
<p><i>Benutzerverwaltung</i> Die Zuordnung von Rechten erfolgt über ein Benutzerkonto.</p>	<p>Ausführlich möglich (siehe 2.4.2). Siehe auch Punkt <i>Dateiberechtigungen</i>.</p>
<p><i>Vorausansicht von Modellen</i> Die Teilmodelle sind von dem zentralen Gremium regelmäßig zu prüfen. Dieses muss lediglich lesen können.</p>	<p>Zwei Varianten zur Ansicht von Modellen im Internet Explorer: (i) Ist in den WSS über ein Webpart (<i>Seiten-Viener-Webpart</i>) eingebettet möglich oder (ii) mit Klick auf das Modell in der Dokumentbibliothek MS Visio im Internet Explorer starten.</p>
Verteilte Modellierung	
<p><i>Versionsbezogene Kommentare</i> Designentscheidungen sollten zum Zweck der Nachverfolgung (<i>monitoring</i>) auf die Version bezogen festgehalten werden können.</p>	<p>Bei der Freigabe einer neuen Version (<i>Einchecken</i>) kann ein Kommentarfeld gefüllt werden. In der Versionsverwaltung, ein vordefinierter Listentyp, werden die Kommentare zu den jeweiligen Versionen angezeigt.</p>
<p><i>Dateiberechtigungen</i> Zugriffssteuerung auf Ergebnisse anderer Projektbeteiligter (zB zentrale Dokumente, externe Modelle, gemeinsames Modell).</p>	<p>Es lassen sich keine expliziten Rechte auf Ebene <i>eines</i> Dokuments vergeben. Die Rechte eines Benutzers für eine Dokumentbibliothek können aber abweichend von seinen Rechten innerhalb des gemeinsamen Arbeitsbereichs definiert werden. So ließen sich für Dokumente, die ein Leser bearbeiten darf, zusätzliche Dokumentbibliotheken einrichten, auf die er schreibende Rechte erhält.</p>
<p><i>Versionsverwaltung für Modelle</i> Alte Versionen der Modelle sollten sich wiederherstellen lassen. Anwendungsfälle: Fehlerkorrektur und Monitoring des Modellierungsprozesses.</p>	<p>Die Versionsverwaltung wird immer für die ganze Dokumentbibliothek aktiviert. Werden Dokumente gespeichert, erhöhen sich deren Versionsnummern. Es lassen sich die Versionsstände einzelner Dokumente anzeigen, alte Versionen wiederherstellen aber auch löschen.</p>
<p><i>Sperrung von Modellen</i> Sperrung eines Modells zu dessen exklusiver Bearbeitung; Andere</p>	<p>Ein Benutzer kann ein Dokument sperren (<i>Auschecken</i>) und freigeben (<i>Einchecken</i>). Das ist in den WSS über das Kontextmenü des jeweiligen Dokuments und in der MS-</p>

Anforderung	Funktion oder Lösung
können sich das Modell lediglich anzeigen lassen.	Office-Anwendung über das <i>Datei</i> -Menü möglich (siehe Tafel 2 in 2.4.5)
<i>Überprüfung von Konsistenz</i> Mechanismen zur Wahrung Projekt übergreifender Konsistenz.	Keine Lösung
<i>Änderungsmitteilung für Referenzmodelle</i> Eigentümer von Modellen referenzierender Objekte müssen wissen, wann ein Referenzmodell aktualisiert wurde, damit auch sie ihr Modell aktualisieren können.	Für jedes Dokument einer Dokumentbibliothek lassen sich Benachrichtigungen (per Email) für die Benutzer einrichten. Diese können sich auf bestimmte Ereignisse beschränken (zB Aktualisierung eines Modells) und in vordefinierten Intervallen abschicken.
<i>Änderungsmitteilung für Referenzobjekte</i> Gezielte Benachrichtigung des Eigentümers eines referenzierenden Objekts über die Nichtaktualität der externen Referenz.	Keine Lösung
<i>Änderungsmitteilung für externe Verfeinerungen</i> Gezielte Benachrichtigung des Modell-Eigentümers bzgl einer nicht mehr existierenden oder erreichbaren externen Verfeinerung.	Keine Lösung
Knowledge-Ausstattung	
<i>Suche</i> Zum Beispiel nach zentralen Dokumenten oder externen Modellen.	<i>Suche nach Dokumenten:</i> Die WSS-Suche erwartet die Angabe des Dateinamens und der Dateinamenerweiterung. Platzhalter (<i>Wildcard</i> s) werden ignoriert. Außerdem wird nur in dem gemeinsamen Arbeitsbereich gesucht, von welchem die Suche gestartet wurde. <i>Suche nach Dokumentinhalten:</i> Inhalte von MS Office-Dokumenten werden gefunden (also auch in SemTalk-Modellen), aber <i>Wildcard</i> s werden auch hier ignoriert. Der Suchende muss genau wissen, wonach er sucht. <i>Suche nach Listeninhalten:</i> Listen (zB Aufgaben, Termine, Ereignisse, Diskussionen) werden mit der gleichen Einschränkung wie in den anderen Fällen durchsucht.
<i>Beispielbibliothek</i> Bereitstellung einer Bibliothek von Standard-Lösungen und Modellierungsbeispielen.	Kann mit einer zentral verfügbaren Dokumentbibliothek realisiert werden. Der Suchende hat mit den genannten Einschränkungen umzugehen.
Ereignisverwaltung	
<i>Kalender/ gemeinsame Termine</i> Für die Projektplanung sind gemeinsame Termine zu erfassen.	Eine kalenderbasierte <i>Ereignisse</i> -Liste ist im Katalog der vordefinierten Listen verfügbar. Die Ereignisse können mit Anfangs- und End-Terminen sowie einem Text versehen werden. Zudem kann ein Dokument an das Ereignis angehängen werden.
<i>Festlegung von Meilensteinen</i> Die Pflege von Meilensteinen gehören zur erfolgreichen Überwachung	Lassen sich mit einer der Listentypen <i>Ereignisse</i> oder <i>Aufgaben</i> realisieren.

Anforderung	Funktion oder Lösung
des Projektverlaufs. <i>Bekanntmachungen</i> Ereignisse, von der alle Beteiligten wissen sollten, müssen zentral veröffentlicht werden.	Die <i>Ankündigungen</i> -Liste auf oberster Ebene eines gemeinsamen Arbeitsbereich kann von allen Teilnehmern eingesehen werden. Verweise in Dokumentarbeitsbereichen führen deren Mitglieder auf die zentrale Ankündigungen-Liste.

Tafel 6

Es zeigt sich, dass die WSS mit dem standardmäßig zur Verfügung stehenden Funktionen einen Großteil der Anforderungen an eine Plattform für die verteilte Modellierung abdecken. Aber sobald für die Erfüllung einer Anforderung spezielle Anwendungslogik erforderlich ist, so kann sie verständlicherweise nicht unterstützt werden. Für diese Anforderungen gilt es, im Entwurf Möglichkeiten auszuarbeiten, die eine Erfüllung gewährleisten. Doch zunächst werden die von den WSS nicht abgedeckten Anforderungen genauer untersucht.

(1) Überprüfung von Konsistenz. Um die Konsistenz zwischen Modellen einer Projekt-Umgebung¹⁶⁴ herzustellen, ist der SemTalk-Assistent Merlin mit Verwendung der Modell-Indexdatei zuständig (siehe 2.1.1(4)). Abbildung 22 verdeutlicht die Aufgaben, die der SemTalk-Assistent dabei durchführt: Erzeugt der Modellierer ein neues Objekt in einem SemTalk-Modell, dann greift der SemTalk-Assistent automatisch im Hintergrund auf den Modell-Index zu und versucht in diesem eine begriffliche Übereinstimmung zu finden. Für diesen Suchvorgang hat der SemTalk-Assistent Einstellungen (*Optionen*) zu berücksichtigen, die in der SemTalk-Anwendung hinterlegt sind. Hat der SemTalk-Assistent eine Übereinstimmung gefunden, zeigt er diese dem Modellierer an. Eine ähnlich komfortable Lösung sollte mit den Modellen einer Dokumentbibliothek möglich sein. Aber das bisherige Verfahren kann aus technischen Gründen nicht für die WSS angewandt werden.

Die Anforderung kann demzufolge so formuliert werden: Anzeige aller externen Objekte aus den Modellen einer Dokumentbibliothek, die in den dortigen Modellen synonym oder homonym zu dem markierten Objekt des betrachteten Modells verwendet werden. So hätte ein Modellierer einen schnellen Überblick über alle Kandidaten für eine externe Referenz und könnte das Zutreffende als Referenzobjekt auswählen. Wenn zudem eine Gewichtung der externen Objekte stattfände, könnte der Modellierer das externe Objekt zum Referenzobjekt machen, das bei gleichlautender Bezeichnung die höchste Platzierung erreicht.¹⁶⁵ Das könnte die Auswahl stark vereinfachen.

¹⁶⁴ Gemeint ist ein Speicherort in dem mehrere Modelle gesichert werden.

¹⁶⁵ Dabei könnte es sich um ein Objekt handeln, was von vielen Modellen referenziert wird, also besonders wichtig ist. Weiterhin wäre denkbar, dass der Modellierer seine Objekte selbst gewichtet (zB hohes Gewicht, wenn das Objekt in seinem Prozess erzeugt wird).

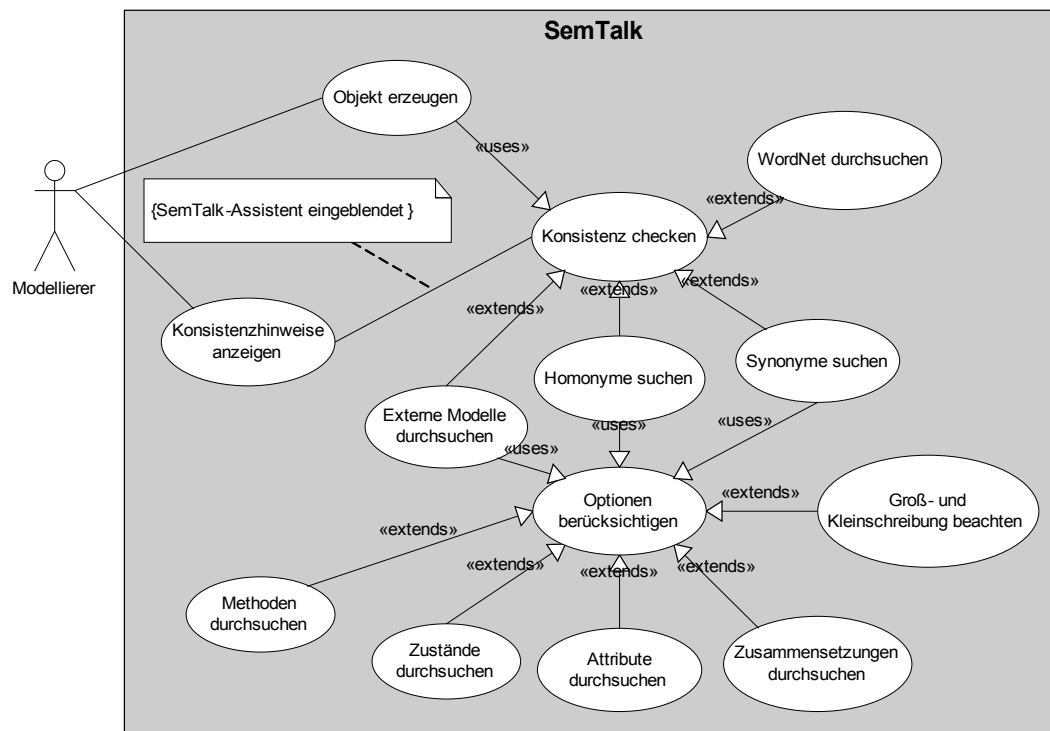


Abbildung 22. Überprüfung der Konsistenz in SemTalk – Anwendungsfälle

Ferner sollten mit Verwendung der WSS als Plattform neue Funktionen nutzbar werden: Zum Beispiel wäre es denkbar, sich eine Liste oder Übersicht aller Diagramme der externen Modelle ausgeben zu lassen. Damit wäre das Auffinden von Kandidaten für eine externe Verfeinerung wesentlich vereinfacht. Der Modellierer, der eine externe Verfeinerung anlegen möchte, kann so gezielt in die externen Modelle hineinsehen und sich vergewissern, dass das externe Diagramm tatsächlich einer Detaillierung seines Objekts entspricht.

(2) Änderungsmitteilung über Referenzobjekte und -modelle. SemTalk überprüft nicht selbstständig, ob die externen Referenzen eines Modells noch korrekt sind. Startet der Modellierer die Aktualisierung des referenzierenden Objekts oder des Modells und kann SemTalk das Referenzobjekt nicht erreichen, dann wird die externe Referenz weder automatisch gelöscht, noch erhält der Modellierer Hinweise (siehe 2.1.1(2)). Daraus ergibt sich die Schlussfolgerung, dass der Eigentümer eines Modells nie weiß, wann sein Modell – insbesondere die darin befindlichen Objekte mit externen Referenzen – korrekt ist. Es wäre wünschenswert, wenn der Eigentümer eine gezielte Benachrichtigung über den geänderten Zustand eines Referenzobjekts erhalten könnte. Diese Mitteilung könnte ihn dazu veranlassen, sein Modell in einen konsistenten Zustand zu bringen.¹⁶⁶ Von Vorteil wäre auch, wenn die Mitteilung unabhängig davon zugestellt wird, ob der Eigentümer ge-

¹⁶⁶ Aktualisieren des referenzierenden Objekts oder des gesamten Modells bzw Entfernen oder Erneuern der externen Referenz.

rade an seinem Modell arbeitet oder nicht. Er könnte damit seine persönliche Aufgabenplanung verbessern.

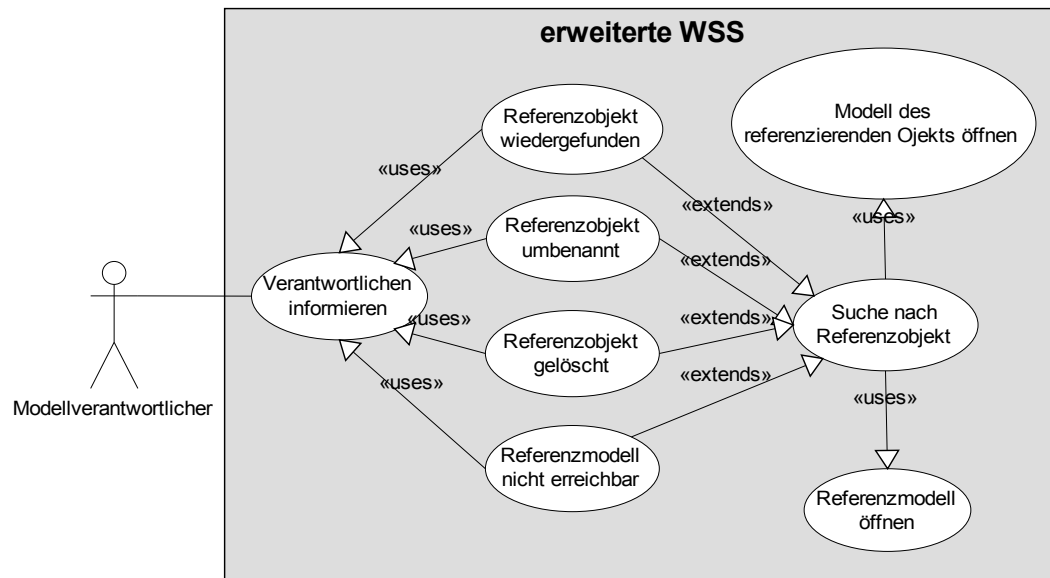


Abbildung 23. Verwaltung externer Referenzen durch die WSS – Anwendungsfälle

Zur Erläuterung der Abbildung 23: Notwendigerweise benötigt man für die Überprüfung der Gültigkeit einer externen Referenz den Zugriff auf das Modell, welches das referenzierende Objekt beinhaltet, und das Referenzmodell. Dabei kann einer von fünf möglichen Zuständen festgestellt werden, die nachfolgend erläutert werden.

Zustand der externen Referenz	Erläuterung
Das Referenzmodell ist unter der Pfadangabe in der externen Referenz erreichbar und das Referenzobjekt wurde gefunden.	In dem Anwendungsfall-Diagramm nicht gezeigt, da es sich um den (korrekten) Normalfall handelt und darum kein Handlungsbedarf besteht.
Das <i>Referenzmodell</i> ist nicht unter der Pfadangabe erreichbar.	Das Referenzmodell ist gelöscht, verschoben oder umbenannt worden.
Das Referenzmodell ist erreichbar, aber das <i>Referenzobjekt</i> wurde gelöscht.	Die Objekt-ID, die in der externen Referenz als SrcID aufgeführt ist, kann in dem Referenzmodell nicht mehr gefunden werden.
Das Referenzmodell ist erreichbar, aber das <i>Referenzobjekt</i> wurde umbenannt.	Die externe Referenz ist weiterhin gültig. Es muss lediglich die Bezeichnung des referenzierenden Objekts an jene des Referenzobjekts angepasst werden.
Das Referenzmodell ist erreichbar, aber das <i>Referenzobjekt</i> wurde nach zeitlicher Löschung neu erzeugt.	Es gibt zwar ein Objekt mit der Bezeichnung des referenzierenden Objekts in dem Referenzmodell, aber es besitzt eine neue Objekt-ID (das frühere Referenzobjekt wurde gelöscht und anschließend neu erzeugt). Da externe Referenzen auf der SrcID (= Objekt-ID des Referenzobjekts) basieren, ist die externe Referenz – aus Sicht des referenzierenden

Zustand der externen Referenz	Erläuterung
	Objekts – ungültig (es verweist auf eine nicht-existierende Objekt-ID).

Tafel 7

Abhängig von der Art des ungültigen Zustands muss der Eigentümer des Modells, welches das referenzierende Objekt beinhaltet, informiert werden. Denn er soll *aktiv* an der Korrektur des eigenen Modells beteiligt sein. Die automatische Richtigestellung seines Modells ohne sein Zutun hätte zur Folge, dass er der Korrektur nicht zustimmen kann. Wäre ein diesbezüglicher Automatismus tatsächlich vorhanden, könnten durch die Änderung oder Löschung eines von mehreren externen Modellen referenzierten Objekts die Modelle der referenzierenden Objekte beschädigt werden.

(3) Änderungsmitteilung über externe Verfeinerungen. Ähnlich wie im Fall ungültiger externer Referenzen lassen sich für die externen Verfeinerungen Anwendungsfälle ermitteln, für die mit Hilfe einer erweiterten Funktionalität der WSS Mehrwerte erzeugt werden können (siehe Abbildung 24).

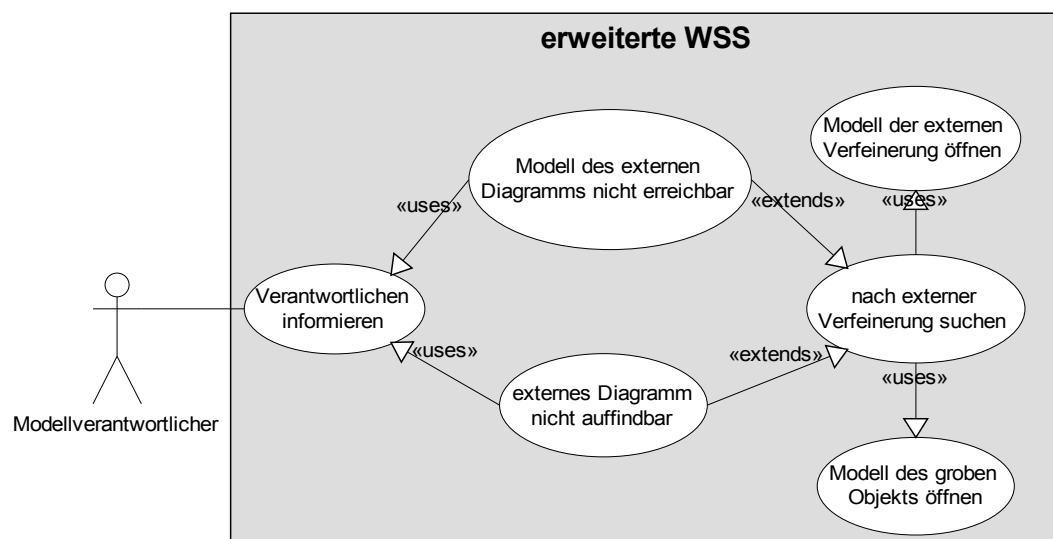


Abbildung 24. Verwaltung externer Verfeinerungen durch die WSS – Anwendungsfälle

Da aber bei der externen Verfeinerung auf eine zur SrcID (siehe 2.1.1(2)) vergleichbare Eigenschaft verzichtet wird, das Verweisziel also nur über die Bezeichnung – den Namen des externen Diagramms – gesucht wird, lassen sich nur zwei mögliche Anwendungsfälle identifizieren: (i) Entweder ist das externe Modell nicht erreichbar oder (ii) das externe Modell ist erreichbar, aber das externe Diagramm ist nicht auffindbar. In beiden Fällen sollte der Eigentümer des Modells mit dem verfeinerten Prozessobjekt informiert werden.

3.2.4 Unterstützung bestehender Techniken zur verteilten Modellierung

Bevor die WSS als Plattform für die verteilte Modellierung mit SemTalk genutzt werden können, muss SemTalk an die Zusammenarbeit mit den WSS angepasst werden. Das ist nötig, da die WSS eine Vielzahl von Mechanismen besitzen (zB Zugriffssteuerung auf Dokumente), die sich nicht umgehen lassen.

Für die Anpassung von SemTalk sind zunächst die Anwendungsfälle zu definieren, die sich aus den Techniken zur verteilten Modellierung ergeben. Ein Studium der verfügbaren Anleitungen und Veröffentlichungen schaffte das nötige Verständnis für die konkreten technischen Abläufe.

In dem in Abbildung 25 gezeigten Anwendungsfall-Diagramm befindet sich SemTalk als anzupassendes System an zentraler Stelle. Einige SemTalk-Funktionen (Modell speichern, externes Modell öffnen) müssen an die Bedingungen der WSS angepasst werden, andere kommen neu hinzu (Anzeigen des *Freigegebenen Arbeitsbereichs*).

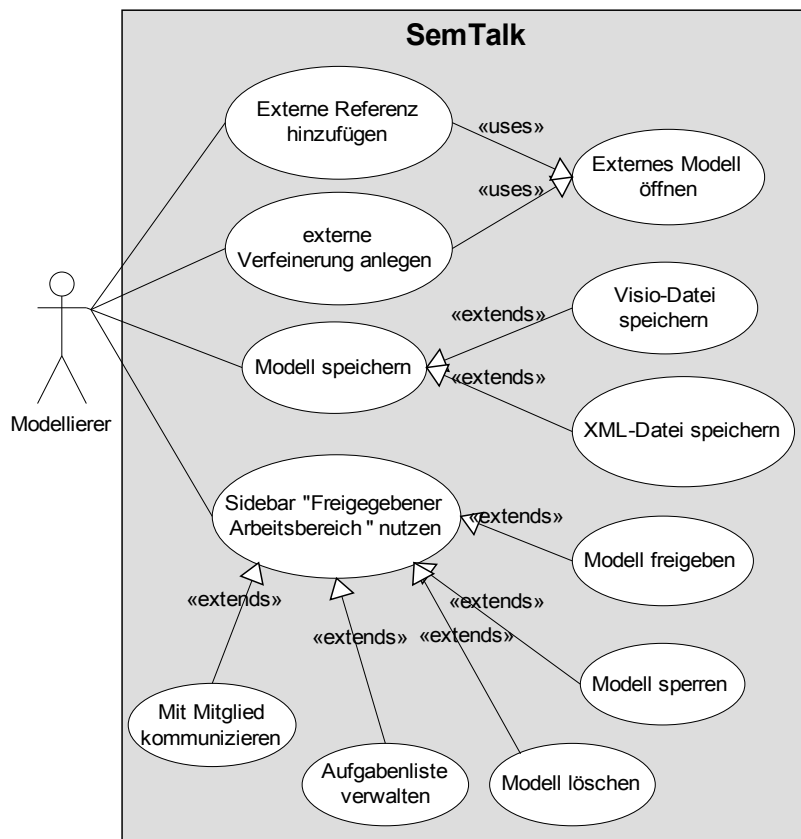


Abbildung 25. Verteilte Modellierung mit SemTalk und den WSS – Anwendungsfälle für SemTalk

Für das Erzeugen von externen Referenzen oder externen Verfeinerungen muss SemTalk auf das externe Modell zugreifen und es lesen können (siehe 2.1.1(2)). Deshalb ist es für

die verteilte Modellierung entscheidend, neben der eigentlichen Modell-Datei im Dateiformat von MS Visio, die korrespondierende XML-Datei zu speichern.¹⁶⁷

Die Anzeige des *Freigegebenen Arbeitsbereichs* in SemTalk ist besonders wichtig, da mit ihm die Funktionen der WSS zur Kommunikation, Koordination und Kooperation bereitgestellt werden.¹⁶⁸ Der Modellierer braucht lediglich SemTalk zu starten und kann auf das parallele Bedienen des IE verzichten.

3.3 Entwurf der Erweiterung für die verteilte Modellierung

„Die Aufgabe des Entwurfs ist es [...], die spezifizierte Anwendung auf einer Plattform unter den geforderten technischen Randbedingungen zu realisieren.“ [Balzert 2005:13]

Das vorrangige Ziel des Entwurfs besteht darin, die Vorteile die sich mit dem Einsatz der WSS als Plattform ergeben, nicht mit zusätzlichem Bedienungsaufwand zu erbringen. Demnach hat die zu entwerfende Lösung die ermittelten Anforderungen zu erfüllen und gleichzeitig möglichst vor dem Modellierer verborgen abzulaufen. Im ersten Schritt werden die Technologien vorgestellt, die im Zusammenhang mit den WSS und der Application-Server-Fähigkeit des Windows Server 2003 für die Umsetzung zur Verfügung stehen. Besteht über die technischen Möglichkeiten Klarheit, wird eine Lösung entworfen, die – um den Rahmen dieser Arbeit nicht zu sprengen – sich auf die Änderungsmitteilung für Referenzobjekte und Referenzmodelle konzentriert (siehe 3.2.3(2)). Der Entwurf und die Umsetzung der Änderungsmitteilung externer Referenzen ließ keinen Spielraum für konkrete Überlegungen zusätzlich zu implementierender Unterstützung. Darum wird die Änderungsmitteilung für externe Verfeinerungen (siehe 3.2.3(3)) nur grob skizziert. Sie bleibt, wie die Wahrung von Projekt übergreifender Konsistenz (siehe 3.2.3(1)), bei der Umsetzung außen vor.

3.3.1 *Verfügbare Technologien*

Die Anpassung und Erweiterung einer Standardsoftware wie den WSS erfordert die Auseinandersetzung mit den Technologien, die der Hersteller zu ihrer Realisierung einsetzte. Meist sind damit die Möglichkeiten zur Umsetzung eigener Funktionen bereits festgeschrieben und gelten somit als zu berücksichtigende Randbedingungen.

Für die Realisierung der WSS hat Microsoft eine Vielzahl von Technologien eingesetzt, die sich auch für Erweiterungen und Anpassungen nutzen lassen. Eine grobe Einteilung

¹⁶⁷Nur wegen des offenen Formats kann auf die XML-Datei mittels einfacher Lese- und Schreibmechanismen zugegriffen werden.

¹⁶⁸Die in dem Anwendungsfall-Diagramm von *Sidebar Freigegebener Arbeitsbereich nutzen* abgeleiteten Anwendungsfälle sind Beispiele für dessen Nutzung.

der Technologien ist möglich, indem in *verwalteten* und *nicht-verwalteten* Programmcode unterteilt wird.

(1) Verwalteter Code. Wird Programmcode in einer (Programmier-)Hochsprache entwickelt, die von dem .NET-Framework unterstützt wird,¹⁶⁹ dann wird dieser Programmcode nicht direkt in Maschinencode¹⁷⁰ übersetzt. Vielmehr wird er in einen plattformspezifischen Zwischencode – die *Common Intermediate Language* (CIL) – überführt [Schwichtenberg 2006a:7].¹⁷¹ Alle Hochsprachen des .NET-Framework greifen auf die zentral bereitgestellte .NET-Klassenbibliothek zu. Mit Hilfe der CIL ist es jedem Entwickler gestattet, in einer von ihm präferierten Hochsprache zu entwickeln.

Der in der CIL vorliegende – verwaltete – Code wird erst zum Zeitpunkt der Ausführung vom Just-in-Time-Compiler (JIT) in den Maschinencode übersetzt [ebenda]. Darum benötigt verwalteter Code etwas mehr Zeit zur Ausführung als bereits vorkompilierter Maschinencode.¹⁷²

(2) Nicht-verwalteter Code. *Prozessorspezifisch* kompilierter Maschinencode wird als nicht-verwalteter oder auch nativer Code bezeichnet [Schwichtenberg 2006a:7]. Dazu gehören die .NET-fremden Programmier- und Skriptsprachen. In den WSS liegt eine Vielzahl von Programmcode in nicht-verwaltetem Code vor.

In Tafel 8 werden die in den WSS verwendeten Technologien nach verwaltetem und nicht-verwaltetem Code eingeteilt.

Verwalteter Code	Nicht-verwalteter Code
ASPX- und ASMX-Webseiten	Remote Procedure Calls (RPC)
Webdienste (web services)	WebDAV
Webparts	CAML
Ereignisroutinen	SQL
	JavaScript

Tafel 8

¹⁶⁹ Im aktuellen .NET-Framework werden ca 43 Hochsprachen unterstützt [Schwichtenberg 2006b:17].

¹⁷⁰ Maschinencode ist prozessorspezifischer Binärkode (Instruktionen und Daten), der von dem Mikroprozessor ausgeführt werden kann <<http://de.wikipedia.org/wiki/Maschinencode>> (12.4.2006). Er wird von Compilern oder Assemblern erzeugt.

¹⁷¹ Man spricht von einem Intermediationskonzept, wie es auch bei der Sprache Java vorliegt. Bei Java wird der Quellcode ebenfalls in eine Zwischensprache – den sogenannten *Bytecode* – übersetzt. Er wird zur Laufzeit von der *Java Virtual Maschine* in Maschinencode übersetzt.

¹⁷² Es war und ist üblich, dass der Quellcode einiger Hochsprachen in *nativen* Maschinencode übersetzt – kompiliert – wird. Erst nach der Kompilierung ist der Programmcode auf dem Rechner ablauffähig.

In der Abbildung 26 wird der Unterschied der Behandlung der beiden Arten von Programmcode bei einer clientseitigen Anfrage (zB Internet Explorer) deutlich.

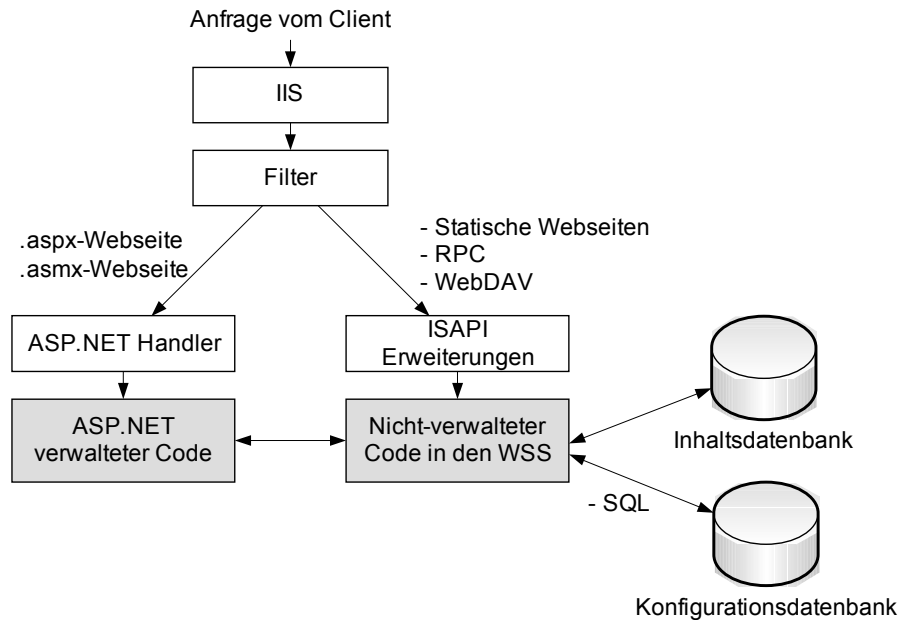


Abbildung 26. Verwalteter und nicht-verwalteter Code in den WSS.

Quelle: [Microsoft 2004]

Bis auf die mit JavaScript entwickelten Funktionen, können alle anderen Technologien auch von Programmen genutzt werden. Das erlaubt ihre Nutzung für eine programmier-technische Anpassung und Erweiterung. Aber wie schon in 2.4.1(1) festgestellt wurde, war es nicht Absicht von MS, die aufgeführten Technologien als Möglichkeit zur Erweiterung durch die Anwender bereitzustellen. Dank der großen Nachfrage vonseiten der Anwender und Entwickler sind im Laufe der Zeit jedoch immer mehr Details zur Implementation der WSS an die Öffentlichkeit gedrungen. Nachfolgend werden die Technologien einzeln vorgestellt. Dabei werden die Möglichkeiten und Einschränkungen der jeweiligen Technologie herausgearbeitet.

Das Microsoft SharePoint Object Model

Im Verlauf der Installation fügen die WSS dem .NET-Framework des Windows Server eine Klassenbibliothek hinzu. Mit dem Namespace `Microsoft.SharePoint` werden die Klassen des *Microsoft SharePoint Object Model* verfügbar, die von den WSS zur Ausführung des eigenen Programmcodes benötigt werden. Bei dem SharePoint Object Model handelt es sich um verwalteten Code, dessen Klassen dem Entwickler die Möglichkeit geben, eigenen Programmcode zur Verarbeitung von Informationen über die WSS sowie deren Anpassung und funktionale Erweiterung zu entwickeln. Mit den verfügbaren Objekten können administrative und sicherheitsrelevante Aufgaben erledigt werden.

Selbst die Arbeit mit Dokumenten und Listen wird von dem SharePoint Object Model umfangreich unterstützt [Microsoft 2003b].

Das SharePoint Object Model kann von Programmcode, der in ASP.NET geschrieben ist,¹⁷³ von clientseitigen Windows-Anwendungen oder auch von den Prozessen des Windows Server 2003¹⁷⁴ erreicht werden: Damit innerhalb des Programmcodes auf die Inhalte von gemeinsamen Arbeitsbereichen zugegriffen werden kann, benötigt das SharePoint Object Model den Bezug zu dem virtuellen Server. Da die eigen entwickelten Erweiterungen oft auch auf andere WSS-Installationen angewendet werden sollen, ist eine feste Zuordnung nicht angebracht. Stattdessen müssen die Eigenschaften des aktuellen HTTP-Aufrufs abgefragt werden, was das SharePoint Object Model möglich macht. Hieraus kann der virtuellen Server bzw der avisierte gemeinsame Arbeitsbereich im Programmcode ermittelt werden.

Webdienste in den WSS

Webdienste sind von einem Application Server bereitgestellte Dienste, die von Anwendungen (nicht Endbenutzern) eines anderen Rechners unter Verwendung definierter Parameter genutzt werden können. Der Grundgedanke von Webdiensten besteht darin, Anwendungslogik weltweit über das Internet zur Verfügung zu stellen. [Prosise 2002:463] geht davon aus, dass

”Webdienste eines Tages das Rückgrad eines programmierbaren Internets bilden, das [...] auch die Kommunikation zwischen Servern ermöglicht und Anwendungen von den Fesseln ihrer jeweiligen Plattform befreit“.

Genaugenommen handelt es sich bei Webdiensten um die veröffentlichten Methoden von Klassen eines Programms.¹⁷⁵ Die Veröffentlichung und der Aufruf eines Webdienstes erfolgt über eines der herkömmlichen Kommunikationsprotokolle des Internet, wie zum Beispiel HTTP, SMTP¹⁷⁶ oder MIME¹⁷⁷ [Beimborn; Mintert; Weitzel 2002:277]. An den Aufruf eines Webdienstes schließt sich die Verarbeitung der übermittelten Daten durch den Application Server an. Der Vorgang wird mit der Zurücksendung der verarbeiteten Daten an den anfragenden Client abgeschlossen.¹⁷⁸

Im Zusammenhang mit der Architektur von Webdiensten sind SOAP, WSDL und UDDI elementare Begriffe. Sie sind Teile der *serviceorientierten Architektur* (SOA), auf denen

¹⁷³Zum Beispiel ASPX-Webseiten, Webdienste und Webparts.

¹⁷⁴Der Arbeitsprozess des Internet Information Server.

¹⁷⁵[Prosise 2002] verwendet hier den Begriff der *Webmethoden*.

¹⁷⁶Das *Simple Mail Transfer Protocol* (SMTP) ist ein Protokoll des Internets, was den Versand von Emails ermöglicht.

¹⁷⁷Der Standard *Multipurpose Internet Mail Extensions* (MIME) dient der Kodierung von Inhalten von Emails und anderen Internetnachrichten [http://de.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions] (20.4.2006).

¹⁷⁸Ein asynchroner Vorgang

die Webdienste basieren [Buxmann; Wüstner 2002:1434]. SOAP, WSDL und UDDI bauen allesamt auf XML auf.

Die eXtensible Markup Language (XML) ist wie die Hypertext Markup Language (HTML) eine Auszeichnungssprache für Dokumente und ein offenes Dateiformat¹⁷⁹. Beide besitzen als Ursprung die Standard Generalized Markup Language (SGML). XML ist im Gegensatz zu HTML, das lediglich eine „Strukturbeschreibung für Web-Dokumente“ darstellt, eine *Metasprache*.¹⁸⁰ Das heißt, mit XML lassen sich „beliebig viele Sprachen wie HTML definieren“ [XMLidP], da sie Informationen *über* den Inhalt eines Dokumentes definiert. Ziel der Ausarbeitung von XML – 1996 wurde sie auf der SGML-Konferenz erstmals vorgestellt – war die Notwendigkeit, eine erweiterbare Auszeichnungssprache umzusetzen. Der entscheidende Vorteil von XML gegenüber HTML liegt in der Definierbarkeit eigener Elemente und Datenstrukturen. Da die selbst gewählten Elemente Attribute besitzen können und diese mit Werten gefüllt werden können, schürte dies die Vorstellung, „mit XML ein einheitliches [...] Datenformat zu besitzen, das sowohl maschinenlesbar als auch für menschliche Augen verständlich ist“ [XMLidP]. Müssen Daten zwischen unterschiedlichen Anwendungen interoperabel ausgetauscht werden (heterogene Umgebungen in Unternehmen) ist heute der weitgehend anerkannte Standard XML.

Die SOA besteht aus dem *Service-Provider*, der *Service-Registry* und dem *Service-Requestor* [Buxmann; Wüstner 2002:1434].

(1) WSDL. Mit der *Web Service Description Language* (WSDL) wird der Webdienst beschrieben (dessen Methoden und erwartete Parameter). Der Service-Provider stellt neben dem Webdienst das WSDL-Dokument bereit [ebenda]. Der Service-Requestor kann einen Webdienst erst dann verwenden, wenn er die WSDL-Datei verarbeitet hat.

(2) UDDI. Als Service-Registry fungiert der Verzeichnisdienst *Universal Description, Discovery, and Integration* (UDDI): UDDI dient – analog zu einem Branchenverzeichnis – der Veröffentlichung des Webdienstes im Internet mit der näheren Beschreibung des Service-Providers.

(3) SOAP. Als standardisierte Schnittstelle zu einem Webdienst wird SOAP¹⁸¹ verwendet. Damit ist es unerheblich, in welcher Programmiersprache der Webdienst entwickelt wurde, solange er SOAP-Nachrichten verarbeiten kann. Die SOAP-Nachricht, in der die Webdienst-Methode mit den dazugehörigen Parametern eingebettet ist,¹⁸² wird mit dem Aufruf an den Webserver via HTTP-Anfrage übertragen. Die Antwort wird ebenfalls in Form einer SOAP-Nachricht an den anfragenden Client zurückgesandt.

¹⁷⁹ Quelltext liegt stets im Klartext vor.

¹⁸⁰ Siehe [XMLidP] <<http://www.linkwerk.com/pub/xmlidp/2000/mehr-ist-mehr.html>>.

¹⁸¹ Eine genaue Ausschreibung von SOAP existiert nicht (<<http://de.wikipedia.org/wiki/SOAP>> (1.4.2006)).

¹⁸² Man spricht von dem SOAP-Envelope.

Die Erstellung eines Webdienstes erfordert die Verwendung einer geeigneten Programmiersprache. Der eingesetzte Application Server schränkt die verfügbare Programmiersprache meist ein.¹⁸³ Der Windows Server 2003 ist auf Grund seiner Application-Server-Funktionalität in der Lage, in einer .NET-Programmiersprache umgesetzte Webdienste auszuführen. Die in der WSS bereits enthaltenen Webdienste umfassen Funktionen für administrative Aufgaben¹⁸⁴, für die Arbeit mit Listen und deren Elementen, mit gespeicherten Dokumenten und gemeinsamen Arbeitsbereichen. Sie sind physikalisch an einer zentralen Stelle im Windows Server gespeichert. Mit Hilfe der Virtualisierung dieses Verzeichnisses können sie von jedem gemeinsamen Arbeitsbereich aufgerufen werden.

Die Entwicklungsumgebung MS Visual Studio.NET gibt eine Vielzahl von Hilfestellungen zur Erstellung eigener Webdienste.¹⁸⁵ Besteht die Aufgabe des Webdienstes darin, auf Inhalte der gemeinsamen Arbeitsbereiche zuzugreifen, ist im Programmcode das Microsoft SharePoint Object Model zu verwenden. Es lassen sich aber auch alle Funktionen des Windows Server nutzen, sofern sie über das .NET-Framework programmiertechnisch ansprechbar sind. Mit Webdiensten lassen sich nahezu alle erdenklichen Aufgaben erfüllen. Dabei ist jedoch zu bedenken, dass der Webdienst über die nötigen Rechte zur Ausführung der Aufgabe verfügen muss.

Ereignisroutinen für Dokumentbibliotheken

Die WSS erlauben es, Ereignisse,¹⁸⁶ die auf einer Dokumentbibliothek stattfinden, zu erkennen und abzufragen. Für die Ereignisse können Ereignisroutinen (*event handler*) implementiert werden, deren Programmcode automatisch bei Eintritt des Ereignisses ausgeführt wird. Ereignisse auf einer Dokumentbibliothek finden immer dann statt, wenn eine Änderung an einem Dokument der Dokumentbibliothek vorgenommen wurde. Folgende Ereignisse lösen die Ereignisroutine aus und lassen sich programmiertechnisch in der Ereignisroutine verarbeiten: (i) ein neues Dokument wurde eingefügt, (ii) ein Dokument wurde aktualisiert, (iii) ein Dokument wurde gelöscht, (iv) ein Dokument wurde verschoben,¹⁸⁷ (v) ein Dokument wurde kopiert, (vi) ein Dokument wurde freigegeben (*check in*), (vii) ein Dokument wurde gesperrt (*check out*) sowie (viii) die Sperrung des Dokuments wurde abgebrochen (*cancel check-out*).

Ereignisroutinen werden stets nach dem stattgefundenen Ereignis, das heißt asynchron, ausgeführt. Darum können die WSS während der Ausführung der Ereignisroutine neue Benutzereingaben entgegen. Das bedeutet aber auch, dass wenn ein Dokument aus

¹⁸³ Wichtige Anbieter von Application Server unterstützen meist die Java-Technologie der Firma Sun oder das .NET-Framework von Microsoft.

¹⁸⁴ Mit dem Aufruf eines WSS-Webdienstes können beispielsweise gemeinsame Arbeitsbereiche eingerichtet oder entfernt, Listen hinzugefügt oder gelöscht werden.

¹⁸⁵ Die WSDL- und DISCO-Dateien (DISCO-Dateien dienen dem UDDI zur Veröffentlichung des Dienstes im Internet) können automatisch erzeugt werden. Selbiges gilt für die .asmx-Datei, die für das Auffinden des Programmcodes durch den Windows Server wichtig ist.

¹⁸⁶ Eine (Zustands-)Änderung wird – in der Welt der Programmierung – als ein *Ereignis* verstanden.

¹⁸⁷ Das Ereignis *Dokument verschoben* deckt auch das Umbenennen eines Dokuments ab.

einer Dokumentbibliothek gelöscht wird, die Ereignisroutine den Löschvorgang nicht abbrechen oder gar rückgängig machen kann.

Die Ereignisroutinen lassen sich nach ihrer Installation an jede beliebige Dokumentbibliothek *binden*. An eine Dokumentbibliothek kann aber immer nur eine Ereignisroutine gebunden werden. Die Entwicklung der Ereignisroutine hat in einer Hochsprache des .NET-Framework zu erfolgen. Das Ergebnis der Entwicklung ist eine verwaltete-Code-Assemblierung (*managed code assembly*), die später in den GAC des Windows Servers zu installieren ist.¹⁸⁸

Im Programmcode der Ereignisroutine ist das SharePoint Object Model zu importieren und die Schnittstelle *IListEventSink* und deren Methode *OnEvent()* zu implementieren. Die von der Ereignisroutine realisierte Methode *OnEvent()* wird ohne Unterschied vom eingetretenen Ereignis ausgeführt. Der Entwickler hat in der *OnEvent()*-Methode herauszufinden, welches Ereignis auf der Dokumentbibliothek stattfand und den Programmcode entsprechend aufzubauen. Befindet sich ein Entwickler im Codeabschnitt der *OnEvent()*-Methode, hat er vollen Zugriff auf das SharePoint Object Model und kann damit umfangreiche Aufgaben lösen (siehe *Das Microsoft SharePoint Object Model*). Dabei ist zu beachten, dass Ereignisroutinen standardmäßig im Sicherheitskontext des WSS-Arbeitsprozess¹⁸⁹ ablaufen.¹⁹⁰

Webparts

In den Grundlagen ist schon Allgemeines zu Webparts ausgeführt worden (siehe 2.4.1(1)). Mit dem MS Visual Studio.NET ist es möglich, eigene Webparts zu entwickeln. Dabei gilt das zu WSS-Webdiensten und Ereignisroutinen gesagte: Webparts sind in verwaltetem Code geschrieben und haben für die Arbeit mit Inhalten aus dem gemeinsamen Arbeitsbereich das SharePoint Object Model zu verwenden.

Einige Besonderheiten sind bei der Entwicklung von Webparts zu berücksichtigen: Der Lebenszyklus eines Webparts wird von Ereignissen bestimmt, die denen von herkömmlichen Webseiten entsprechen:¹⁹¹ Initialisierung (*Init*), Laden (*Load*), Datenanbindung (*Data-Binding*), Berechnung (*PreRender*) und Schließen (*Unload*). Hat man zum Zeitpunkt des Aufrufs der Webseite vor, in dem eigen entwickelten Webpart berechnete Inhalte anzuzeigen, muss eines der genannten Ereignisse in dessen Programmcode abgefangen werden. Soll der Benutzer mit Oberflächenelementen des Webparts interagieren können (zB Knöpfe bedienen), müssen an diese Ereignisroutinen gebunden werden. Da

¹⁸⁸ Befindet sich die Assemblierung im GAC wird sie vom Application Server als sicher eingestuft.

¹⁸⁹ Der Arbeitsprozess der WSS wird mit den Benutzerrechten der Identität ausgeführt, die im Anwendungspool des virtuellen Servers eingestellt wurde.

¹⁹⁰ Ereignisroutinen sind verwalteter Code, die den Sicherheitsmechanismen des .NET-Framework unterworfen sind.

¹⁹¹ <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc_sppt2003_bk/html/WSSInsideOutCh19.asp> (21.3.2006)

die Ereignisroutinen das erneute Laden der gesamten Webseite hervorrufen, sind die im Webpart zuvor angezeigten Inhalte auf dem Server zwischenspeichern (*caching*).

Die Methoden, die verwendet werden, um die Grenzen der asynchronen Kommunikation zwischen dem clientseitigen Internet Explorer und den WSS zu durchbrechen (zB das eben genannte *Caching*), sind weder trivial noch intuitiv anzuwenden. Ist aber ein grundlegendes Verständnis der Entwicklung von Webparts erreicht, lassen sich mit ihnen viele Anwendungsfälle lösen.

Remote Procedure Calls

Von einem Client aus können entfernte Prozeduraufrufe – *Remote Procedure Calls* (RPC) – ausgeführt werden. Diese bringen Programme auf dem WSS-Server zur Ausführung.

Der Aufruf erfolgt mittels einer an den Server gestellten HTTP-Anfrage. Die Anfrage wird aus dem Servernamen, dem Pfad zu dem auszuführenden Programm und einer obligatorisch anzuhängenden Parameterliste zusammengesetzt.¹⁹²

Die von den WSS angebotenen RPCs werden von einer Programmbibliothek (owssvr.dll) bereitgestellt und basieren auf dem FrontPage-RPC-Protokoll. FrontPage-RPC selbst besteht aus dem anfragenden *FrontPage*-Client und den *FrontPage-Servererweiterungen*, die auf einem IIS-Webserver installiert sein müssen.¹⁹³

Da es sich bei RPC um eine asynchron ausgeführte Client-Server-Technologie handelt, wird sie von den MS-Office-Produkten für Aktionen wie dem Sperren von Dokumenten und dem Setzen von Dokumenteigenschaften benutzt. Das heißt, für Aufgaben, die lediglich eine eingerichtete Kommunikation vom Client zum Server erfordern.

WebDAV

Das *Web-based Distributed Authoring and Versioning* (WebDAV) ist ein „offener Standard zur Bereitstellung von Dateien im Internet“¹⁹⁴. Es handelt sich dabei um eine Erweiterung des HTTP-Protokollstandards und erlaubt das einfache Bearbeiten der auf einem Webserver gespeicherten Dateien von einer clientseitigen Desktop-Anwendung aus.¹⁹⁵ Im Windows Explorer des Microsoft-Windows-Betriebssystems sind WebDAV-Ordner als *Webordner* bekannt.

Gegenüber dem einfacheren HTTP-Protokoll verfügt WebDAV über die Befehle: *Copy* (Kopieren), *Move* (Verschieben), *Lock* (Sperren), *Unlock* (Sperre lösen), *Mkcol* (für *Make Collection*, erstelle Verzeichnis), *Propatch* (Attribute ändern) und *Propfind* (Attribute ermitteln).¹⁹⁶ In den WSS sind alle Dokumente eines gemeinsamen Arbeitsbereichs per WebDAV von einem clientseitigen Windows Explorer aus erreichbar.

¹⁹² Beispiel für einen RPC-Aufruf, wobei owssvr.dll die Programmbibliothek ist und die Parameter nach dem Fragezeichen stehen würden: http://server/_vti_bin/owssvr.dll?

¹⁹³ <<http://www-pc.uni-regensburg.de/systemsw/ÖFFICE/serk/axrpc.htm>> (2.4.2006)

¹⁹⁴ <<http://de.wikipedia.org/wiki/WebDAV>> (11.1.2008)

¹⁹⁵ Darunter Microsoft Windows, UNIX-Derivate, Mac OS X, IBM OS/2 [ebenda].

¹⁹⁶ [ebenda]

Das WebDAV-Protokoll wird im .NET-Framework von der `webclient`-Klasse implementiert und kann durch sie für Eigenentwicklungen genutzt werden. Wird die `webclient`-Klasse mit den nötigen Rechten ausgestattet, kann sie Dokumente aus Dokumentbibliotheken lesen und auch schreiben.

Die Collaborative Application Markup Language

Die von allen SharePoint-Produkten verwendete CAML dient der Festlegung aller sichtbaren Aspekte¹⁹⁷ einer WSS-Seite [Hillier 2005:1].¹⁹⁸ Die CAML gehört zur Gruppe des *nicht verwalteten Codes* und wird von Programmbibliotheken, die ebenfalls aus nicht verwaltetem Code bestehen, interpretiert.

Soll eine neue Vorlage für WSS-Seiten erstellt werden – das können Änderungen am Layout und an der Zusammenstellung der verwendeten Webparts sein –, so ist eine Bearbeitung der in CAML verfassten Seiten-Vorlagen nötig. Mit der Beherrschung von CAML sind weitreichende Anpassungen von Sichten auf den Inhalt von gemeinsamen Arbeitsbereichen möglich.¹⁹⁹ Jedoch kann die WSS-Anwendung bei unsachgemäßer Bearbeitung der CAML-Vorlagen zerstört werden [Hillier 2005:1].

Änderungen an den CAML-Vorlagen betreffen stets die gesamte WSS-Anwendung mit all ihren auf den virtuellen Servern installierten Root-Webseiten, denn die Vorlagen sind in einem zentralen Verzeichnis des Windows Servers gespeichert und werden zum Zeitpunkt des Aufrufs der Webseite interpretiert. Eine sichere Abgrenzung der Vorlagen ist nur über die Erstellung neuer Seiten-Vorlagen möglich. Zudem findet eine Unterteilung über die sogenannte Lokalisierung statt: Die CAML-Vorlagen sind in unterschiedlichen, kulturabhängigen, Verzeichnissen gespeichert.²⁰⁰

CAML ist also eine Möglichkeit Inhalte in angepasster Darstellungsweise auszugeben. Das betrifft die Webseiten, aber auch die Listen und Webparts. Es ist mit CAML nicht möglich, neuen Inhalt in die Inhaltsdatenbank zu speichern oder bestehenden zu modifizieren.

Datenbankoperationen mittels SQL

Die WSS benötigen ein Microsoft-SQL-Datenbanksystem für die Konfigurations- und Inhaltsdatenbanken der jeweiligen virtuellen Server (siehe 2.4.3). Das kann entweder ein bereits installierter MS SQL 2000 Server, ein MS SQL 2005 Server oder die kostenfrei mit ausgelieferte WMSDE sein.

¹⁹⁷ Das sind die anzuzeigenden Daten und deren Darstellung.

¹⁹⁸ Siehe hierzu auch 2.4.3

¹⁹⁹ Anschauliche Beispiele dafür sind die Seiten von Ian Morrish (<<http://www.wssdemo.com/default.aspx>>) und Mike Walsh (<<http://www.asaris-matrix.com/mvp-walsh/default.aspx>>).

²⁰⁰ Für alle deutschsprachigen Vorlagen wäre dies `C:\Programme\Gemeinsame Dateien\Microsoft Shared\web server extensions\60\TEMPLATE\1031\`

Mit der Datenbanksprache SQL werden Inhalte in die Datenbanken hineingeschrieben und ausgelesen. Der direkte Zugriff auf die Inhaltsdatenbanken der WSS eignet sich jedoch nur für lesende Aktivitäten. Sowohl MS als auch die einschlägige Literatur empfehlen ausdrücklich, von schreibenden Aktionen abzusehen [Hillier 2005:51]. Wird dennoch in die Datenbank unsachgemäß geschrieben, kann die Zerstörung der WSS-Installation die Folge sein. Unbedenklich sind jedoch Aktionen in anderen Datenbanken als in den von den WSS angelegten.²⁰¹

Die Verwendung von SQL für die Arbeit mit Inhalten von gemeinsamen Arbeitsbereichen eignet sich also nicht. Sie kann aber in bestimmten Fällen sinnvoll sein, in denen auch das SharePoint Object Model Grenzen hat. Als Beispiel führt [Hillier 2005:51f] hier die Anzeige aller Dokumente einer WSS-Installation an, was mit anderen Technologien als SQL nicht zu bewerkstelligen ist. Ansonsten sollte stets auf das SharePoint Object Model zurückgegriffen werden.

JavaScript

Eine Skriptsprache ist eine Programmiersprache, die nicht mehr kompiliert werden muss, sondern lediglich *interpretiert* wird. Skriptsprachen können entweder vom Server²⁰² oder von einem Client²⁰³ verarbeitet werden. Sie sind für kleinere programmiertechnische Aufgaben gedacht, weshalb ihr Quellcode zur schnelleren Bearbeitung immer im Klartext vorliegt.²⁰⁴

Soll die Benutzbarkeit der ansonsten statischen Oberfläche von Webseiten mit dynamischen Menüs aufgewertet werden, wird zumeist JavaScript eingesetzt.²⁰⁵ Der Scriptcode wird zusammen mit der Webseite als Antwort auf die HTTP-Anfrage an den anfragenden Client-PCs gesandt und von dessen Browser interpretiert.

In den WSS wird JavaScript zum Beispiel für die Realisierung der Kontextmenüs für Listeneinträge eingesetzt (siehe das Beispiel in 2.4.5). Der CAML-Code, der bei einer Webseiten-Anfrage zuerst interpretiert wird, erzeugt den JavaScript-Code. Das ist insofern hilfreich, als dass der CAML-Code in lokalisierten Vorlagen in zentralen Verzeichnissen hinterlegt ist (siehe *Die Collaborative Application Markup Language*) und damit die Anpassung des JavaScript an die Sprache der WSS-Webseite vornehmen kann.²⁰⁶

Eine interessante in JavaScript umgesetzte Funktion existiert für den Dokumentarbeitsbereich. Für jedes Dokument einer Dokumentbibliothek kann ein separater Dokumentarbeitsbereich erstellt werden (siehe Abbildung 27). In diesem besitzt das Doku-

²⁰¹ Ein Datenbanksystem kann mehrere Datenbanken verwalten.

²⁰² Beispiele für serverseitige Skriptsprachen sind Perl, PHP und Python.

²⁰³ JavaScript und VBScript sind Beispiele für clientseitige Skriptsprachen.

²⁰⁴ <<http://de.wikipedia.org/wiki/Skriptsprachen>> (2.4.2006)

²⁰⁵ Der Wikipedia-Artikel zu JavaScript führt neben typischen Anwendungsfällen wie Popup-Werbung, Laufschriften, Hervorhebungen des Mauszeigers oder von Schaltflächen als interessanten Punkt die Erweiterung der Bedienbarkeit von Webseiten an.

²⁰⁶ Die mit JavaScript realisierten Kontextmenüs sind an die jeweilige Sprache des gemeinsamen Arbeitsbereichs angepasst.

ment, das Hauptgegenstand des Arbeitsbereiches ist, ein spezielles Kontextmenü, das die Funktion *Am Quellort veröffentlichen* (eng *Publish to Source Location*) zur Verfügung stellt. Mit ihr kann dieses gemeinsam bearbeitete Dokument aus dem Dokumentarbeitsbereich an die übergeordnete Dokumentbibliothek zurückgesandt werden und erzeugt dort eine neue Version.

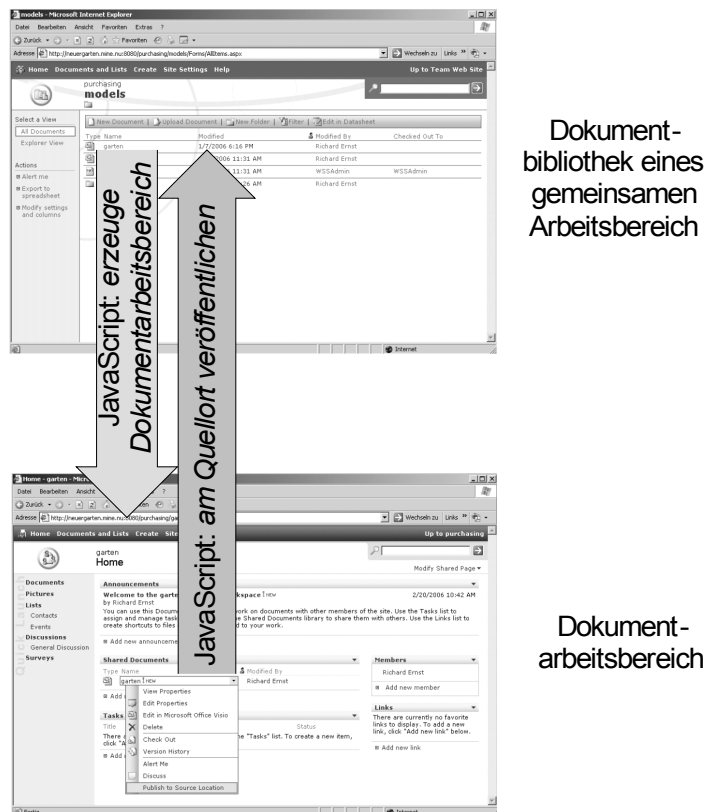


Abbildung 27. JavaScript-Funktion: *Am Quellort veröffentlichen*.

Mit ausreichender Kenntnis von JavaScript und CAML ließen sich mit der Anpassung einer Seiten-Vorlage in den darin enthaltenen Kontextmenüs neue Funktionen realisieren.

Gesamtübersicht der Technologien

Die bisher beschriebenen Technologien haben ihre Möglichkeiten und Grenzen. Entwickler von SharePoint-Erweiterungen zeigen in ihren Präsentationen²⁰⁷ darum oft eine Übersicht, die der in Tafel 9 ähnelt. Ausgehend von der Arbeit mit den Inhalten von Dokumentbibliotheken werden die vorhandenen Möglichkeiten den Technologien gegenübergestellt, die in den WSS verfügbar sind. In Abhängigkeit der zu lösenden Aufgabe muss daraus das Portfolio der für die Umsetzung zu nutzenden Technologien ausgewählt werden.

²⁰⁷ Diese sind meist nur den Teilnehmern von Lehrgängen zugänglich und nicht im Internet veröffentlicht. Ein Beispiel kann unter http://www.munichdot.net/Downloads/Downloads_GetFile.aspx?id=338 (21.3.2006) heruntergeladen werden.

Technologie	entfernter Zugriff	Zugriff auf Meta-daten ²⁰⁸	Hochladen mehrerer Dateien ²⁰⁹	Dateien schreiben/lesen	Versionsverwaltung	Dokument sperren / Dokument freigeben
Microsoft SharePoint Object Model	- ²¹⁰	+	-	+	+	+
Vorhandene WSS-Webdienste	+	+	-	-	-	-
WebDAV	+	-	-	+	-	-
RPC	+	+	+	+	+	+
CAML	-	-	-	-	-	-
SQL	- ²¹¹	+	-	-	-	-
JavaScript	+	-	-	-	+	+

Tafel 9

3.3.2 Distribution-Index

Bei der ersten Gegenüberstellung der Anforderungen mit dem standardmäßigen Funktionsumfang und den Möglichkeiten der WSS ergaben sich mehrere noch unbeantwortete Fragen, deren Schwerpunkt auf der *Prüfung*, *Herstellung* und *Wahrung* von übergreifender Konsistenz in einem Modellierungsprojekt verteilter Modelle liegt (siehe 3.2.3).

In 3.2.3(1) *Überprüfung von Konsistenz* sind die von SemTalk bereitgestellten Mechanismen zur Prüfung auf projektübergreifende Konsistenz vorgestellt worden. Der Entwurf und die Umsetzung einer diesbezüglichen Lösung für die Plattform auf Basis der WSS muss zukünftigen Arbeiten vorbehalten bleiben.

Die Herstellung von Konsistenz zwischen verteilten Modellen übernimmt SemTalk als clientseitiges Modellierungswerkzeug. Dazu ist SemTalk an die Bedingungen, die im Zusammenhang mit den WSS gelten, anzupassen.

Ist die Konsistenz über mehrere Modelle mit Hilfe von externen Referenzen hergestellt, liegt die Herausforderung in deren Wahrung. Darum muss genauer untersucht werden, was bei der Erzeugung von externen Referenzen passiert und worin die daraus resultierenden Probleme bestehen: Wie in 2.1.1(2) *Die externe Referenz* festgestellt, wird nur dem referenzierenden Objekt mitgeteilt, dass es sich ab nun mit einem externen Objekt

²⁰⁸ Metadaten sind Informationen *über* Dokumente, die dem jeweiligen Dokument zugeordnet sind.

²⁰⁹ Mehrere Dokumente in einem Vorgang hochladen.

²¹⁰ Das Object Model gilt immer nur auf dem Rechner, der die Anwendung ausführt (Windows PC oder Windows Server).

²¹¹ Nicht auf Inhalte der Konfigurations- oder der Inhaltsdatenbanken der WSS.

(Referenzobjekt) abzugleichen hat. Will der Modellierer die referenzierenden Objekte auf einem aktuellen Stand halten, hat er von sich aus die SemTalk-Funktion *Aktualisieren* aufzurufen. Da das Referenzobjekt bzw dessen Modell nicht über den Sachverhalt der Referenz informiert ist, kann von dem Modell keine Nachricht über geänderte Eigenschaften des Referenzobjekts oder dessen Löschung an das Modell des referenzierenden Objekts (oder dessen Eigentümer) gesendet werden.²¹² Demzufolge kann, wie in Abbildung 28 gezeigt, bei der externen Referenz von einem gerichteten Graphen gesprochen werden.

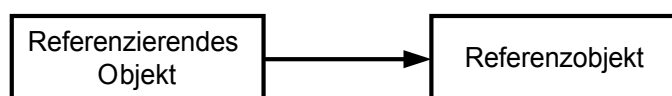


Abbildung 28. Die externe Referenz als gerichteter Graph

Eine ineffiziente und damit unpraktische Lösung bestünde darin, dem Referenzobjekt die Information über die externe Referenz mitzuteilen. In einem Szenario der verteilten Modellierung, in dem sich Objekte vieler separater Modelle auf *ein und dasselbe* externe Objekt beziehen, würden die im Modell des Referenzobjekts zu speichernden Informationen schnell schwer handhabbar werden.²¹³ Für jedes Erzeugen einer neuen externen Referenz mittels Schreibzugriffen auf die XML-Datei des einen Modells zugegriffen werden. Bei diesem Vorgang fänden keinerlei Maßnahmen zur Transaktionskontrolle²¹⁴ statt und konkurrierende Zugriffe auf die Datei könnten diese unbrauchbar machen.

Eine andere Lösung für dieses Problem besteht darin, eine dritte Instanz einzuführen. Diese speichert Informationen über alle externen Referenzen, die zwischen Modellen einer Dokumentbibliothek existieren. Würde diese dritte Instanz, der Distribution-Index, von den WSS automatisch verwaltet, müsste der Programmcode von SemTalk nicht verändert werden.

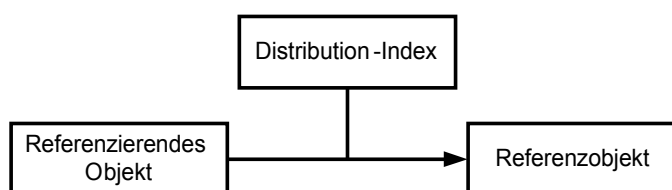


Abbildung 29. Distribution-Index mit Informationen über sämtliche externen Referenzen

Die Aktualität und Gültigkeit des Distribution-Indexes ist mit seiner ständigen Erreichbarkeit gewährleistet: Der Distribution-Index ist für eine Dokumentbibliothek verantwortlich und kann von jedem Modell aus aktualisiert werden. Der Inhalt des Distribution-Indexes kann damit als Repräsentation des Zustands der externen Referenzen zwischen –

²¹² Dieser Umstand macht die *zyklisch referenzierten Objekte* in 2.1.3 erst möglich.

²¹³ Man vergleiche einen Automobil-Hersteller, der das unternehmensweite Konzept „Auto“ bereitstellt, auf das sich mehrere Hundert Ingenieure beziehen.

²¹⁴ Sichert das erfolgreiche Durchführen von Operationen in einem System nebenläufiger Operationen.

in einer Dokumentbibliothek gespeicherten – SemTalk-Modellen angesehen werden. Folgende Überlegungen schließen sich daran an: (i) Wie und wann werden die Informationen über eine externe Referenz aus einem Modell erhoben und in dem Distribution-Index gespeichert, (ii) wie und wann werden Veränderungen am Referenzobjekt in dem Distribution-Index berücksichtigt und schließlich (iii) wie und wann erhält der Eigentümer des Modells, welches das referenzierende Objekt beherbergt, eine Mitteilung über die stattgefundenen Veränderungen.

Die hier vorgestellte Lösung erinnert an ein typisches Datenbank-Schema. Auch die Problematik lässt diese Analogie zu: Zwischen zwei in Beziehung stehenden Entitäten (Objekten) wird eine Relation (externe Referenz) aufgebaut. Die Information über diese Beziehung wird im Fall der Datenbank nur in der Relation-Tabelle gespeichert, was sehr effizient ist. Die zwei Entitäten-Tabellen wissen nichts von der Beziehung zwischen ihnen. Im Bereich von SemTalk handelt es sich bei den in Beziehung stehenden Entitäten um Objekte separater Modelle. Anstelle der Relation wird hier eine neue Datei – der Distribution-Index – eingeführt. Sie ist sowohl vom Modell des referenzierenden Objekts, als auch vom Modell des Referenzobjekts aus erreichbar.

Finden zu einem Zeitpunkt Änderungen an den im Distribution-Index gespeicherten Informationen statt, ließen sich Funktionen auslösen, die den Anforderungen an die Wahrung von projektübergreifender Konsistenz gerecht werden: Der Eigentümer eines Modells, das ein Objekt mit einer externen Referenz beinhaltet, wird über Änderungen an dem referenzierten Modell oder dem Referenzobjekt umgehend informiert. Mit Kenntnis dieses Sachverhalts hat er die Möglichkeit – bewusst – die Aktualisierung oder die gegebenenfalls nötige Korrektur seines Modells vorzunehmen.

Im Folgenden wird die Realisierung und die Verwaltung des Distribution-Indexes unter Verwendung eines Mix aus den in 3.3.1 vorgestellten Technologien entworfen. Dazu werden im ersten Schritt Überlegungen zu Art und Struktur des Distribution-Indexes ausgeführt.

(1) Art des Distribution-Indexes. Für eine Verwaltung der externen Referenzen durch die WSS bieten sich zwei Lösungen an – entweder basiert der Distribution-Index auf einer Datenbank oder auf einer Datei. Die erste Variante erfordert eine neue Datenbank in dem Datenbanksystem, das auch für die WSS benötigt wird. Diese ließe sich mit der Datenbankabfragesprache SQL pflegen. Ein wesentlicher Grund spricht gegen diese Variante und für eine Datei-basierte Speicherung: Da die WSS bereits mit dem kostenfreien Datenbanksystem WMSDE ausgeliefert werden, ist nicht immer damit zu rechnen, dass bei dem späteren Nutzer ein MS SQL Server eingesetzt wird (siehe 2.4.3). Die WMSDE ist in ihrer Funktion stark beschränkt und erlaubt kaum manipulative Möglichkeiten für den Entwickler. Und da das problemlose Schreiben in die vorhandene Datenbank(en) der WSS ausgeschlossen ist, wird von einer Datenbank-Lösung abgesehen.

Die Datei-basierte Lösung sieht vor, eine Datei in der Dokumentbibliothek zu erzeugen, die Informationen strukturiert speichern kann. Das gewählte Dateiformat muss durch einfache Lese- und Schreibzugriffe zugänglich sein.²¹⁵ Die Strukturierung von Daten ist nötig, um ein Datenbank-ähnliche Speicherung sicherzustellen. Das in dieser Arbeit schon oft angesprochene Dateiformat XML (siehe 2.1.1(2) und 3.3.1 Webdienste in den WSS) ist unter diesen Gesichtspunkten das Geeignete.

(2) Struktur des Distribution-Indexes. Es müssen Informationen über beide an der externen Referenz beteiligten Objekte gespeichert werden. Notwendig sind hierbei die Objekt-IDs des referenzierenden und des Referenzobjekts sowie die Pfade zu ihren Modellen. Diese vier Angaben erlauben die eindeutige Identifikation der externen Referenz.

Für jeden Eintrag, der stellvertretend für eine externe Referenz im Distribution-Index steht, ist ein Status erforderlich. Dieser gibt Auskunft darüber, ob die externe Referenz gültig oder ungültig ist. Er wird dann abgefragt, wenn ein Modell in die verwaltete Dokumentbibliothek gespeichert wird. Damit soll verhindert werden, dass ein Modell mit einer als ungültig markierten externen Referenz wiederholt in der Dokumentbibliothek gespeichert wird.

Ist zu einem späteren Zeitpunkt gewünscht, dass über die Änderung des Vorhandenseins und der Bezeichnung des Referenzobjekts hinaus Mitteilungen auch bei Änderungen von Methoden oder Attributen am Referenzobjekt an die Eigentümer referenzierender Objekte gesendet werden, müssen diese Aspekte der zwei beteiligten Objekte mit in den Distribution-Index aufgenommen werden. Der im Listing 4 gezeigte Inhalt des Distribution-Index ist in Abbildung 30 in Form eines umgedrehten Baums dargestellt. In ihm wird die Struktur klar: Das Wurzelement (*Distribution*) steht in der obersten Hierarchiestufe gefolgt von den Kindelementen für die externen Referenzen (*References*) und den externen Verfeinerungen (*Refinements*). Der einzelne sich auf eine externe Referenz beziehende Eintrag im Distribution-Index wird mit dem *Reference*-Element umgesetzt. Neben den darin befindlichen Kindelementen *Source* und *Destination* hat es das Attribut *state*, was den Status der Gültigkeit der externen Referenz wieder spiegelt. Das *Source*-Element repräsentiert das referenzierende Objekt, mit seiner Bezeichnung (*ReferringObject*-Attribut), dem Pfad zu dem beinhaltenden Modell (*path*) und schließlich seiner Objekt-ID (*id*). Die Informationen zu dem Referenzobjekt werden in dem *Destination*-Element gespeichert. Dessen Attribute werden analog zu denen des *Source*-Elements gefüllt.

²¹⁵ Man spricht von einem *offenen* Dateiformat.

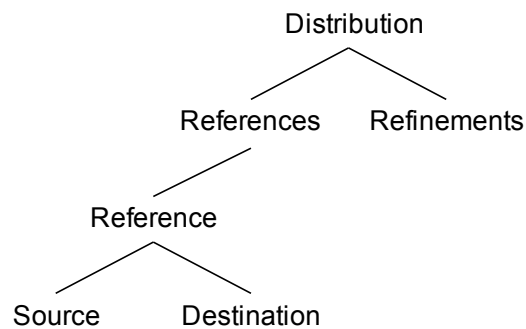


Abbildung 30. Struktur-Baum des Distribution-Index

```

<Distribution>
  <References>
    <Reference state="1">
      <Source ReferringObject="In#Strauch"
        path="Grünanlage.xml" id="2404" />
      <Destination ReferenceObject="In#Strauch"
        path="Garten.xml" id="2413" />
    </Reference>
  </References>
  <Refinements />
</Distribution>
  
```

Listing 4

3.3.3 Aktualisierung des Distribution-Indexes

Das hier vorgestellte, zugegebener Maßen prototypische Szenario zur Unterstützung der verteilten Modellierung mit Hilfe des Distribution-Indexes sieht vor, dass sich alle Modelle des verteilten Modellierungsprojektes in einer Dokumentbibliothek eines gemeinsamen Arbeitsbereiches befinden. Es ist nicht möglich, untergeordnete Dokumentarbeitsbereiche²¹⁶ für die einzelnen Modelle einzurichten, mehrere parallel existierende Dokumentbibliotheken zu nutzen oder gar verteilte Modelle über mehrere virtuelle Server herzustellen. Die mögliche Größe des unterstützten Modellierungsprojekts bleibt somit begrenzt. Welche Möglichkeiten existieren, wird aber angedeutet.

Im Verlauf der Entwurfsphase zu dieser Arbeit sind weitere Szenarien untersucht worden. Diese sahen unter anderem auch die Unterstützung von Dokumentarbeitsbereichen und der Nutzung der Veröffentlichungs-Funktion von Dokumenten vor (siehe Abschnitt JavaScript in 3.3.1). Um aber die Durchführbarkeit der Arbeit nicht durch einen hohen Grad

²¹⁶ Diese könnten es einem Team von Modellierern erlauben, einen Bereich nur für sich und das zu bearbeitende Modell einzurichten. Die Modelle anderer Teams wären in anderen gemeinsamen Arbeitsbereichen.

an Komplexität zu gefährden, wurde auf die weiterführende Ausarbeitung dieser Szenarien verzichtet.

Das Ziel, dass sich der Modellierer die funktionalen Mehrwerte nicht mit zusätzlichem Bedienungsaufwand erkaufen muss (siehe 3.3), sondern die dafür notwendigen Schritte vorrangig automatisiert im Hintergrund ausgeführt werden, kann mit der Verwendung einer Ereignisroutine erreicht werden. Die Aktualisierung des Distribution-Indexes sollte dann erfolgen, wenn eine Änderung (siehe *Ereignisroutinen* in 3.3.1) einer Modell-Datei in einer Dokumentbibliothek stattgefunden hat. So bleibt der Vorgang der Aktualisierung des Distribution-Indexes für den Modellierer transparent.

Für die nun folgenden Ausführungen zum groben Ablauf der Aktualisierung des Distribution-Indexes ist Anhang IV zur Hand zu nehmen. Der dort gezeigte Aktualisierungsprozess ist mit SemTalk in der Modellierungsmethode Kommunikationsstrukturanalyse (KSA-Methode) modelliert.²¹⁷ Gezeigt wird, was passiert, wenn in dem betrachteten Modell ein Referenzobjekt gelöscht wurde und dieses Modell in der Dokumentbibliothek eines gemeinsamen Arbeitsbereichs gespeichert wird. In dem folgenden Text werden nummerierte Aufgaben erwähnt, die sich auf die Nummern in den Prozesselementen beziehen.

Wie der Abbildung zur *Aktualisierung des Distribution-Indexes* zu entnehmen ist, sind an dem gesamten Vorgang der Aktualisierung fünf Akteure beteiligt: Eine Person (der Modellierer) und vier elektronische Akteure. Clientseitig sind hier *MS Visio* und das *SemTalk-Add-In*²¹⁸ für MS Visio zu finden; die Ereignisroutine *ModellLibEvtHandler* und der *semtalk-Webdienst* agieren vonseiten des Windows Server.

MS Visio muss als Haupt-Anwendung auf dem Rechner des Modellierers ausgeführt werden. Die Vorteile, die sich mit der Microsoft Office System Integration (siehe 2.4.5) ergeben, machen dies erforderlich. Ausschlaggebend ist die Fähigkeit von MS Visio zu erkennen, dass es sich bei einem zu öffnenden Dokument um ein Dokument einer Dokumentbibliothek der WSS handelt. Es schließt sich die Begründung an, die für MS Visio und nicht SemTalk als Anwendung spricht.

(1) Der Freigegebene Arbeitsbereich. Diese zusätzliche Bedienungsfläche, die alle wichtigen Funktionen für die gemeinsame Zusammenarbeit in einer clientseitigen Anwendung vereint, ist nur in MS Visio oder anderen MS-Office-Produkten verfügbar. Entwickler können diese Oberfläche, im Gegensatz zu anderen MS Office-Funktionen, nicht in ihren Anwendungen verwenden.

(2) Zugriffsrechte auf Dokumente. Es ist für den Zugriff auf Dokumente in gemeinsamen Arbeitsbereichen nötig, die Benutzerrechte der zugreifenden Person zu über-

²¹⁷ Mehr zur KSA-Methode kann in der KSA-spezifischen Dokumentation zu SemTalk nachgelesen werden: Tutorial SemTalk Version 2.1 KSA Edition. <<http://www.semtalk.com/pub/tutsem21ksag.pdf>> (14.3.2006)

²¹⁸ Add-In: Von einer Standard-Software zusätzlich zu ladende Software.

mitteln. MS Visio fragt die Benutzerdaten des Modellierers ab, sobald es feststellt, dass es sich bei dem zu öffnenden Modell um ein Dokument einer Dokumentbibliothek handelt. Nur wenn der Modellierer über ausreichende Rechte verfügt, wird das Dokument heruntergeladen und in MS Visio angezeigt.²¹⁹

(3) Direktes Bearbeiten. Das Öffnen, Bearbeiten und direkte Speichern von Modellen in einer Dokumentbibliothek ist nur möglich, wenn MS Visio als Anwendung und SemTalk als Add-In ausgeführt wird. Die *alleinstehende* SemTalk-Anwendung ist nur in der Lage, Modelle, die auf einem Webserver gespeichert sind, herunterzuladen und lokal zu sichern.

Da SemTalk derzeit als alleinstehende Anwendung entwickelt wird,²²⁰ muss ein früheres Szenario neu aufgelegt werden: MS Visio hat SemTalk als zusätzliches Softwaremodul (Add-In) zu laden, sobald ein Modell oder eine Modellvorlage geöffnet wird. Damit lassen sich die Funktionen beider Programme nutzen.²²¹

Die modularisierte Entwicklung der *serverseitig* beteiligten Softwarekomponenten wird ermöglicht, indem ein separat zu entwickelnder Webdienst die eigentliche Aktualisierung des Distribution-Indexes vornimmt. Der ModellLibEvtHndler ruft – in Abhängigkeit des ermittelten Ereignisses – den semtalk-Webdienst auf.

Der Prozess der Aktualisierung geht, wie der Abbildung *Aktualisierung des Distribution-Indexes* zu entnehmen ist, von einer von fünf möglichen Aktivitäten des Modellierers aus: (i) Löschung eines Modells aus der Dokumentbibliothek, (ii) Öffnen eines Modells, (iii) Modellierer erstellt ein neues Modell, (iv) ein Modell wird innerhalb eines gemeinsamen Arbeitsbereich verschoben und schließlich (v) kann ein Modell umbenannt werden. Da der Modellierer die Aktivität der Umbenennung, der Verschiebung und der Löschung eines Modells immer direkt auf der Dokumentbibliothek vornimmt, tritt sofort ein Ereignis in der Dokumentbibliothek ein und der ModellLibEvtHndler wird ausgeführt. Bis ein Ereignis bei der Öffnung eines bestehenden oder der Erzeugung eines neuen Modells ausgelöst wird, laufen im Vorfeld weitere Aktivitäten ab: Nachdem der Modellierer ein Modell mit MS Visio geöffnet hat (Aktivität 2), erkennt Visio, dass es sich bei dem Modell um ein Dokument aus einer Dokumentbibliothek handelt (Aktivität 10). Infolgedessen wird neben dem Starten des SemTalk-Add-In der Freigegebene Arbeitsbereich eingeblendet. Nun könnte eine externe Referenz, die von einem der Objekte des gerade geöffneten Modells ausgeht, nicht mehr korrekt sein. Dann sollte dieser Sachverhalt dem Modellierer mitgeteilt werden.²²² Der Freigegebene Arbeitsbereich verfügt über eine An-

²¹⁹ Für Speichervorgänge gemeinsamer Dokumente hat der Modellierer über Schreibrechte zu verfügen.

²²⁰ Da SemTalk Komponenten von MS Visio benötigt, muss MS Visio auf dem Rechner installiert sein.

²²¹ MS Visio: Microsoft Office System Integration; SemTalk: Modellierungsmethoden, verteilte Modellierung

²²² Dies wäre ein Mehrwert für die verteilte Modellierung, der sich aus dem Einsatz der WSS als Plattform ergäbe.

sicht der Aufgabenliste des gemeinsamen Arbeitsbereichs. Hier könnte nun eine Aufgabe angezeigt werden, die entsprechend der Ursache für die Ungültigkeit der externen Referenz (siehe 3.2.3(2)) von dem semtalk-Webdienst generiert wurde (Aufgabe 14). Der Aufgabentext gibt dem Modellierer eine Hilfestellung, wie er die externe Referenz wieder in einen gültigen Zustand versetzen kann. Hat er das Problem beseitigt (Aufgabe 15), kann er die Aufgabe aus der Aufgabenliste löschen (Aufgabe 16) und das Modell weiter bearbeiten (Aufgabe 11). Sind hingegen alle externen Referenzen in dem soeben geöffneten Modell gültig oder existieren in dem Modell gar keine externen Referenzen, dann kann der Modellierer sofort nach der Öffnung des Modells mit der Modellierung fortfahren (Aufgabe 10 und 11).

Ist ein Modell neu erzeugt worden (Aufgabe 3), arbeitet MS Visio zunächst noch auf dem lokalen Rechner. Das bedeutet, es hat noch keine Kenntnis von dem gemeinsamen Arbeitsbereich. Der Modellierer beginnt mit der Modellierung (Aufgabe 11). Hat er einen Zustand erreicht, in dem er das Modell sichern möchte, wählt er die Dokumentbibliothek in dem gemeinsamen Arbeitsbereich als Speicherort aus. MS Visio erkennt nun, dass es sich um eine Dokumentbibliothek handelt. Demnach passieren im Fall der Speicherung eines existierenden Modells und eines vollkommen neuen Modells in einer Dokumentbibliothek die gleichen Dinge: MS Visio speichert das Modell in die Dokumentbibliothek (Aufgabe 12) und das SemTalk-Add-In die zu dem Modell korrespondierende XML-Datei in den Ordner XmlFiles (Aktivität 13) der gleichen Dokumentbibliothek.

Systembedingt wird – unabhängig von der ursprünglichen Aktion des Modellierers – ab dem Zeitpunkt der Änderung des Modells die Ereignisroutine ModelLibEvtHandler ausgeführt (Aktivität 6).²²³ Von dem ModelLibEvtHandler müssen nach der Feststellung des eingetretenen Ereignisses, davon abhängige Aktivitäten ausgeführt werden. Der ModelLibEvtHandler ruft aber in jedem Fall den semtalk-Webdienst auf. Dieser aktualisiert dann den Distribution-Index (Aufgabe 7), stellt den Eigentümern von Modellen gegebenenfalls Mitteilungen zu (Aufgabe 8) und erzeugt eine Aufgabe in der Aufgabenliste des gemeinsamen Arbeitsbereichs (Aufgabe 9). Damit ist der Kreislauf zwischen der Erstellung einer Aufgabe (Aufgabe 9) sowie deren Bearbeitung und Entfernung (Aufgaben 14, 15 und 16) hergestellt.

Bevor die Ereignisverarbeitung und die Aktualisierung des Distribution-Indexes anhand eines UML-Sequenzdiagramms detaillierter ausgeführt wird, werden die umzusetzenden Softwarekomponenten in dem UML-Klassenmodell vorgestellt.

Die Klassen der neuen Komponenten

Für das oben beschriebene prototypische Szenario ist eine Ereignisroutine und ein Webdienst zu entwickeln. Abbildung 31 zeigt die Klassen in der UML-Notation. Die Ereignis-

²²³Die an eine Dokumentbibliothek gebundene Ereignisroutine wird in jedem Fall ausgeführt (siehe *Ereignisroutinen* in 3.3.1).

routine erhält den Namensraum `ModelLibEvtHndler`. Da für die zu erledigenden Aufgaben eine Klasse genügt, soll sie `EventHandler` heißen. Die Klasse `EventHandler` hat die von dem Microsoft SharePoint Object Model bereitgestellte Schnittstelle `IListEventSink` zu implementieren (siehe 3.3.1 *Ereignisroutinen*). `IListEventSink` stellt die Methode `OnEvent()` bereit, die immer dann ausgeführt wird, wenn auf der verbundenen Dokumentbibliothek ein Ereignis stattfindet.

Der Namensraum `semtalk` gilt für alle Klassen des zu realisierenden Webdienstes. Von der Klasse `ModelLibEvtHndler.EventHandler` wird, wie im oben genannten groben Entwurf festgestellt, der Webdienst – repräsentiert durch die Klasse `semtalk.DistributionIndex` – angestoßen. Die Klasse `DistributionIndex` hat neben der öffentlichen Webmethode des `semtalk`-Webdienstes eine Steuerungsfunktion: Sie erzeugt die Objekte der Klassen `ReferencesManager` (Verwaltung von externen Referenzen) sowie `RefinementsManager` (Verwaltung externer Verfeinerungen). Von diesen ausgehend werden gegebenenfalls Objekte der `Email`- und der `TaskList`-Klasse instantiiert.²²⁴

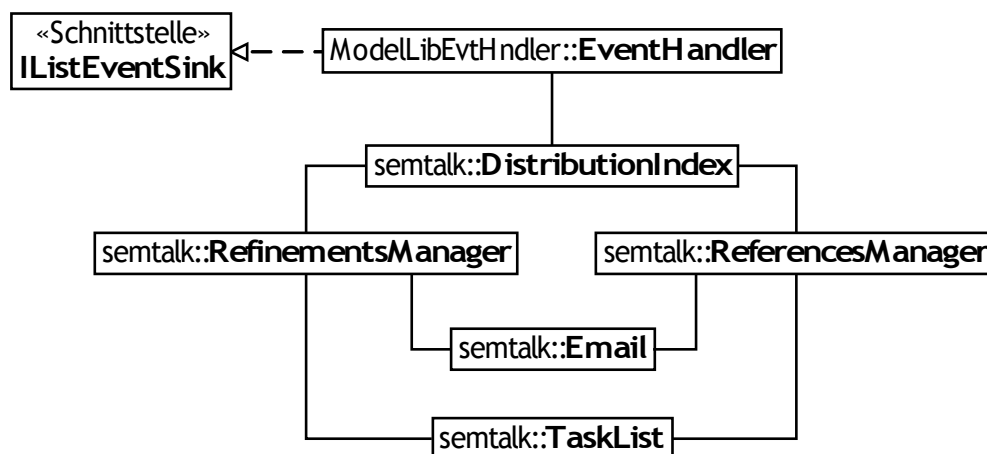


Abbildung 31. Klassendiagramm der Ereignisroutine und des `semtalk`-Webdienstes

(1) ModelLibEvtHndler.EventHandler – die Ereignisroutine. Die `EventHandler`-Klasse realisiert die `OnEvent()`-Methode. Sie hat im Wesentlichen die Aufgabe, den `semtalk`-Webdienst aufzurufen (`CallDistributionIndexService()`-Methode). Zuvor muss in der `OnEvent()`-Methode die Art des Ereignisses ermittelt werden. Dabei sind die im vorangegangenen Teil beschriebenen Aktivitäten des Modellierers zu berücksichtigen.²²⁵ In den Fällen der erstmaligen und der wiederholten Speicherung wird nur der `semtalk`-Webdienst mit dem Pfad zu der XML-Datei als Parameter aufgerufen. Ist hingegen eine Modell-Datei gelöscht, verschoben oder umbenannt worden, wird zuerst die

²²⁴ Je nach dem eine Benachrichtigung des Eigentümers stattfinden soll.

²²⁵ Das Löschen, Umbenennen, Verschieben, einfache Speichern und das erstmalige Speichern einer Modell-Datei.

zu dem Modell korrespondierende XML-Datei gelöscht und erst anschließend wird der semtalk-Webdienst aufgerufen. Ist der semtalk-Webdienst instantiiert, wird das EventHandler-Objekt terminiert.

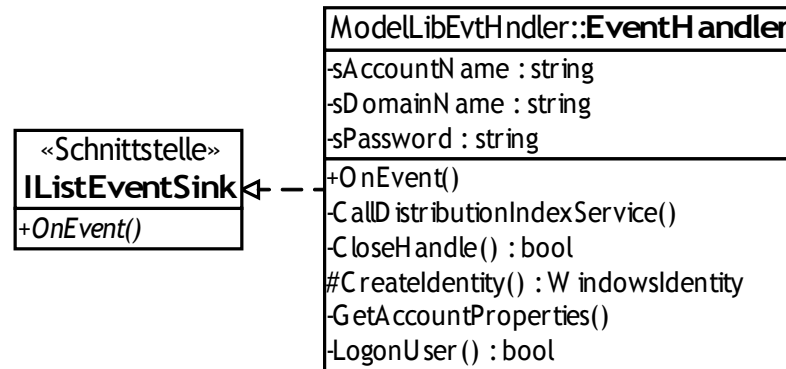


Abbildung 32. Klasse `ModelLibEvtHndler.EventHandler` und Schnittstelle `IListEventSink`

Alle Eigenschaften der `EventHandler`-Klasse und die Methoden `CloseHandle()`, `CreateIdentity()`, `GetAccountProperties()` und `LogonUser()` werden im Teil 4.4 *Der Webdienst und die Codezugriffssicherheit* näher erläutert.

(2) Die DistributionIndex-Klasse. Als Steuerungsobjekt des semtalk-Webdienstes hat das `DistributionIndex`-Objekt vor der Instantiierung der `ReferencesManager`-Klasse die Ausgangssituation zu überprüfen (`UpdateIndex()`). Gab es in der Dokumentbibliothek noch keine `Distribution-Index-Datei`, wird sie von `semtalk.DistributionIndex` angelegt (`CreateIndexFile()`-Methode). Abhängig davon, ob die XML-Datei des Modells vorhanden ist (Aktivitäten des Modellierers, siehe oben), wird ein Objekt der `ReferencesManager`-Klasse instantiiert. Ist dieses `ReferencesManager`-Objekt terminiert, wird die `Distribution-Index-Datei` gesichert (`SaveIndexFile()`). Das `DistributionIndex`-Objekt hat seine Aufgaben erfüllt und wird nicht mehr benötigt.

Die Eigenschaft `iCultureId` wird für die Anpassung der Mitteilungs- und der Aufgabentexte an die eingestellte Sprache des gemeinsamen Arbeitsbereich benötigt. Die Methode `XmlDocumentFromFile()` dient der Erzeugung von DOM-Objekten der XML-Dateien (`Distribution-Index` und der `Modell-Dateien`).²²⁶

²²⁶Das Objekt des Document Object Model (DOM) stellt den Inhalt einer XML-Datei im Hauptspeicher des Rechners bereit und erlaubt dessen Manipulation.

semtalk::DistributionIndex
+iCultureId : int = 1033
+UpdateIndex() +SaveIndexFile() -CreateIndexFile() -XMLOfFile() : XMLOfFile

Abbildung 33. Klasse
semtalk.DistributionIndex

(3) ReferencesManager – die Arbeiter-Klasse. Diese Klasse übernimmt die komplette Untersuchung der im betrachteten Modell bestehenden externen Referenzen. Des Weiteren analysiert sie alle Einträge im Distribution-Index, die im Zusammenhang mit dem aktuellen Modell stehen. Sie ermittelt (i) alle im aktuellen Modell bestehenden externen Referenzen (`FindReferencesInModel()`), (ii) löscht alle Einträge im Distribution-Index, die ihren Ursprung im betrachteten Modell haben (`ClearReferencesInIndex()`), (iii) fügt alle im betrachteten Modell gefunden externen Referenzen dem Distribution-Index hinzu (`AddReference()`), (iv) sucht im aktuellen Modell nach Referenzobjekten (`LookupReferenceObjects()`) und (v) markiert ggf die daraufhin ungültigen Einträge im Distribution-Index (`MarkReferenceEntryInIndex()`). Sind bei einer dieser Tätigkeiten neue oder veränderte Sachverhalte festgestellt worden,²²⁷ wird der Eigentümer des betreffenden Modells informiert (`PrepareNotification()`).

²²⁷ Mögliche Fälle siehe 3.2.3(2)

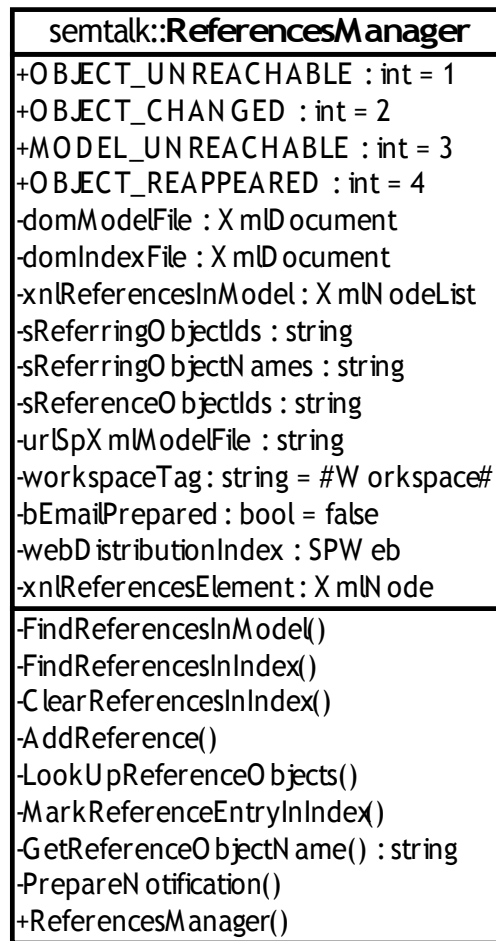


Abbildung 34. Klasse semtalk.ReferencesManager

(4) **Benachrichtigungen mit der Email-Klasse.** Unter den Voraussetzungen, dass für den Eigentümer eines Modells in dessen WSS-Profil eine Email-Adresse eingetragen ist und auf dem Windows Server 2003 die Email-Server-Funktion wurde, können Benachrichtigungen in Form von Emails an die Modell-Eigentümer versendet werden. Dieser Vorgang wird immer dann ausgelöst, wenn ein ReferencesManager-Objekt Inkonsistenzen bei der Aktualisierung des Distribution-Indexes festgestellt hat.

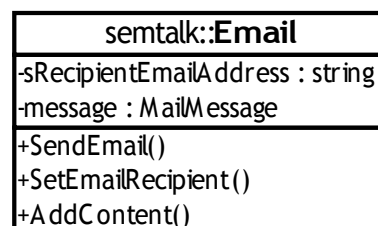


Abbildung 35. Klasse semtalk.Email

Der Inhalt der Email wird abhängig von der Art der Inkonsistenz erstellt (`AddContent()`) und an den Eigentümer des Modells verschickt (`SendEmail()`). Der Empfänger (`SetEmailRecipient()`) kann der Eigentümer des gerade betrachteten Modells oder die Eigentümer der Modelle sein, die sich auf das betrachtete Modell beziehen.²²⁸

(5) TaskList – Aufgaben für die Eigentümer. Da es – wie in (4) beschrieben – vorkommen kann, dass dem betreffenden Modell-Eigentümer wegen fehlender Voraussetzungen keine Email zugestellt werden kann, wird für ihn immer auch eine Aufgabe in der Aufgabenliste des gemeinsamen Arbeitsbereich erzeugt (`AddTaskListItem()`). Diese hat, neben der nicht immer möglichen Zustellung einer Email den Vorzug, direkt in dem *Freigegebenen Arbeitsbereich* von MS Visio angezeigt zu werden. Wie in 3.3.3 *Aktualisierung des Distribution-Indexes* beschrieben, kann der Eigentümer eines Modells sofort reagieren, sein Modell aktualisieren und die Aufgabe aus der Aufgabenliste löschen.



Abbildung 36. Klasse *semtalk.TaskList*

Lebenszyklen und Kommunikation der Komponenten

Für das Verständnis der folgenden Ausführungen zum Programmablauf der Aktualisierung des Distribution-Indexes, sind die im Anhang V zu findenden UML-Sequenzdiagramme hilfreich.

Im Gegensatz zu den statischen Konzepten der UML – zu denen das oben gezeigte Klassendiagramm gehört – dient das Sequenzdiagramm der Darstellung dynamischen Verhaltens der zu entwerfenden Software. Soll die konkrete Reihenfolge der Methodenaufrufe abgebildet werden, ist das Sequenzdiagramm als der primäre Diagrammtyp auszuwählen [Balzert 2005:328].

In Sequenzdiagrammen wird der Lebenszyklus von Objekten anhand einer vertikal verlaufenden Zeitachse verdeutlicht. Zunächst sind die an der Sequenz beteiligten Objekte in einer horizontalen Linie in Form von beschrifteten Quadraten aufzuzeichnen. Die Zeitachsen der Objekte²²⁹ verlaufen in vertikal abwärts gestrichelten Linien [Balzert 2005:81]. Auf ihnen erfolgt die Dokumentation der zwischen den Objekten stattfindenden Kommunikation in Abhängigkeit zur abgelaufenen Zeit. In Sequenzdia-

²²⁸ Auf die Behandlung der unterschiedlichen Fälle wird in den folgenden Teilen genauer eingegangen.

²²⁹ Die sogenannten „Lebenslinien“ [ebenda].

grammen ist es somit möglich, Parallelität während der Ausführung, synchrone und asynchrone Nachrichten²³⁰ zwischen den Objekten und ihre Lebensdauer abzubilden.

Die vorliegenden Sequenzdiagramme dokumentieren den konkreten Ablauf der Aktualisierung des Distribution-Indexes mit den Zeitachsen der beteiligten Objekte des `ModelLibEvtHndler`- und des `semtalk`-Namensraums.

(1) Lebenslinie der EventHandler-Klasse. Die `OnEvent()`-Methode der `EventHandler`-Klasse wird von den WSS ausgeführt, sobald ein Dokument der Dokumentbibliothek geändert wurde (siehe *Sequenzdiagramm 1*).

Die zwei Methoden (`GetAccountProperties()` und `CreateIdentity()`) werden, wie oben schon angedeutet, im Teil Umsetzung näher beschrieben. Sie haben für die Aktualisierung des Distribution-Indexes keine funktionale Bedeutung.

Innerhalb der `OnEvent()`-Methode werden die Ereignisse untersucht: Wurde ein Modell neu in die Dokumentbibliothek gespeichert oder – wenn es bereits in der Dokumentbibliothek existierte – lediglich wiederholt gespeichert, tritt die erste der zwei verfügbaren Alternativen (*alt*) ein. Dabei wird die `CallDistributionIndexService()`-Methode ausgeführt, die ein neues Objekt der `DistributionIndex`-Klasse (`semtalk`-Webdienst) erzeugt und deren öffentliche `Update()`-Methode aufruft. Der Pfad zu der XML-Datei des gerade gespeicherten Modells ist als wesentlicher Parameter der bei dem Aufruf der `Update()`-Methode zu übergeben. Das `EventHandler`-Objekt wird nun terminiert, wohingegen der Programmablauf der `DistributionIndex`-Objekts in dem Sequenzdiagramm 2 gekapselt verfolgt wird.

Hat der Modellierer hingegen ein Modell, das sich in der Dokumentbibliothek befand, umbenannt, verschoben oder gar gelöscht, wird in der `OnEvent()`-Methode die zweite Alternative ausgeführt. Dabei wird vor der Instantiierung des `semtalk`-Webdienstes die zu dem betreffenden Modell korrespondierende XML-Datei aus der Dokumentbibliothek gelöscht. Ungeachtet der Tatsache, dass das Modell in der XML-Datei nun nicht mehr existiert, wird das Objekt der `DistributionIndex`-Klasse erzeugt und dessen Methode `Update()` ausgeführt.

(2) Das DistributionIndex-Objekt. Wie soeben angeführt, wird das zum `semtalk`-Webdienst gehörende Objekt der `DistributionIndex`-Klasse in jedem Fall erzeugt. Entsprechend der im Sequenzdiagramm 2 abgebildeten Lebenslinie des `DistributionIndex`-Objekts, wird in dem ersten Schritt die Existenz der `Distribution-Index-Datei` ermittelt. Kann sie nicht gefunden werden, wird sie in der `CreateIndexFile()`-Methode angelegt. Anschließend wird die `Update()`-Methode erneut aufgerufen. Da nun die `Distribution-Index-Datei` existiert, wird sie in den Arbeitsspeicher geladen (`XmlDocu-`

²³⁰ Aufruf von Operationen des Objekts.

mentOfFile()).²³¹ Daran wird festgestellt, ob die XML-Datei des Modells vorhanden ist. Ist die Datei vorhanden,²³² wird sie in ein DOM-Objekt geladen und das ReferencesManager-Objekt erzeugt. Dabei wird der standardmäßige Konstruktor der Klasse ausgeführt. Anderenfalls²³³ wird das ReferencesManager-Objekt mit einem anderen Konstruktor erzeugt.

Ist das ReferencesManager-Objekt terminiert, wird die Distribution-Index-Datei gespeichert. Der Webdienst hat damit seine Arbeit getan.

(3) ReferencesManager-Objekt bei existierendem Modell. In dieser meistens auftretenden Situation sind bei der Aktualisierung des Distribution-Indexes alle Fälle zu berücksichtigen, die unter 3.2.3(2) aufgeführt sind. Zuerst werden, wie im Sequenzdiagramm 3 nachvollzogen werden kann, alle im betrachteten Modell bestehenden externen Referenzen aufgenommen (Methode FindReferencesInModel()). Von der sich daran anschließenden FindReferencesInIndex()-Methode gehen alle folgenden Aktivitäten aus: Erst werden sämtliche den externen Referenzen zuzuordnenden Einträge des Distribution-Indexes, die ihren Ursprung im aktuellen Modell haben, gelöscht (ClearReferencesInIndex()). Ist einer der zu löschenden Einträge im Distribution-Index als ungültig markiert,²³⁴ wird die betreffende externe Referenz in der Menge der aus dem soeben gespeicherten Modell ermittelten externen Referenzen gesucht. Kann sie gefunden werden, wird weder der betreffende Eintrag im Distribution-Index gelöscht, noch die ungültige externe Referenz neu in den Distribution-Index aufgenommen.²³⁵ Stattdessen erhält der Eigentümer eine Benachrichtigung (Sequenzdiagramm 5).

Ist der Distribution-Index um die ehemaligen Einträge bereinigt, wird versucht, für jede externe Referenz aus der Menge der im aktuell gespeicherten Modell ermittelten externen Referenzen ein Eintrag anzulegen (AddReference()). Gelingt es, das jeweilige referenzierte externe Modell zu finden, wird darin das Referenzobjekt gesucht und dessen Bezeichnung recherchiert (GetReferenceObjectName()-Methode). Konnte das referenzierte Modell nicht gefunden werden, erhält der Eigentümer des gerade gespeicherten Modells genauso eine Benachrichtigung (Sequenzdiagramm 5), wie für den Fall, dass das Referenzobjekt nicht gefunden werden konnte.

Sind alle gültigen externen Referenzen des Modells in dem Distribution-Index aufgenommen, müssen alle Einträge untersucht werden, in denen das betrachtete Modell von Objekten anderer Modelle referenziert wird (LookUpReferenceObjects()).²³⁶ Im

²³¹ Es wird ein DOM-Objekt erzeugt. Das heißt, der Inhalt des XML-Dokuments liegt nun in dem DOM-Objekt im Speicher des Windows Server vor.

²³² Modell neu oder wiederholt in die Dokumentbibliothek gespeichert.

²³³ Modell wurde umbenannt, verschoben oder aus der Dokumentbibliothek gelöscht.

²³⁴ status-Attribut des Reference-Elements (siehe 3.3.2(2)).

²³⁵ Der Modellierer hat ein Modell gespeichert, dessen externe Referenzen nicht alle gültig sind obwohl ihm dieser Sachverhalt schon bekannt ist.

²³⁶ Name der Modell-Datei ist im Destination-Element eines Eintrages als Wert des path-Attributes angegeben (siehe 3.3.2(2)).

Normalfall wird der Eintrag im Distribution-Index gültig sein und das Referenzobjekt kann im betrachteten Modell gefunden werden.²³⁷ Stimmt im Eintrag des Distribution-Indexes angegebene Bezeichnung des Referenzobjekts jedoch nicht mit der Bezeichnung des Objekts überein, die bei der Suche im betrachteten Modell ermittelt wurde, ist über diese Änderung der Eigentümer des referenzierenden Objekts zu informieren (Sequenzdiagramm 5). Er hat sein Modell zu aktualisieren.

Ist ein sich auf das aktuelle Modell beziehender Eintrag zwar als gültig markiert, aber das referenzierte Objekt kann in dem betrachteten Modell nicht gefunden werden, wird erst der Eintrag im Distribution-Index als ungültig markiert (`MarkReferenceEntryInIndex()`-Methode) und anschließend erfolgt die Benachrichtigung des Eigentümers des referenzierenden Objekts (Sequenzdiagramm 5).²³⁸

Bezieht sich eine externe Referenz in einem als ungültig ausgewiesenen Eintrag auf das aktuelle Modell, besteht die Möglichkeit, dass das vorher nicht gefundene Referenzobjekt neu erzeugt wurde.²³⁹ Das heißt, es wird nicht nach der Objekt-ID sondern nach der Bezeichnung des Referenzobjekts im betrachteten Modell gesucht. Konnte ein Objekt mit der gleichen Bezeichnung wie das ehemals vorhandene Referenzobjekt gefunden werden, ist es sehr wahrscheinlich, dass der Eigentümer des referenzierenden Objekts an diesem Fakt interessiert ist. Er erhält also eine Nachricht über den Fund und es wird eine Aufgabe für ihn erzeugt (Sequenzdiagramm 5). Entspricht das gefundene Objekt dem ehemaligen Referenzobjekt, kann er seine externe Referenz erneuern. Anderenfalls hat er sie zu löschen.

(4) ReferencesManager-Objekt bei nicht existierendem Modell. Wie schon angesprochen, kann ein Modell von dem Modellierer aus der Dokumentbibliothek entfernt worden sein. Denkbar ist auch die Umbenennung oder die Verschiebung. In jedem dieser möglichen Fälle wurde die zu der Modell-Datei korrespondierende XML-Datei von dem `EventHandler`-Objekt (siehe 3.3.3(1)) bereits gelöscht. Es müssen also – wie im Sequenzdiagramm 4 abgebildet – lediglich alle Einträge in dem Distribution-Index entfernt werden, in denen das gerade gelöschte Modell der Ursprung einer externen Referenz war (Überladung der `ClearReferencesInIndex()`-Methode). Anschließend werden alle Einträge im Distribution-Index als ungültig markiert, in denen das gelöschte Modell als Referenzmodell angegeben war (überladene `MarkReferenceEntryInIndex()`-Methode). Die Eigentümer der Modelle mit den nun ungültigen externen Referenzen sind über die Ungültigkeit der externen Referenzen zu informieren (Sequenzdiagramm 5).

²³⁷ Gesucht wird über die Objekt-ID.

²³⁸ Die Markierung eines Eintrags wird als Vorsichtsmaßnahme gegen wiederholtes Speichern von ungültigen externen Referenzen vorgenommen.

²³⁹ Jemand stellt fest, dass seine externe Referenz auf Grund der Löschung des Referenzobjekt ungültig geworden ist und beantragt bei dem Eigentümer des Referenzmodells die Wiederherstellung des betreffenden Objekts.

(5) Benachrichtigung und Erzeugung einer Aufgabe. In den schon vorgestellten Diagrammen wird oft das Sequenzdiagramm 5 eingebunden. In diesem ist dargestellt, wie die Benachrichtigungen an den Eigentümer eines betroffenen Modells realisiert werden. Das `ReferencesManager`-Objekt erzeugt jeweils ein neues Objekt der `Email`-Klasse: Die `PrepareNotification()`-Methode fügt den Inhalt der Email abhängig von der Art der Ungültigkeit ein (`AddContent()`) und es wird die Email-Adresse des Modell-Eigentümers als Empfänger eingetragen (`SetEmailRecipient()`). Zudem wird ein Objekt der `TaskList`-Klasse erzeugt. Das `TaskList`-Objekt repräsentiert die Aufgabenliste des gemeinsamen Arbeitsbereichs. Ihm wird nun eine neue Aufgabe hinzugefügt (`AddTaskListItem()`). Diese Aufgabe wird dem Eigentümer des betroffenen Modells zugeordnet. Das Versenden der Email (`SendEmail()`) schließt den Vorgang der Benachrichtigung ab.

4 Umsetzung der Komponenten

Die für die Plattform entworfenen Komponenten wurden auf einem von der Semtation GmbH, Potsdam bereitgestellten Windows Server umgesetzt. Als Betriebssystem stand der MS Windows Server 2003 Enterprise Edition mit Service Pack 1 zur Verfügung. Die hergestellte Testumgebung setzte sich aus den Microsoft Windows SharePoint Services 2003 und einem MS SQL Server 2000 Service Pack 4 als Datenbanksystem zusammen. Als Entwicklungswerkzeug kam MS Visual Studio 2003 zum Einsatz.

Die in 3.3.1(1) vorgestellte *Common Intermediate Language* (CIL) des .NET-Frameworks erlaubt es, eine vom Entwickler präferierte Hochsprache für die Programmierung zu verwenden. Für die Umsetzung der hier entworfenen Komponenten wurde das objektorientierte C#.net (Csharp) ausgewählt. Diese Wahl bot sich an, da zwischen C# und Java eine syntaktische Nähe besteht und der Autor somit bereits vorhandene Java-Kenntnisse einsetzen konnte. Diese Entscheidung wurde dadurch gestützt, dass viele im Internet verfügbar gemachten Code-Beispiele für SharePoint-Lösungen in C# verfasst sind.

Nachfolgend wird auf die wesentlichen Erfahrungen eingegangen, die sich während der Umsetzung des Entwurfs ergaben. Das Vorgehen während der Umsetzung kann als evolutionär und teilweise erforschend²⁴⁰ beschrieben werden. Dabei entstand immer ein wiederverwendbarer Prototyp.

4.1 Codezugriffssicherheit im .NET-Framework

Jede programmatische Arbeit mit Microsofts .NET-Plattform erfordert, dass der auszuführende Programmcode über ausreichend Rechte für die ihm übertragenen Aufgaben besitzt. Dafür hat MS die Codezugriffssicherheit (*Code Access Security* - CAS) als integralen Bestandteil in das .NET-Framework eingebaut. Obwohl das Thema CAS weit über den hier verfügbaren Rahmen hinausgeht, müssen seine Grundlagen verinnerlicht werden. Denn es nicht möglich, Anwendungen auf der .NET-Plattform zu entwickeln und gleichzeitig Maßnahmen zur Einhaltung der CAS außer Acht zu lassen.

Wohingegen die aus MS Windows bekannte Sicherheit auf der Benutzerebene der zentralen Fragestellung folgt: „Wer ist der Benutzer und was kann der Benutzer tun?“, lautet sie in Hinblick auf den ausführbaren .NET-Programmcode „Woher stammt der Code, wer hat den Code geschrieben, und was kann der Code tun“.²⁴¹ Die CAS „autorisiert Code beim Versuch, auf geschützte Ressourcen wie Dateisystem, Registrierungsda-

²⁴⁰ Siehe exploratives Prototyping in [Stahlknecht; Hasenkamp 2005:219]

²⁴¹ „.NET-Sicherheit – Überblick.“ <<http://www.microsoft.com/germany/msdn/library/security/ErhoehenDerSicherheitVonWebanwendungen/secmod79.aspx>> (27.01.2006)

tei und Netzwerk zuzugreifen, oder beim Versuch, andere privilegierte Vorgänge auszuführen[...]“ [ebenda]. Daraus folgt, dass die Rechte eines an Windows angemeldeten Benutzers nicht per se das aufgerufene Programm zur Ausführung von dessen Funktionen autorisieren. Der Programmcode *selbst* muss über die dazu nötigen Rechte verfügen. Um die Rechte des Programmcodes festzustellen, werden beim Laden des Codes standort-²⁴² und autorbezogene²⁴³ Beweise gesammelt, die dann verwendet werden, um dem Code die Codezugriffsrechte zu erteilen.²⁴⁴ Das alles übernimmt das Windows-Betriebssystem, wenn verwalteter Code ausgeführt werden soll.

4.2 Anpassung von SemTalk

Die Ausführungen in 3.2.4 machen deutlich, dass an SemTalk einige Anpassungen vorgenommen werden mussten, damit ein Zusammenspiel mit den WSS überhaupt möglich wird.

(1) SemTalk als Add-In für MS Visio. Erst mit der Einbettung von SemTalk in die MS-Office-Visio-Anwendung können nun die Modellierer von dem erweiterten Funktionsumfang der Microsoft Office System Integration profitieren. Bei der Öffnung eines SemTalk-Modells oder einer SemTalk-Modellvorlage durch MS Visio wird SemTalk von MS Visio nachgeladen und über einen zusätzlichen Menüpunkt verfügbar gemacht. Zusätzlich werden der *Modell-Explorer* und die für die jeweilige Modellierungsmethode vorgesehenen Modellelemente (Visio-Schablonen) geöffnet. In Abbildung 37 ist die Benutzungsoberfläche bestehend aus MS Visio mit SemTalk als Add-In vollständig geladen. Es sind alle Vorteile der Microsoft Office System Integration (Freigegebener Arbeitsbereich) und von SemTalk nutzbar – gültige Modelle können erzeugt und bearbeitet werden. Alle wesentlichen Benutzungsoberflächen (Freigegebener Arbeitsbereich, Modell-Explorer und Schablonen der Modellierungsmethode) lassen sich nach der persönlichen Vorliebe um die Modellierungsoberfläche anordnen.

Der Anwender hat, will er die WSS als Plattform für die verteilte Modellierung mit SemTalk nutzen, neben der genuinen SemTalk-Anwendung das SemTalk-Add-In für Visio zu installieren.

²⁴²Zentrale Fragen hier: Von wo aus wurde der Programmcode aufgerufen, in welchem Verzeichnis wird er ausgeführt.

²⁴³Wer ist der Herausgeber des Codes; was ist sein *strong name* (wird bei der Signierung der Assemblierung mit einem Öffentlichen Schlüssel angelegt) etc.

²⁴⁴Microsoft. „Codezugriffssicherheit in der Praxis“.
<<http://www.microsoft.com/germany/msdn/library/security/ErhoehenDerSicherheitVonWebanwendungen/secmod81.msp>> (22.3.2006)

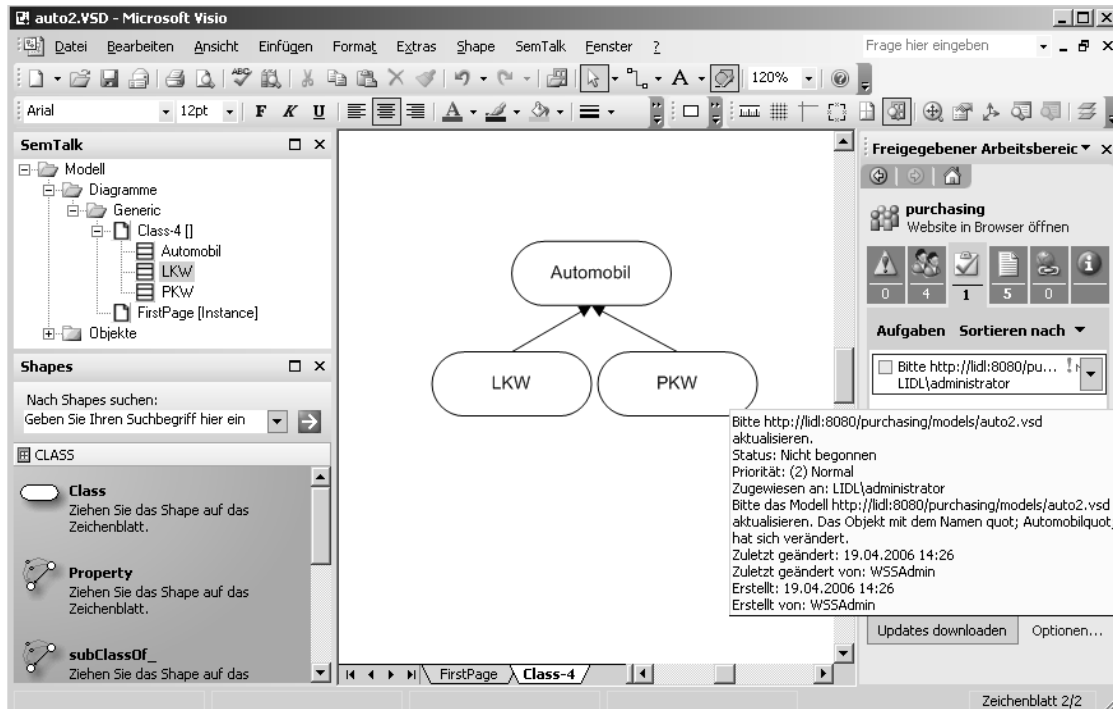


Abbildung 37. Ein Modell, geöffnet in MS Visio mit SemTalk-Add-In und Anzeige einer automatisch erzeugten Aufgabe.

(2) **Speichern der XML-Datei.** SemTalk hat, damit verteiltes Modellieren möglich wird, neben der eigentlichen Modell-Datei im Format von MS Visio eine XML-Repräsentation des Modells zu speichern (siehe 3.2.4). Die für das Öffnen und Speichern der Modell-Datei benötigten Benutzerrechte fragt MS Visio zu Anfang einmal ab und speichert sie für die aktuelle Sitzung. Wird die Modell-Datei in dem gemeinsamen Arbeitsbereich gespeichert und versucht der Anwender ein anderes Modell zu öffnen, wird zunächst geprüft, ob er über die entsprechenden Rechte verfügt. Da die Sitzung zwischen MS Visio als Clientanwendung und den WSS zwischenzeitlich nicht beendet war, erfolgt das Öffnen der Datei ohne neue Abfrage der Benutzerdaten.

Das gleichzeitige Speichern der XML-Datei in der selben Dokumentbibliothek durch SemTalk war zunächst nicht möglich. Die WSS verweigerten diesen Schreibvorgang auf Grund fehlender Rechte. Der für diesen Zweck in SemTalk implementierte Programmcode verfügte nicht über die nötigen Rechte (siehe 4.1). Die Benutzerrechte der Sitzung zwischen MS Visio und den WSS werden auf dem zugreifenden PC gekapselt gespeichert.²⁴⁵ Es gelang vorerst nicht, diese gespeicherten Benutzerrechte für die parallele Speicherung des XML-Datei zu nutzen. Die zu diesem Thema verfügbare Online-Dokumentation bot keine Hilfestellung. Aus diesem Grund war es unumgänglich, in SemTalk einen Dialog einzubauen, der die für den gemeinsamen Arbeitsbereich gültigen Benutzerdaten sichert.

²⁴⁵ MS Office sichert die Rechte einer Client-Server-Sitzung in einem *Cache*.

Mit diesen Informationen konnte nun der Programmcode von SemTalk für den Vorgang der Speicherung der XML-Datei *personalisiert* werden.

Das Speichern der zu den Modellen korrespondierenden XML-Dateien erfolgt in einem in der Dokumentbibliothek anzulegenden Ordner *XMLFiles*. Das hat den Vorteil, dass die Sicht auf die Dokumentbibliothek so angepasst werden kann, dass dieser Ordner (samt Inhalt) nicht auf der Webseite angezeigt wird. Somit bleibt das Vorhandensein der XML-Dateien vor dem Modellierer verborgen.

(3) Externe Modelle öffnen. Nachdem es möglich war sowohl die Modelle im MS Visio- und im XML-Format in dem gemeinsamen Arbeitsbereich zu öffnen und zu speichern, galt es die von SemTalk verwendeten Mechanismen bei der verteilten Modellierung für die WSS anzupassen. Die URL zu einem Modell in einem gemeinsamen Arbeitsbereich unterscheidet sich nicht von einer URL, die auf ein Modell zeigt, das auf einem herkömmlichen Webserver gespeichert ist. Jedoch zeigen sich bei dem Versuch, das Modell zu öffnen, signifikante Unterschiede. Soll SemTalk eine externe Referenz oder eine externe Verfeinerung zu einem externen Modell anlegen, so liest SemTalk die XML-Datei des externen Modells im Hintergrund ein (siehe 2.1.1). Das ist in Bezug auf die WSS aber nur möglich, wenn der hierfür entwickelte Programmcode die nötigen Rechte besitzt. Das bedeutet programmiertechnisch, dass SemTalk ein anderes Verfahren bei der Angabe eines Modells aus einem gemeinsamen Arbeitsbereich anwenden muss, als es für herkömmliche Webserver üblich war. Daraus entstand die Idee, sich alle in einer Dokumentbibliothek vorhandenen SemTalk-Modelle aufzulisten (Menüpunkt *Team Workspace*).

Der *Team Workspace* zeigt nur Modelle an, wenn (i) MS Visio den gemeinsamen Arbeitsbereich erkannt hat, (ii) die in SemTalk hinterlegten Benutzerrechte für die Dokumentbibliothek ausreichen und (iii) wenn in der Dokumentbibliothek außer dem gerade geöffneten, noch weitere Modelle vorhanden sind. Er wird immer dann angezeigt, wenn ein sich ebenfalls in der Dokumentbibliothek befindliches externes Modell als Referenzmodell oder als Ziel einer externen Verfeinerung dienen soll. Des Weiteren nutzt die Import-Funktion des SemTalk-Add-Ins die *Team Workspace*-Funktion.

(4) Örtlich unabhängiges Modellieren. In einigen Versuchen zeigte sich, dass es für die Arbeit an Modellen in einem gemeinsamen Arbeitsbereich von unterschiedlichen Orten aus erforderlich war, die bisherige Vorgehensweise – in den Eigenschaften eines referenzierenden Objektes die komplette URL zu dem Referenzmodell zu speichern – nicht mehr funktionierte. Vielmehr muss die URL zu den externen Modellen als relativ betrachtet werden, da sich der Standort eines Projektbeteiligten immer ändert. Die prototypische Umgebung in der Semtation GmbH verdeutlicht die Problematik: Der Windows Server, der die WSS bereitstellt, befindet sich im lokalen Netzwerk und besitzt eine private IP-Adresse sowie einen Rechnernamen. Über beide ist er im lokalen Netzwerk unter einer

bestimmten Anschlussnummer erreichbar (`http://semtation:8080`). Eine externe Referenz, die von einem Modell auf ein anderes Modell in diesem gemeinsamen Arbeitsbereich verweist, hat als Pfadangabe zu dem Referenzmodell die folgende, beispielhafte URL: `<http://semtation:8080/modelle/XMLFiles/Garten.xml>`.

Mit der Öffnung der Anschlussnummer am *Internet-Gateway*²⁴⁶ und der Weiterleitung von Anfragen aus dem Internet an diese Anschlussnummer an die private IP-Adresse im lokalen Netzwerk, ist der Windows Server und alle unter der Anschlussnummer eingerichteten gemeinsamen Arbeitsbereiche vom Internet aus erreichbar. Folglich kann standortunabhängig, sofern keine technischen Unterbrechungen auftreten, auf den gemeinsamen Arbeitsbereich und die darin befindlichen Modelle zugegriffen werden.

Hervorzuheben ist, dass der Windows Server in der prototypischen Umgebung im Internet unter einer anderen Internetadresse erreichbar ist, als im lokalen Netzwerk (hier: `http://neuegarten.mine.nu:8080`).²⁴⁷ Befand sich nun ein Modellierer zum Zeitpunkt der Erzeugung einer externen Referenz im lokalen Netzwerk und öffnete später das gleiche Modell vom Internet aus, war das Referenzobjekt nicht erreichbar, denn die in der Pfadangabe der externen Referenz festgelegte URL (`http://semtation:8080`) existiert im Internet nicht. Diese Problematik konnte gelöst werden, indem eine externe Referenz, die zwischen Modellen in einer Dokumentbibliothek erzeugt wurde, auf die vollständige URL verzichtet und lediglich den Namen des Referenzmodells speichert. Diese Maßnahme bringt einige Einschränkungen mit sich, weshalb sie nicht als optimal bezeichnet werden kann. So können externe Referenzen und externe Verfeinerungen nur zwischen Modellen in einer Dokumentbibliothek erzeugt werden.

4.3 Integration der entwickelten Komponenten

Des öfteren wurde in den vorangegangenen Teilen (siehe 3.3.1) davon gesprochen, dass eine Ereignisroutine an eine Dokumentbibliothek gebunden werden muss. Die Ereignisroutine ist erst dann durch die WSS ausführbar, wenn sie aus einer vertrauenswürdigen Quelle stammt. Da sie in einer Programmiersprache des .NET-Framework entwickelt ist, handelt es sich um eine Assemblierung verwalteten Codes (*managed code assembly*). Damit die Assemblierung als vertrauenswürdiger Code anerkannt und ausgeführt werden kann, ist sie zu signieren²⁴⁸ und anschließend in den GAC des Windows Server zu installieren.²⁴⁹

²⁴⁶ Meist eine Hardware-Komponente mit Firewall-Funktionalität, mit der die Verbindung des lokalen Netzes zum Internet hergestellt wird.

²⁴⁷ Dies konnte mit Hilfe eines DynDNS-Eintrags erreicht werden.

²⁴⁸ Hierzu muss ein öffentlicher Schlüssel erzeugt und der Assemblierung zum Zeitpunkt der Compilierung des Programmcodes als Parameter mitgeteilt werden. Sie erhält dann auch einen *strong name*.

²⁴⁹ Verzeichnis *assembly* im Windows-Programmverzeichnis.

Anschließend, sofern noch nicht geschehen, müssen auf der Verwaltungsebene des virtuellen Servers Ereignisroutinen gestattet werden.

Liegt die Ereignisroutine als signierte Assemblierung im GAC und sind Ereignisroutinen für den virtuellen Server erlaubt, kann nach dem Neustart des IIS die Ereignisroutine an jede Dokumentbibliothek aller gemeinsamen Arbeitsbereiche in dem virtuellen Server gebunden werden. Das geschieht in den erweiterten Einstellungen einer Dokumentbibliothek, in denen die Angaben zu der anzubindenden Ereignisroutine eingetragen werden müssen.

Erweiterte Einstellungen der Dokumentbibliothek: models

Mit Hilfe dieser Seite können Sie die erweiterten Einstellungen dieser Dokumentbibliothek ändern. Sie können die E-Mail-Einstellungen und den Ereignishandler für die Bibliothek ändern.

<p>E-Mail-Einstellungen</p> <p>Geben Sie die Adresse des öffentlichen Ordners für diese Dokumentbibliothek an. Der SharePoint-Timerdienst wird diesen öffentlichen Ordner regelmäßig lesen und die Anlagen aus neuen Nachrichten im öffentlichen Ordner in der Dokumentbibliothek einfügen.</p>	<p>Adresse des öffentlichen Ordners: <input type="text"/></p>
<p>Ereignishandler</p> <p>Geben Sie den Assemblierungsnamen, den Klassennamen und die Eigenschaften des Ereignishandlers der Dokumentbibliothek ein. Windows SharePoint Services wird diesen Ereignishandler aufrufen, wenn Einträge in der Dokumentbibliothek eingefügt, aktualisiert oder gelöscht werden.</p>	<p>Assemblierungsname: <input type="text" value="ModellLibEvtHndler, Version=1.0.1.0, Culture=N"/></p> <p>Klassenname: <input type="text" value="ModellLibEvtHndler.EventHandler"/></p> <p>Eigenschaften: <input type="text"/></p>

Abbildung 38. EventHandler-Klasse, gebunden an eine Dokumentbibliothek.

Die Abbildung 38 zeigt, dass im Feld *Assemblierungsname* alle Informationen aus den Eigenschaften der Assemblierung eingegeben werden müssen. Daran schließt sich die Eingabe des kompletten Klassennamens an, der aus dem Namensraum und dem Namen der Klasse besteht, welche die `OnEvent()`-Methode implementiert. Das dritte, hier noch leerstehende Feld *Eigenschaften* dient der Übergabe eines maximal 255 Zeichen langen Parameters an die Ereignisroutine. Dieser kann im Programmcode der Ereignisroutine in der `OnEvent()`-Methode abgefragt werden.

Im Gegensatz zur Ereignisroutine ist die Entwicklung und die Verwendung eines eigenen Webdienstes für die WSS eine größere Herausforderung. Für die Entwicklung des Webdienstes wird ein neuer virtueller Server benötigt, der nicht mit den WSS-Funktionen erweitert wurde. In diesem virtuellen Server ist unter Verwendung der Webdienst-Vorlage des MS Visual Studio 2003 der Webdienst zu entwickeln und zu kompilieren. Die so erstellte Programmbibliothek ist in das gleiche physikalische Verzeichnis zu kopieren, in dem die bereits vorhandenen Webdienste der WSS gespeichert sind.²⁵⁰

²⁵⁰ <Laufwerk>:\Programme\Gemeinsame Dateien\Microsoft

Um den Webdienst im Windows Server *bekannt* zu machen, ist eine WSDL- und eine DISCO-Datei zu erzeugen, die Informationen über den Webdienst beinhalten (siehe *Webdienste* in 3.3.1). Diese sind in das dafür vorgesehene physikalische Verzeichnis abzuliegen.²⁵¹ Hinzu kommt schließlich eine ASMX-Datei, mit deren Hilfe der IIS den Webdienst im Internet veröffentlicht. Das Verzeichnis der WSDL- und DISCO-Dateien wird von den WSS *virtualisiert* und ist so in jedem gemeinsamen Arbeitsbereich aller mit der WSS-Anwendung erweiterten virtuellen Server erreichbar.²⁵²

Steht der Webdienst als Ressource zur Verfügung, kann er in die Assemblierung der Ereignisroutine eingebunden werden. Diese instantiiert ihn zur Laufzeit und ruft zusammen mit einer Parameterliste die öffentliche Methode (Webmethode) des Webdienstes auf (hier: `DistributionIndex.Update()`).

4.4 Der Webdienst und die Codezugriffssicherheit

Dem .NET-Programmcode kann auf unterschiedliche Weise der nötige Umfang an Rechten (siehe 4.1) übertragen werden. Für den semtalk-Webdienst bestand die Herausforderung darin, dass er für alle existierenden (und noch nicht existierenden) gemeinsamen Arbeitsbereiche, auf alle darin befindlichen Dokumente und Listen uneingeschränkten Zugriff benötigte. Der Webdienst musste demzufolge über Administrator-Rechte verfügen. Da der semtalk-Webdienst von dem `ModelLibEvtHndler.EventHandler` instantiiert und mittels der Webmethode `Update()` aufgerufen wird, hat

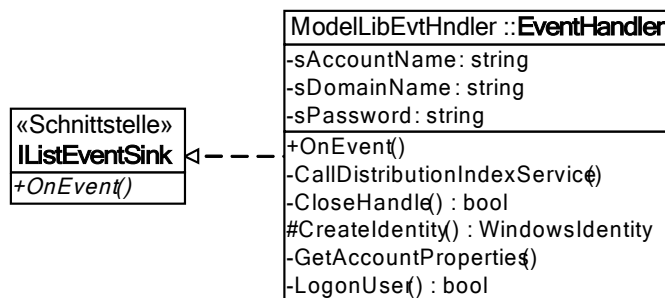


Abbildung 39. Klasse `ModelLibEvtHndler.EventHandler`

dieser Aufruf mit der *Identität* eines Administrators zu erfolgen. Zudem musste die Ereignisroutine nach Löschung, Umbenennung und Verschiebung der Modell-Datei durch den Modellierer die korrespondierende XML-Datei löschen. Auch dieser Vorgang benötigt administrative Zugriffsrechte in der gesamten WSS-Umgebung. Um den Programmcode

Shared\web server extensions\60\ISAPI\BIN

²⁵¹<Laufwerk>:\Programme\Gemeinsame Dateien\Microsoft

Shared\web server extensions\60\ISAPI\

²⁵²`http://<Servername>:<Anschlussnummer>/<GemeinsamerArbeitsbereich>/_vti_bin/DistributionIndex.asmx`

mit Administrator-Rechten auszustatten, muss die Personifizierung des Codes erfolgen. Dazu wird noch einmal die Klasse `ModelLibEvtHndler.EventHandler` in Abbildung 39 dargestellt.

Für die Personifizierung des Programmcodes wird ein `WindowsIdentity`-Objekt erzeugt (`CreateIdentity()`), das die Identität eines Benutzers mit Administrator-Rechten verkörpert. Dazu müssen der `CreateIdentity()`-Methode der Benutzername, das Passwort und die Anmelde-Domäne des Benutzerkontos mitgeteilt werden. Dieses Identitäts-Objekt hat sich am Windows-Betriebssystem anzumelden (`LogonUser()`). Die Anmelde-Funktion wird von Windows-Programmbibliotheken bereitgestellt. Der Programmcode, der zwischen der Anmeldung und der Abmeldung des Identitäts-Objekt ausgeführt wird, verfügt nun über die Rechte des personifizierten Benutzers.

Unter Berücksichtigung der in Zukunft möglichen Anwendungsfälle des `semtalk`-Webdienstes, musste bereits zu diesem Zeitpunkt eine Parametrisierung implementiert werden. Diese hat zu gewährleisten, dass der Administrator eines Windows Server 2003 – will er die neuen Komponenten nutzen – den für die Personifizierung zu verwendenden Benutzer mit Administrator-Rechten selbst auswählen kann. Die hierfür ausgearbeitete Lösung sieht vor, dass die Ereignisroutine nach ihrem automatischen Aufruf auf einen Schlüssel in der Windows-Registrierungsdatenbank zugreift, in dem die Benutzerdaten hinterlegt wurden. Für die Hinterlegung der Benutzerdaten entwickelte der Autor eine Windows-Anwendung²⁵³, die vor der Installation der Komponenten zur verteilten Modellierung auf dem Windows Server auszuführen ist. Diese listet zunächst alle Benutzer des Betriebssystems auf und erlaubt, einen der aufgeführten Benutzer auszuwählen. Nach Eingabe von dessen Passwort werden die ermittelten Daten – das Passwort mit dem RSA-Verschlüsselungsalgorithmus verschlüsselt – in einem allgemein zugänglichen Schlüssel der Windows-Registrierungsdatenbank ablegt.²⁵⁴ Der im .NET-Framework verfügbare Implementierung des RSA-Algorithmus erzeugt bei der Verschlüsselung ein Schlüsselpaar, bestehend aus einem privaten und einem öffentlichen Schlüssel. Ohne den öffentlichen Schlüssel ist die Ereignisroutine nicht in der Lage, das Passwort des Administrator-Benutzerkontos zu entschlüsseln und damit den eigenen Programmcode zu personifizieren. Darum muss die Ereignisroutine neben den lesenden Zugriffsrechten auf den Registrierungsschlüssel über den öffentlichen Schlüssel verfügen.

Die entgeltliche Erlangung der benötigten Rechte musste stufenweise realisiert werden. Obwohl die Ereignisroutine in dem GAC installiert war und damit als vertrauenswürdige Assemblierung galt, gelang es zunächst nicht, aus dem Programmcode der Ereignisroutine heraus auf den Registrierungsschlüssel lesend zuzugreifen.²⁵⁵ Das Benutzerkonto, in dem

²⁵³ Bezeichnung: *Account2Registry*.

²⁵⁴ `HKEY_LOCAL_MACHINE\SOFTWARE\SemTalk4WSS`

²⁵⁵ Jeder *menschliche* Benutzer konnte lesend und schreibend auf den Schlüssel zugreifen. Ein Beispiel dafür, wie restriktiv die CAS ist.

der Arbeitsprozess der WSS ausgeführt wird,²⁵⁶ erlaubt keinen direkten Zugriff auf die Registrierungsdatenbank des Windows-Betriebssystems. Demnach war eine Änderung der Anwendungspool-Identität nötig. Eine Identität auszuwählen die sofort alle Rechte in dem Windows-System besitzt, ist keine akzeptable Lösung. Das Betriebssystem sollte auch weiterhin vor einer unbefugten Benutzung mit möglicherweise komprimierten WSS-Programmcode geschützt bleiben. Also wurde dem Anwendungspool ein nur mit einfachen Rechten ausgestatteter Benutzer zugeordnet. Diesem Benutzer konnte aber nun das explizite Leserecht für den Registrierungsschlüssel in der Windows-Registrierungsdatenbank gestattet werden. Damit war es möglich, den Registrierungsschlüssel mit dem Programmcode der Ereignisroutine auszulesen.

Die Windows-Anwendung zur Ablage der Benutzerdaten in der Registrierungsdatenbank speichert den während des Vorgangs zur Verschlüsselung erzeugten öffentlichen Schlüssel in eine Datei im XML-Format. Das passiert, nachdem das Passwort verschlüsselt in den Registrierungsschlüssel geschrieben wurde. Damit der Administrator einen gewissen Schutz mit eigenen Maßnahmen umsetzen kann, könnte er den Namen der XML-Datei ändern und an einem anderen Ort auf der lokalen Festplatte ablegen.

Der Ereignisroutine kann, wie unter 4.3 erwähnt, ein 255 Zeichen langer Parameter mitgeteilt werden. Diese Mechanismus wird nun verwendet, um der Ereignisroutine den Pfad und den Dateinamen der erzeugten XML-Datei des Schlüssel-paares mitzuteilen (Abbildung 40). Die Methode `GetAccountProperties()` der `EventHandler`-Klasse liest nun bei jedem Ereignis die XML-Datei und anschließend den Registrierungsschlüssel aus. So ermittelt die Methode die für den Personifikationsprozess nötigen Benutzerdaten des Administrator-Kontos. Ist die Personifizierung des Programmcodes erfolgreich verlaufen, kann die zu dem betroffenen Modell korrespondierende XML-Datei gelöscht werden und der semtalk-Webdienst seine Aufgaben erfüllen. Sofort nach dem Aufruf des semtalk-Webdienstes wird die Personifizierung des Programmcodes der Ereignisroutine rückgängig gemacht.



Abbildung 40. Eigenschaften der Ereignisroutine `ModelLibEvtHandler.EventHandler`

²⁵⁶ Zuordnung erfolgt in den Eigenschaften des Anwendungspools (siehe 2.4.3). Standardmäßig ist das Prozesskonto des IIS ausgewählt, das wenige Rechte besitzt.

5 Abschließende Betrachtung

In diesem Abschnitt, der gleichzeitig den Schluß dieser Arbeit markiert, erfolgt die Bewertung der geschaffenen Lösung und der Ausblick. Im Mittelpunkt der Bewertung stehen die WSS, die um zusätzliche Funktionen erweitert wurden. Es soll festgestellt werden, inwieweit diese Lösung als Plattform für die verteilte Modellierung mit SemTalk geeignet ist. Im Ausblick wird ein Resümee gezogen und auf Entwicklungen und Arbeiten hingewiesen, die in der Zukunft nötig sind.

5.1 Bewertung der ausgearbeiteten Lösung

Die Bewertung einer Anwendung kann auf unterschiedliche Art und Weise erfolgen. So kann die Qualität beurteilt werden,²⁵⁷ die Wirtschaftlichkeit der Nutzung, die technische Umsetzbarkeit sowie die Effekte, die sich mit der Einführung der Software auf die Aufgabenerfüllung ergeben.²⁵⁸ Da hier die Eignung der Lösung als Plattform für die verteilte Modellierung beurteilt werden soll, konzentriert sich die folgende Bewertung auf die Effekte. Diese werden im Allgemeinen in quantifizierbare und ggf monetär bewertbare sowie in nicht quantifizierbare, dh qualitative Effekte unterteilt.²⁵⁹ Zur Ermittlung der quantifizierbaren Effekte ist der messbare Einsatz in einer Realsituation erforderlich. Da jener im Rahmen dieser Arbeit jedoch nicht durchgeführt werden konnte, liegt der Schwerpunkt der nun folgenden Bewertung auf den *qualitativen* Effekten, die sich aus der Einführung und Nutzung der WSS als Plattform für die verteilte Modellierung ergibt.

5.1.1 Strategische und wirtschaftliche Aspekte

Die Einführung und Nutzung einer Anwendung ist auch immer eine Investition, zu deren Entscheidung eine Bewertung nach wirtschaftlichen Gesichtspunkten zu treffen ist. Die Portfolioanalyse ist ein Werkzeug, die diese Entscheidung unterstützen kann. Mit ihr ist es möglich, die rein wirtschaftliche bzw die strategische Bedeutung des Anwendungssystems für die Organisation einzuschätzen und so eine (Investitions-)Entscheidung leichter zu fällen. Die bekannteste Variante der Portfolioanalyse stammt von der Boston Consulting Group (BCG) [Jung 2001:601].²⁶⁰ Mit den Jahren entdeckten immer mehr Gruppen die Vorteile der BCG-Matrix und entwickelten weitere Anwendungsfälle. So hat

²⁵⁷ Zum Beispiel die Qualität der Software mit Hilfe des ISO-Standards 9126 oder die Qualität der Benutzbarkeit anhand des ISO-Standards 9241 Teil 10.

²⁵⁸ Grundlegende Aussage aus [Stahlknecht; Hasenkamp 2005:446] sowie [Mertens ua 2004:189ff].

²⁵⁹ [Stahlknecht; Hasenkamp 2005:251], [Bearing Point 2003a:3]

²⁶⁰ Bruce D. Henderson. 1973. „The Experience Curve – Reviewed – IV. The Growth Share Matrix“. <http://www.bcg.com/publications/publication_view.jsp?pubID=720> (26.3.2006)

sich beispielsweise in der Betriebswirtschaft die vier-Felder-Matrix der Portfolioanalyse zur Darstellung des Produktlebenszyklusses durchgesetzt.²⁶¹

An den Spalten der Matrix wird die klassische Wirtschaftlichkeit und an den Zeilen die strategische Bedeutung des einzuführenden Anwendungssystems abgezeichnet [Stahlknecht; Hasenkamp 2005:254]. Während sich die klassische Wirtschaftlichkeit in betriebswirtschaftlichen Zahlen ausdrücken lässt (Aufwand vs Ertrag), bezieht sich die strategische Bedeutung auf qualitative Aspekte, die sich mit der Nutzung des Systems ergeben.

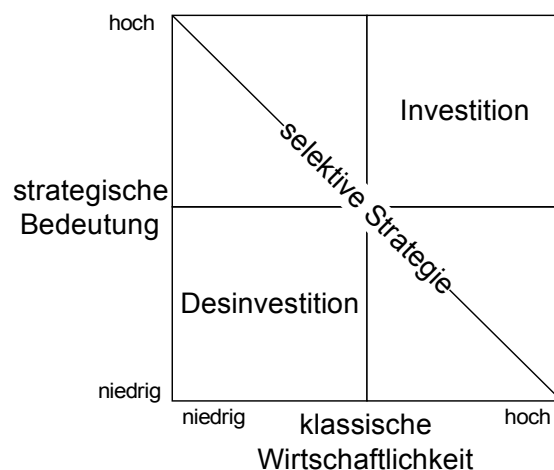


Abbildung 41. Klassische Portfolio-Analyse

Können mit der einzuführenden Anwendung Kosten gesenkt werden (hohe klassische Wirtschaftlichkeit) und bereichert das Vorhaben die IT-Landschaft des Unternehmens gleichzeitig derart, dass daraus eine verbesserte Wettbewerbsfähigkeit hervorgeht (hohe strategische Bedeutung), ist die Einführung unbedingt zu empfehlen. Ist beides nicht gegeben, sollte von einer Einführung abgesehen werden. Abzuwägende Grenzfälle ergeben sich dann, wenn ein System einen der beiden Aspekte voll zu erfüllen scheint, den anderen hingegen kaum. Dann ist genauer zu prüfen und selektiv zu verfahren, wie in Abbildung 41 gezeigt.

Da der Nutzen, der sich mit der Verwendung der WSS als Plattform zur verteilten Modellierung ergibt, hauptsächlich qualitativer Natur ist,²⁶² muss von einer hauptsächlich strategischen Bedeutung ausgegangen werden. Das heißt, die WSS sind in der linken oberen Ecke der Matrix anzusiedeln. Ihre Einführung und Nutzung ist also aus rein wirtschaftlichen Gesichtspunkten mit einem Fragezeichen zu versehen. Zu einem ähnlichen Ergebnis kommen [Mertens ua 2004:190]: Sie bewerten in ihrer Portfolio-Analyse jedoch nicht die WSS im Speziellen, sondern die wirtschaftliche Bedeutung von Workgroup-Support-Systemen im Allgemeinen.

²⁶¹ <<http://de.wikipedia.org/wiki/Produktlebenszyklus>> (26.3.2006)

²⁶² Verkürzte Dauer der Modellierungsprojekte oder gar Kostensenkungen können nicht nachgewiesen werden.

Können trotz alledem wesentliche qualitative Effekte festgestellt werden, kann die Einführung der WSS durchaus sinnvoll sein. Die Ermittlung der qualitativen Effekte erfolgt im nächsten Abschnitt.

5.1.2 Bewertung des Nutzens

Die qualitativen Effekte, die sich aus der Verwendung der erweiterten WSS als Plattform für die verteilte Modellierung ergeben, sind als Nutzen zu verstehen. Für die Bewertung des qualitativen Nutzens wird in der Literatur die Verwendung einer Multifaktorenmethode – der sogenannten Nutzwertanalyse – empfohlen.²⁶³ Die Durchführung der Nutzwertanalyse erfolgt in vier Schritten: Nach (i) Festlegung der Bewertungskriterien werden (ii) sie gewichtet, es wird (iii) der Grad der Veränderung des *IST-Zustands* anhand der Bewertungskriterien in der Nutzwerttabelle bestimmt und (iv) der Nutzenkoeffizient errechnet [Stahlknecht; Hasenkamp 2005:253]. Ist der Nutzenkoeffizient größer als 1, ist der zu erwartende Nutzen positiv und es ergeben sich aus der Verwendung der WSS als Plattform für die verteilte Modellierung Mehrwerte [ebenda]. Dieses Prinzip beruht auf der allgemeinen Bewertung der Wirtschaftlichkeit, bei welcher der Quotient aus Ertrag und Aufwand mindestens 1 sein muss [Jung 2001:29].

Der IST-Zustand, an der sich die Bewertung des Gefüges aus den erweiterten WSS und SemTalk zu orientieren hat, setzt sich wie folgt zusammen: Die alleinstehende SemTalk-Anwendung ist auf den PCs der Modellierer installiert und ein freigegebenes Verzeichnis im lokalen Netzwerk dient als Speicherort der beteiligten Modelle. Hier befindet sich auch der von allen genutzte Modell-Index. Die Modellierer haben sich entweder unter persönlicher Anwesenheit abzustimmen oder kommunizieren via Email, Instant Messenger oder Telefon miteinander.

(1) Bewertungskriterien. In [Bearing Point 2003b:5] wird der Katalog der Bewertungskriterien in vier große Kategorien eingeteilt. Dabei wird in anbieter- und anwendungsbezogene, technische sowie einführungs- und nutzungsbezogene Kriterien unterschieden. Diese Kategorisierung hilft dem Anwender, Anforderungen zu formulieren, so dass alle Spektren abgedeckt sind.

Die für diese Arbeit relevanten Bewertungskriterien umfassen im Wesentlichen die in 3.2.3 zusammengestellten Anforderungen an eine Plattform zur verteilten Modellierung mit SemTalk.²⁶⁴ In der Kategorisierung von [BearingPoint 2003b] können sie größtenteils den anwendungsbezogenen Kriterien²⁶⁵ zugeordnet werden. Sie beinhalten die An-

²⁶³ [Jung 2001:71], [Stahlknecht; Hasenkamp 2005:252]

²⁶⁴ Virtueller Projektraum und Workgroup Funktionalität, Werkzeuge, Verteilte Modellierung, Knowledge Ausstattung und Ereignisverwaltung.

²⁶⁵ In [Stahlknecht; Hasenkamp 2005:300] als *fachinhaltliche* Kriterien bezeichnet.

forderungen, die den funktionalen Leistungsumfang der Plattform, die Unterstützung der verteilten Modellierung und die Integration von Anwendungen (wie SemTalk) betreffen.

Daneben sind bisher noch nicht angesprochene Anforderungen wichtig und müssen im Kontext der Bewertung berücksichtigt werden. Hierzu zählt der *Lernaufwand*, der nötig ist, um das Anwendungsgefüge aus SemTalk und WSS anzuwenden zu können. Er korreliert stark mit dem zu erwartenden Nutzen, da er sich negativ auf die Akzeptanz der Anwender auswirkt. Von dem Aufwand zum Ausbau des Funktionsumfangs ist abhängig, inwieweit sich die Windows SharePoint Services an die speziellen Gegebenheiten der verteilten Modellierung anpassen lassen. Wegen seiner Bedeutung ist der *Aufwand zur Anpassung* die zweite Anforderung, die neben den schon bekannten Kriterien in den Katalog aufgenommen wird.

Der Katalog der Bewertungskriterien ließe sich um beliebig viele Kriterien ergänzen, die das zu bewertende System mehr oder weniger brauchbar erscheinen lassen.²⁶⁶ Die grundlegende Kritik die an dieser Bewertungsmethode geäußert werden kann, ist die Subjektivität bei der Gewichtung der Kriterien sowie der anschließenden Festlegung des Zielerreichungsgrades. Aber in Ermangelung anderer brauchbarer Ansätze ist die Nutzwertanalyse die in den Unternehmen am häufigsten eingesetzte Methode und wird darum auch hier angewandt.

(2) Gewichtung der Bewertungskriterien. Zur Feststellung des Gewichts der einzelnen Kriterien müssen diese in einer Matrix einander gegenüber gestellt werden [Bearing Point 2003b:9]. Darin ist jedes Kriterium (y-Achse) mit jedem anderen Kriterium (x-Achse) zu vergleichen und festzustellen, ob es wichtiger (2), gleichbedeutend (1) oder unwichtiger (0) als das ihm gegenübergestellte Kriterium ist. Die Summe der Werte aus der Zeile ergeben das absolute Gewicht des einzelnen Kriteriums innerhalb des Kriterienkatalogs. Die für diesen erstellte Gewichtungstabelle ist im Anhang VI zu finden.

(3) Bewertung der Veränderung. Soll ein optimales System aus einer Menge von Angeboten ausgewählt werden, sind alle Systeme in die Nutzwertanalyse einzubeziehen. Unter Anwendung unterschiedlicher Regeln lässt sich anhand des Grades der Zielerreichung der Systeme das einzuführende System ermitteln [Bearing Point 2003b:12ff]. Im Gegensatz dazu wird für die Feststellung des Nutzenkoeffizienten bei der Einführung eines bestimmten Systems dem IST-Zustand die zu erwartende Zustandsänderung gegenübergestellt. Die Skala der Bewertung reicht darum von -3 (wesentliche Verschlechterung) über 0 (keine Veränderung) bis zu +3 (starke Verbesserung).

Die Bewertung erfolgt auf Grundlage einiger prototypisch erstellter Modellierungsszenarien und den Erfahrungen, die während der Umsetzung und der Testphase der in dieser

²⁶⁶Man könnte die Anforderung formulieren, dass die technische Plattform zur verteilten Modellierung systemunabhängig sein solle. Das würde die Bewertung der WSS deutlich verschlechtern, da sie an das Microsoft Windows Server-System gebunden sind.

Arbeit entworfenen Komponenten gemacht werden konnten. Weitere Erkenntnisse wurden bei der Entwicklung von Webparts für den SPS gewonnen.²⁶⁷

Kriterium	Gewichtung	Grad der Änderung	Produkt
<i>Virtueller Projektraum</i>			
Anzeige der Anwesenheit	3	0	0
Gruppenräume	18	3	54
Hohe Verfügbarkeit	40	2	80
Einfache Bedienung	38	2	76
Diskussionsrunden	37	2	74
Besprechungsräume	8	2	16
Aufgabenplanung	35	2	70
Konferenzen	8	0	0
Argumentation Framework	7	1	7
<i>Werkzeuge</i>			
Integration von SemTalk	40	2	80
Integration von MS Office	28	2	56
Integration heterogener Dokumenttypen	27	1	27
Benutzerverwaltung	42	3	126
Vorausansicht von Modellen	5	0	0
<i>Verteilte Modellierung</i>			
Versionsbezogene Kommentare	9	1	9
Dateiberechtigungen	38	1	38
Versionsverwaltung für Modelle	45	3	135
Sperrung von Modellen	35	3	105
Überprüfung von Konsistenz	40	-2	-80
Änderungsmitteilung für Referenzmodelle	43	2	86
Änderungsmitteilung für Referenzobjekte	41	2	82
Änderungsmitteilung für externe Verfeinerungen	39	0	0
Import von externen Modellen	48	0	0
<i>Knowledge Ausstattung</i>			
Suche	32	1	32
Beispielbibliothek	12	0	0
<i>Ereignisverwaltung</i>			

²⁶⁷ Diese sind nicht Gegenstand der vorliegenden Arbeit.

Kriterium	Gewichtung	Grad der Änderung	Produkt
Kalender/gemeinsame Termine	42	3	126
Meilensteine	15	1	15
Bekanntmachungen	10	1	10
<i>Sonstige</i>			
Lernaufwand	43	-2	-86
Aufwand zur Anpassung	38	-2	-76
<i>Summe</i>	866		1062

Tafel 10

(4) Berechnung des Nutzenkoeffizienten. Im ersten Schritt wird das Produkt aus Gewicht des Kriteriums und dem Grad der erwarteten Veränderung bei Einführung berechnet. Anschließend ist der Quotient aus der Summe der Produkte und der Summe der Gewichte zu errechnen. Den Ergebnissen aus Tafel 10 zufolge ergibt sich folgende Rechnung:

$$\text{Nutzenkoeffizient} = 1062 / 866 = 1,23$$

Der errechnete Nutzenkoeffizient ist größer als 1, was bedeutet, dass mit der Einführung und Nutzung der WSS als Plattform für die verteilte Modellierung mit SemTalk ein positiver Nutzen entsteht. Besonders positiv haben sich die umfangreich vorhandenen Groupware-Funktionen (Gruppenräume, Termine, Aufgabenplanung) und Werkzeuge für die Verwaltung gemeinsamer Dokumente (Versionsverwaltung und Zugriffssperre) auf das Ergebnis ausgewirkt. Der hohe Lernaufwand, der Aufwand zur Anpassung der WSS und die naturgemäß noch nicht ausreichend implementierte Unterstützung der Mechanismen der verteilten Modellierung belasten hingegen das Ergebnis.

Das Verhältnis hätte wesentlich besser ausfallen können, wenn Microsoft die Anpassbarkeit der WSS besser unterstützen würde und damit die Umsetzung von Lösungen für die verteilte Modellierung einfacher wäre. Die meist umfangreich vorhandene Dokumentation zum System und die Möglichkeiten zu dessen Anpassung und Erweiterung sind auf Grund des mangelhaften methodisch-didaktischen Aufbaus oft schwer zu verstehen. Allein die Aufarbeitung der verfügbaren Quellen würde die Akzeptanz erhöhen und vermutlich zu einer stärkeren Anwendung der WSS in den Unternehmen führen.

5.2 Ausblick

Die verteilte Modellierung stellt an ihre Durchführung verschiedene Ansprüche, die sich auf die Möglichkeiten der Kopplung von separaten Modellen, die angewandte Methodik und Organisationsform sowie die eingesetzte technologische Plattform beziehen. In diesem Zusammenhang wurde gezeigt, dass drei dieser vier Anforderungsbereiche – die Herstellung von verteilten Modellen, das Vorgehen und die Organisation – bereits weitestgehend diskutiert oder technisch umgesetzt sind. Anforderungen, die an die technologische Plattform zu stellen sind, wurden bei verteilt durchgeführten Modellierungs-Vorhaben formuliert, konnten aber in Ermangelung einer geeigneten Plattform bislang nicht genauer überprüft werden. Aus diesem Grund sind in dieser Arbeit die Windows SharePoint Services vorgestellt und ihre Eignung als Plattform für die verteilte Modellierung untersucht worden.

Für die Bewertung der Lösung waren die festgestellten Anforderungskriterien und der von den erweiterten WSS erreichte Erfüllungsgrad maßgebend. In der ersten Gegenüberstellung der Anforderungen mit den standardmäßig vorhandenen Funktionen konnte gezeigt werden, dass ein Großteil der Anforderungen von den WSS bereits erfüllt wird. Die nicht abgedeckten Anforderungen waren die Grundlage für den Entwurf einer Lösung, die – basierend auf Technologien der WSS – deren Funktionen erweitert. Der in diesem Zusammenhang untersuchte umfangreiche Fundus an Techniken gestattet eine Vielzahl von Lösungsansätzen.

Wegen des begrenzten Rahmens konzentrierte sich die entworfene Lösung auf die Verwaltung von externen Referenzen – einer speziellen Form der Kopplung separater Modelle durch SemTalk. Die Lösung besteht aus einer Ereignisroutine (*event handler*), die einen Webdienst anstößt. Erst mit dieser zusätzlich geschaffenen Erweiterung ist es möglich, mehrwertige Funktionen für die verteilte Modellierung umzusetzen. Die hier prototypisch realisierten Funktionen sehen vor, dass Modell-Eigentümer in einem Projekt verteilter Modelle über Änderungen in fremden Modellen, sofern es für sie von Bedeutung ist, automatisch informiert werden. Diese Funktion war mit dem bisherigen System nicht verfügbar.

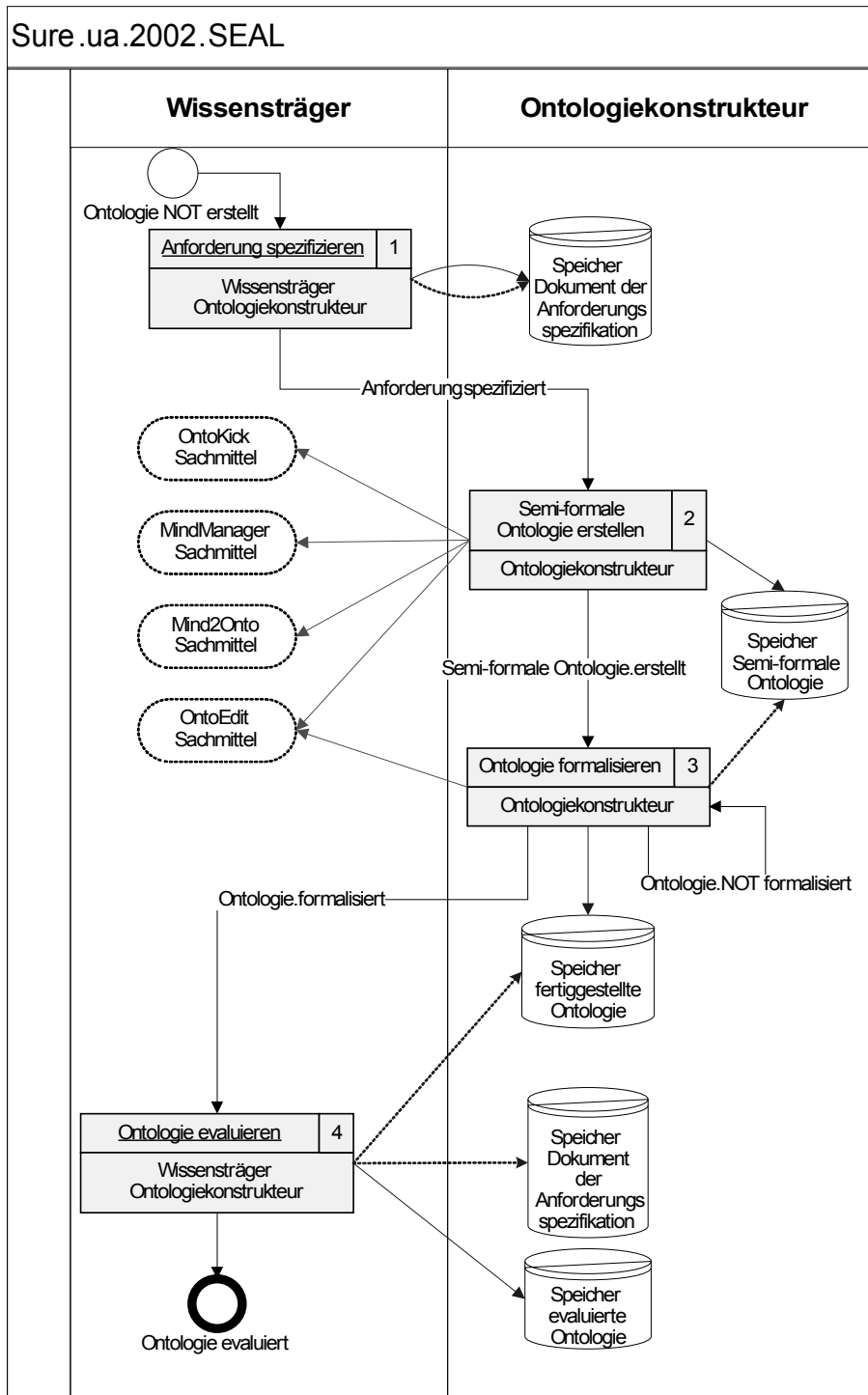
Weitere, in der Zukunft umzusetzende Aufgaben bestehen aus der Verwaltung weiterer Kopplungsformen separater Modelle – den externen Verfeinerungen und den *externen Verfeinerungen von* – sowie der Implementierung von Automatismen und Funktionen zur Wahrung der begrifflichen und inhaltlichen Konsistenz in einer Menge von verteilten Modellen. Dafür konnten hier nur die grundlegenden Anwendungsfälle angeführt werden.

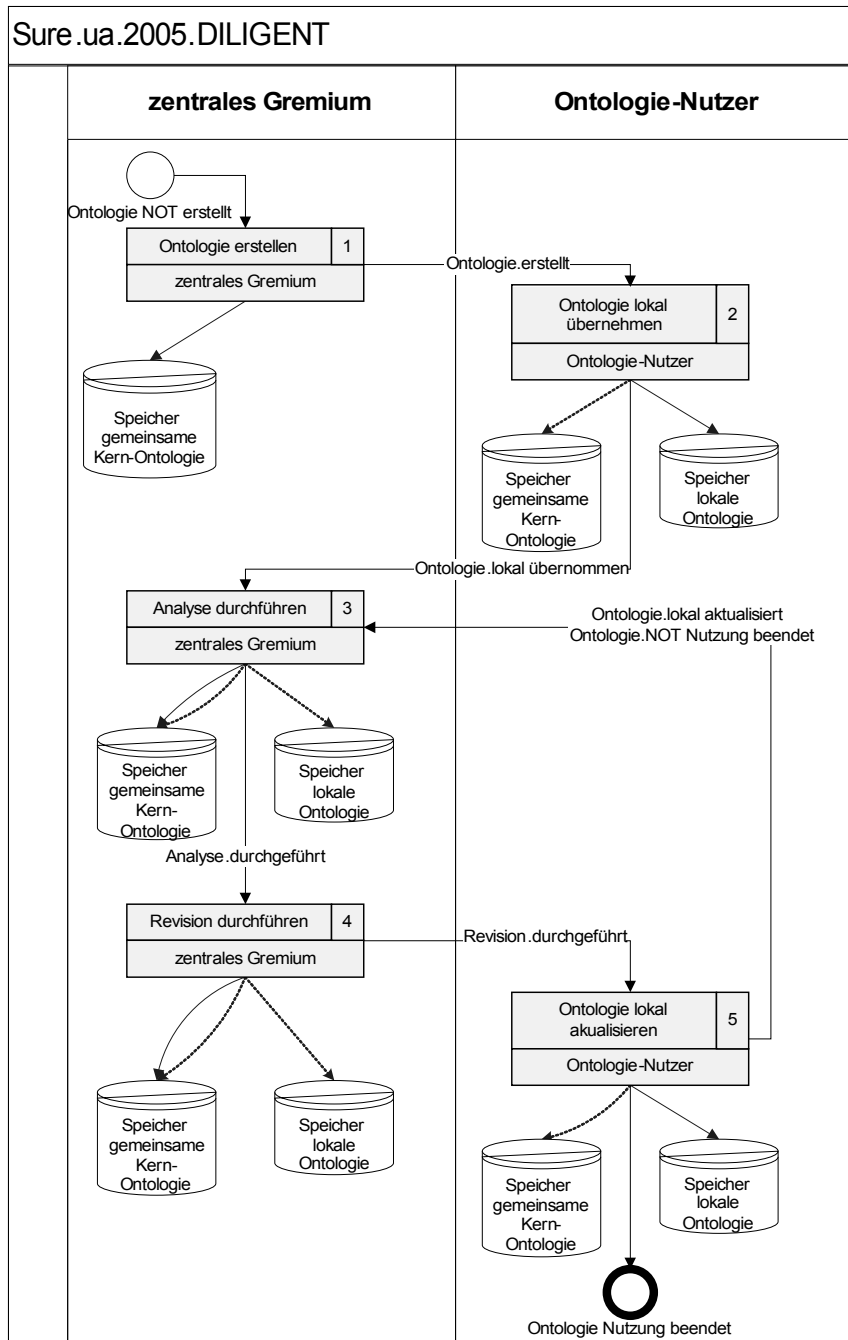
Die in dieser Arbeit umgesetzte Lösung zeigt, dass aus der Kombination des Modellierungswerkzeugs SemTalk mit den Windows SharePoint Services als Plattform und den assistierenden Arbeiten von [Simon 2004] und [Weichhardt 2005] ein um-

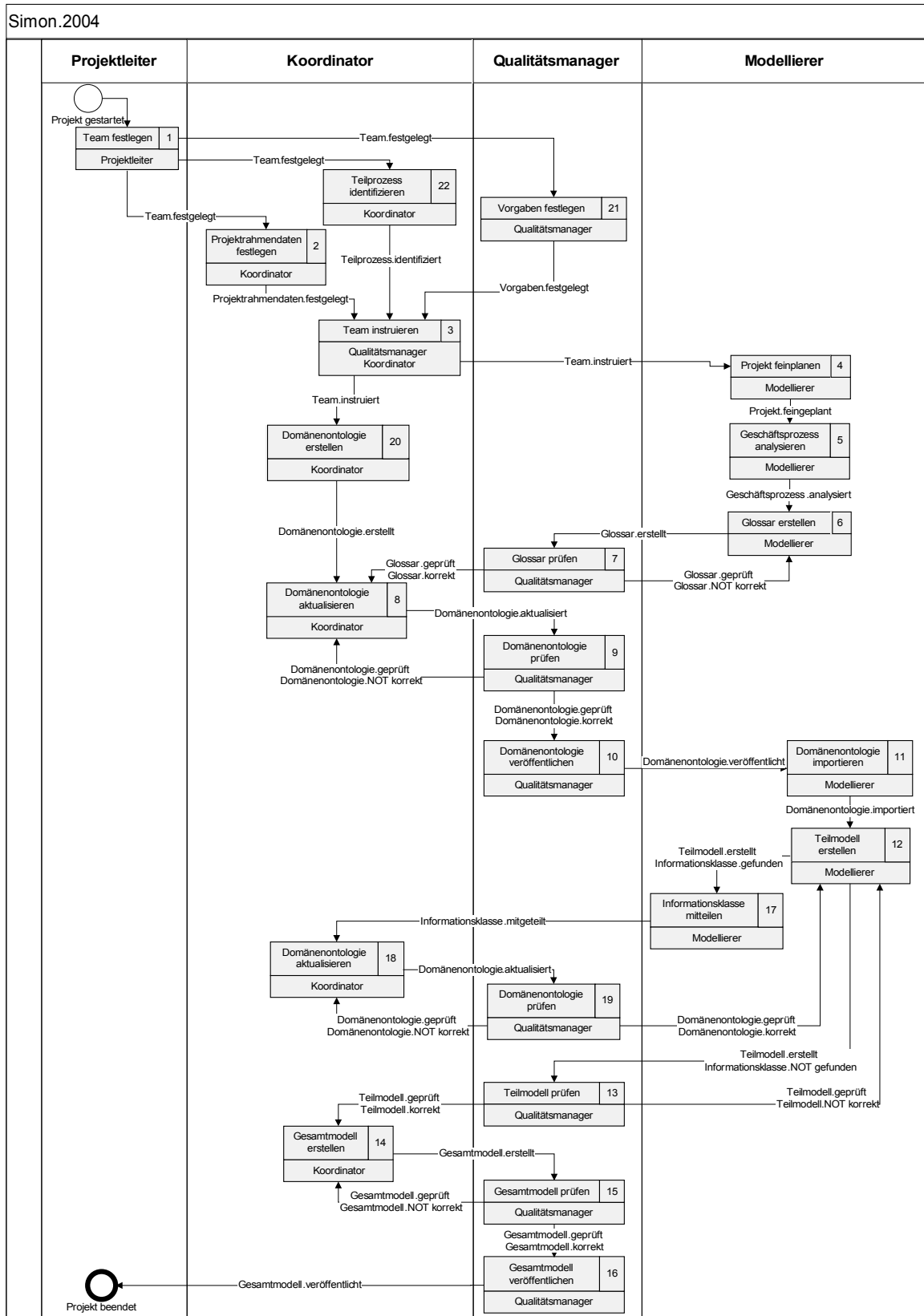
fassendes Angebot für die verteilte Modellierung bereitsteht. Der positive Nutzen, der sich mit der Verwendung der WSS als Plattform für die verteilte Modellierung ergibt, konnte mit Hilfe der Verwendung von Methoden zur Betrachtung der Wirtschaftlichkeit und Effekte (Portfolioanalyse und Nutzwertanalyse) nachgewiesen werden. Dieser ließe sich zukünftig steigern, indem weitere Anpassungen und Erweiterungen umgesetzt würden. Die dafür zur Verfügung stehenden technischen Möglichkeiten und die noch nicht gelösten Anwendungsfälle wurden in dieser Arbeit gezeigt. Des Weiteren stehen die nächsten Programmversionen des Microsoft Office Systems und damit auch der WSS an. Zukünftig kann also mit besseren, als den hier festgestellten Ergebnissen gerechnet werden.

Der in dieser Arbeit nachgewiesene Nutzen könnte verzögert eintreten, wenn sich die Anwender noch in dem Stadium der Erlernung des Modellierungswerkzeugs und der Modellierungsmethode befinden. Erst wenn in diesen Bereichen die nötige Sicherheit bei der Benutzung herrscht, sind die modellierenden Anwender befähigt, neue Funktionen und Technologien – wie eine gemeinsame Plattform auf Basis der WSS – effizient zu verwenden.

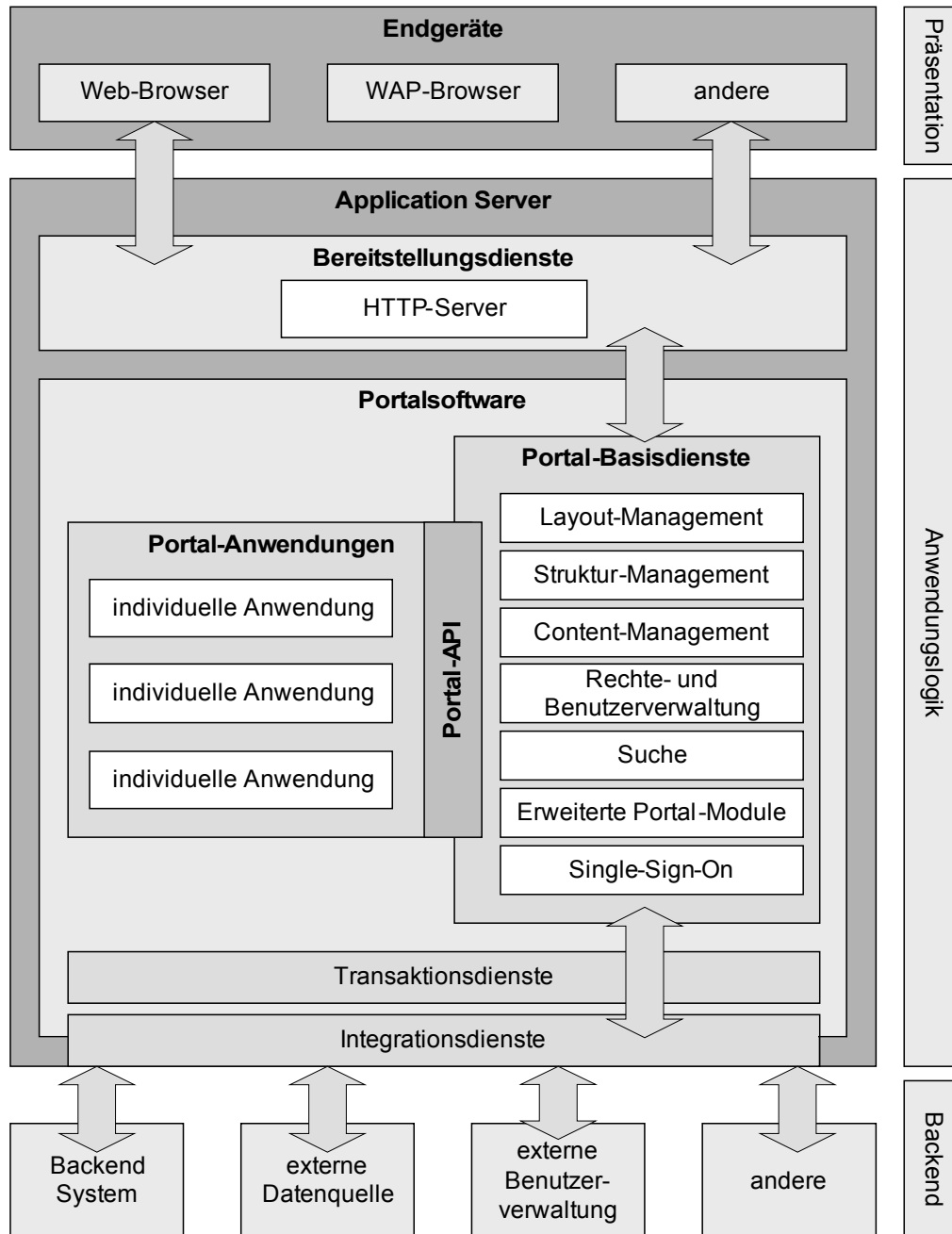
Anhang I – Vorgehensmodelle







Anhang II – Referenzarchitektur für Portalsoftware



Quelle: [Gurzki; Hinderer 2003:158]

Anhang III – Fragebogen

Gliederung:

- i. Allgemein zum Modellierungsvorhaben
- ii. Organisation
- iii. Funktionsbereich, eigene Tätigkeit sowie vor- und nachgelagerte Schritte
- iv. Verteilte Modellierung mit SemTalk
- v. Daten
- vi. IT
- vii. Plattform für die verteilte Modellierung

i. Allgemein zum Modellierungsvorhaben

- 1 Wie lautet das Modellierungsvorhaben und was ist das Ziel?
 - 2 Seit wann führen Sie das Vorhaben durch?
-

ii. Organisation

- 3 Wieviele Personen sind an dem Modellierungsvorhaben beteiligt?
 - 4 Wie groß ist der zu modellierende Unternehmens-Bereich?
 - 5 Wie ist das Modellierungsvorhaben personell organisiert?
Organigramm mit Leiter, Stellen... kurz beschreiben!
 - 6 Wer hat im Rahmen des Modellierungsvorhabens welche Befugnisse?
 - 7 Können Sie Projektergebnisse herausgeben?
-

iii.(a) Funktionsbereich – Eigene Tätigkeit

- 8 Was sind Ihre Aufgaben innerhalb der Modellierungsvorhabens? Welches Ziel hat Ihre Tätigkeit?
 - 9 Welche Arbeiten führen Sie dabei täglich aus?
 - 10 Welche Sach-, Arbeits- und Hilfsmittel benötigen Sie?
 - 11 Wo befinden Sie sich bei Ihrer Aufgabenerfüllung?
-

iii.(b) Funktionsbereich – vor- und nachgelagerte Arbeitsschritte

- 12 Wie sieht Ihr Vorgehen im Rahmen der verteilten Modellierung aus?
- 13 Welche Arbeitsschritte sind Ihrer Modellierung vor- bzw nachgelagert?
- 14 Wer trägt für diese vor- bzw nachgelagerten Schritte die Verantwortung?
- 15 Welcher Organisationseinheit gehören diese Personen an?

iii.(b) Funktionsbereich – vor- und nachgelagerte Arbeitsschritte

16 Wie findet der Austausch von Informationen zwischen Ihnen und diesen Personen statt?

iv. Verteilte Modellierung mit SemTalk

17 Kennen Sie die technischen Möglichkeiten von SemTalk verteilt zu modellieren? Wenn ja, welche der folgenden:

- Anzeigen und Einfügen externer Objekte
- Externe Verfeinerungen mit Kenntnis des verfeinerten Modells (externe Verfeinerung)
- Externe Verfeinerung ohne Kenntnis des verfeinerten Modells (externe Verfeinerung von)
- Import von Teilmodellen zu einem Gesamtmodell
- Verlinkung von Objekten separater Modelle mittels externer Referenzen

18 Welche der soeben genannten Möglichkeiten nutzen Sie?

19 Wie verwalten Sie globale Objekte, welche über das gesamte Projekt Bedeutung haben?

20 Führen Sie regelmäßige Abstimmungsprozesse über globale Objekte und Modellschnittstellen durch? Wenn ja, beschreiben Sie kurz:

- Methodik
- Zyklen

21 Wie werden die Abstimmungsprozesse zwischen den Modellierern durchgeführt?

22 Mit welchen Mitteln wird das Projektergebnis präsentiert?

v. Daten

23 Benötigen Sie für die verteilte Modellierung Informationen Anderer?
(zB Teil- oder Referenzmodelle, Dokumente)

24 Können Sie Aussagen darüber treffen, zu welchem Zeitpunkt Sie diese Informationen benötigen?

25 Auf welche Art erlangen Sie Zugriff auf die von Ihnen benötigten Informationen?

26 Resultieren aus Ihrer Projektstätigkeit neue Modelle/Informationen oder werden existierende Modelle/Informationen verändert?

27 Falls neue Daten erzeugt werden, welcher Art sind diese?

28 Falls verändert: inwieweit verändert?

29 Greift jemand auf die von Ihnen bearbeiteten Modelle/Informationen zu?

30 Wie gelangen diese zu den betreffenden Personen?

vi. IT

31 Besteht eine Verbindung zu anderen Rechnern? Wie ist sie realisiert?

32 Welche Software (außer SemTalk und MS Visio) nutzen Sie, bezogen auf Ihre Projektstätigkeit?

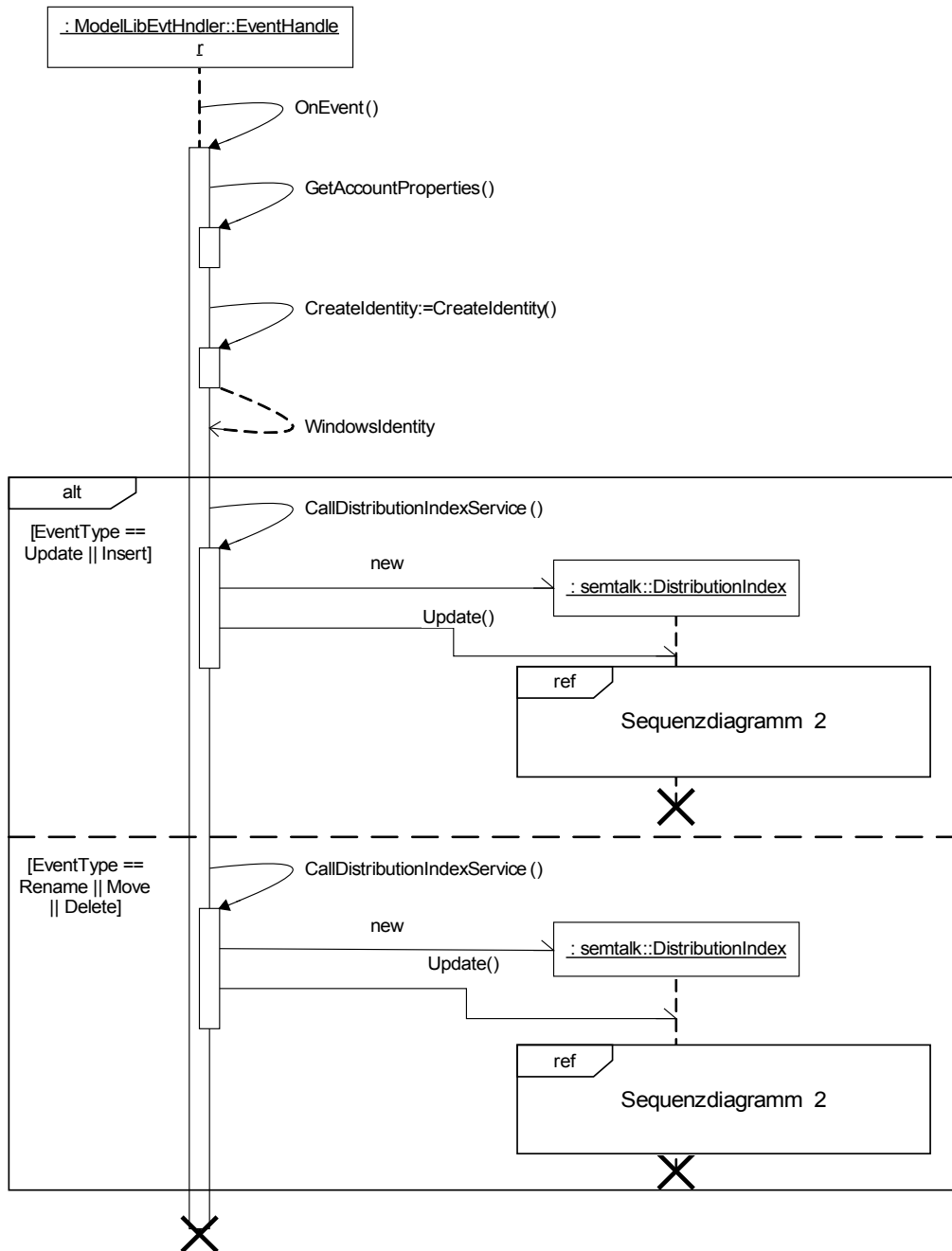
33 Haben und nutzen Sie die Dokumentation von SemTalk?

vii. Plattform für die verteilte Modellierung

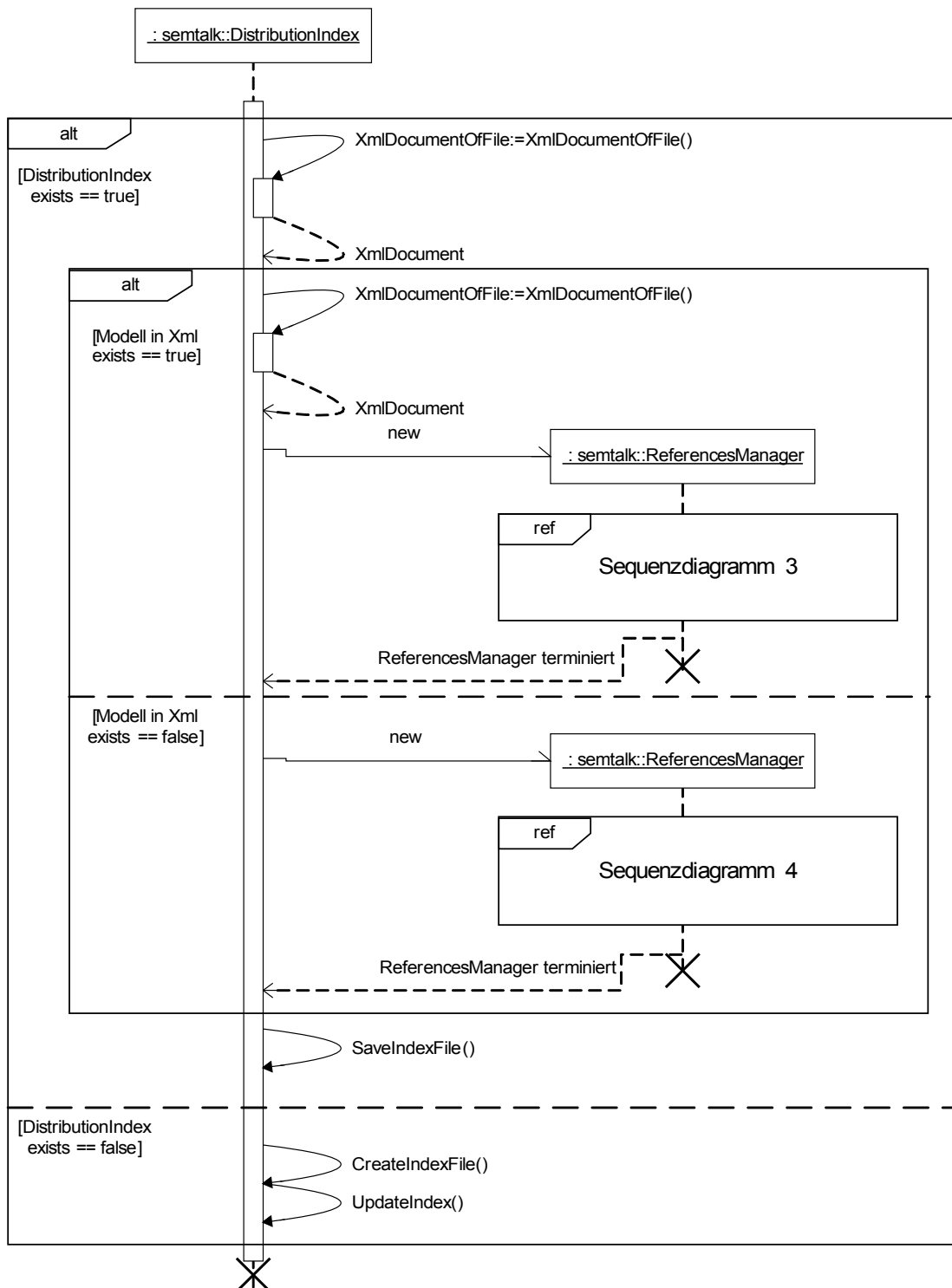
- 34 Konnten Sie Schwierigkeiten bei der verteilten Modellierung feststellen? Wenn ja, waren diese organisatorischer/methodischer/oder technischer Art?
- 35 Könnten Sie bitte kurz auf diese Schwierigkeiten eingehen?
- 36 Denken Sie, dass eine Plattform zur Zusammenarbeit die bisherigen Probleme bei der verteilten Modellierung verhindern oder vorbeugen könnte?
- 37 Was müsste ihrer Meinung nach eine unterstützende Plattform für die verteilte Modellierung leisten, die technischen Erfordernisse ausgenommen?
- 38 Welche Informationen sollte Ihnen die Plattform liefern, um Ihre Tätigkeit besser ausführen/koordinieren zu können?
- 39 Was wäre „nice to have“?
-

Anhang V – UML Sequenzdiagramme

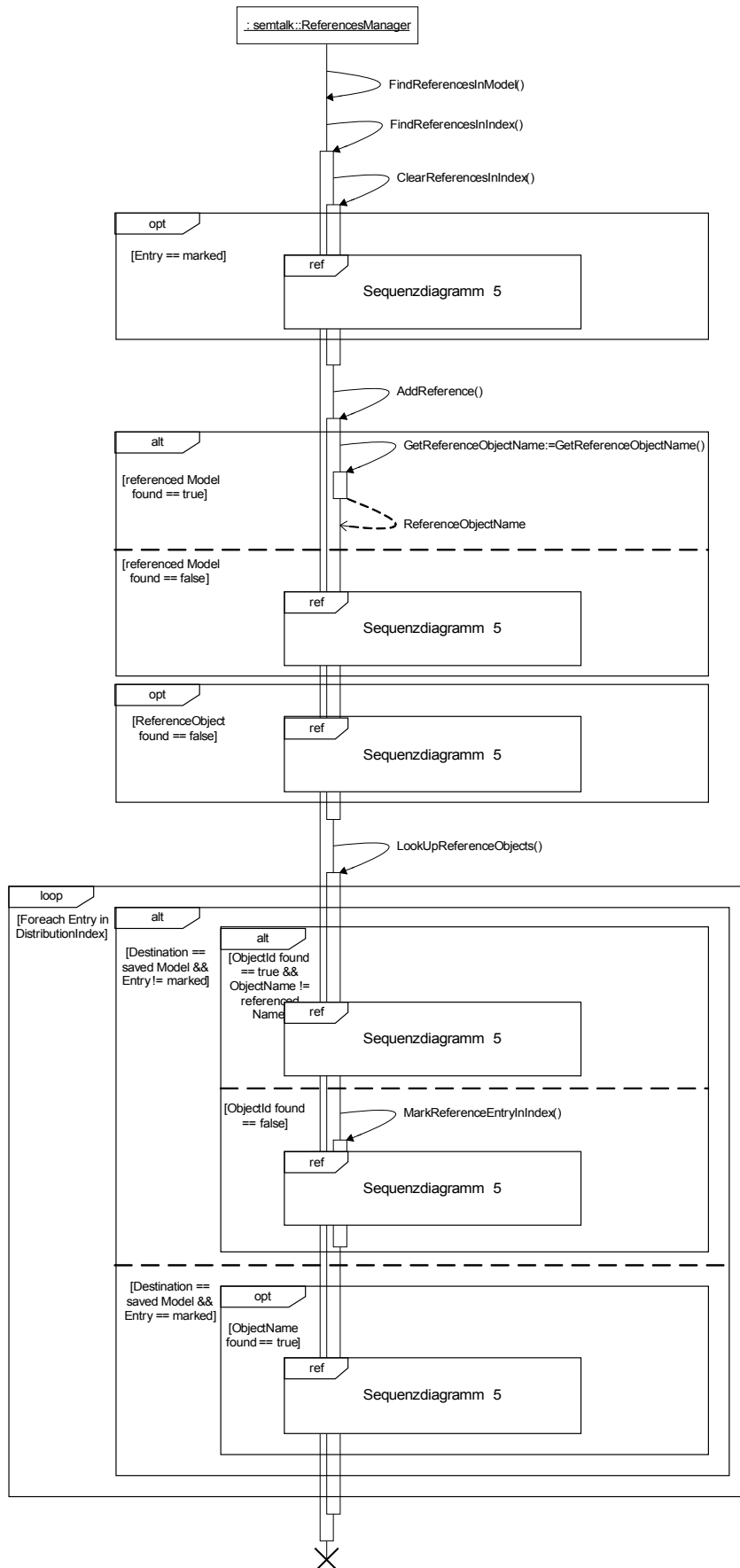
Sequenzdiagramm 1



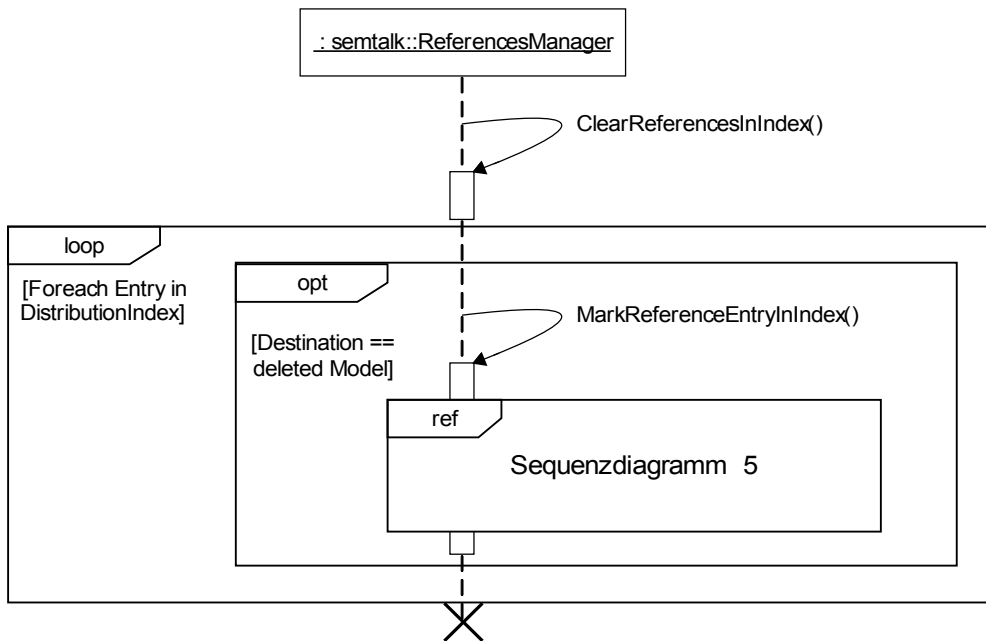
Sequenzdiagramm 2



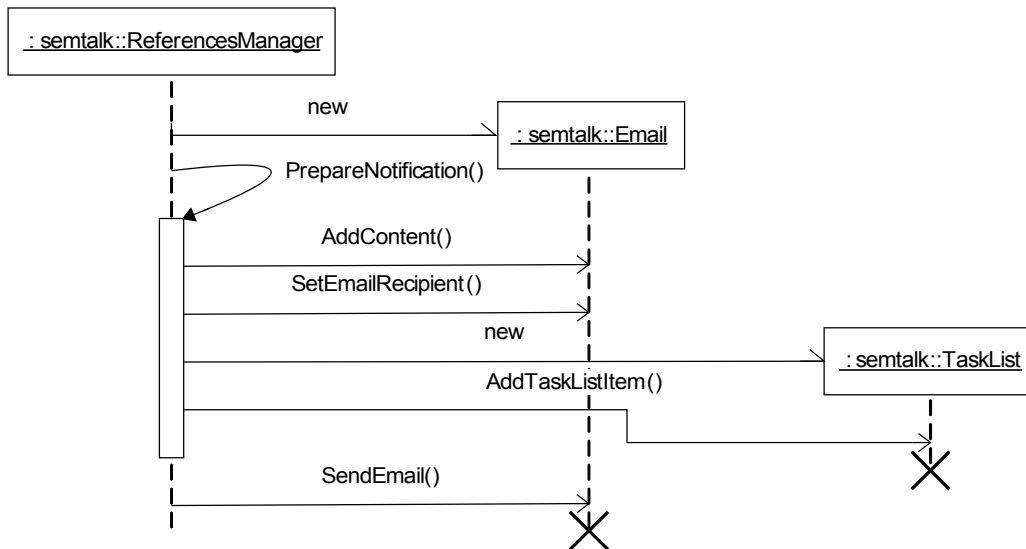
Sequenzdiagramm 3



Sequenzdiagramm 4



Sequenzdiagramm 5



Glossar

Begriff	Erläuterung
Anschlussnummer	Eng <i>Port</i> . Dient der konkreten Adressierung einer Netzwerk-Verbindung. Der Client hat bei dem Versuch, eine Verbindung mit dem Server aufzubauen, die A anzusprechen, an der der Server-Prozess auf Anfragen wartet.
Anwendungspool	Container für nicht-statische Programmteile einer Webseite im IIS. Jeder A wird von einem separaten Prozess ausgeführt. Somit liegt eine logische Trennung zwischen verschiedenen A vor – Probleme eines A wirken sich nur auf ihn aus.
Application Server	In der Drei-Schichten-Architektur (Präsentation, Verarbeitung, Datenhaltung) übernimmt der AS die Bereitstellung und Ausführung von Anwendungslogik. Der clientseitige Aufruf sowie die Eingabe und Ausgabe von Daten erfolgt über standardmäßige Netzwerkprotokolle (zB SOAP, XML-RPC, CORBA). Die Präsentation übernimmt der Client (oft mit dem Internet Browser) und die Datenhaltung ein Datenbanksystem.
Assemblierung	Eng <i>assembly</i> . In der Softwareentwicklung: Ein Container für alle Komponenten (Quellcode, Referenzen und Ressourcen), die eine .NET-Anwendung benötigt.
Basisontologie	Ontologie einer höheren Abstraktionsebene; wird oft auch als Oberontologie bezeichnet.
Branchen-Referenzmodell	Referenzmodell einer Unternehmensbranche
Client	„Anwendung, die den Dienst eines Servers in Anspruch nimmt“ [http://de.wikipedia.org/wiki/Clients] (3.4.2006). Wird oft synonym für einen Arbeitsplatz-Rechner verwendet, der anderen C keine Dienste anbietet.
Diagramm	Teil eines Modells. In MS Office Visio in Form eines Zeichenblattes (eng <i>sheet</i>). Ein MS-Office-Visio-Dokument kann aus mehreren Zeichenblättern bestehen.
Dokumentarbeitsbereich	Ein gemeinsamer Arbeitsbereich, der sich auf die Bearbeitung eines Dokuments konzentriert [http://office.microsoft.com/de-de/assistance/HP030923041031.aspx] (3.4.2006).
Dokumentbibliothek	Spezielle Art einer Liste in den WSS. Wird zur Verwaltung von gemeinsamen Dokumenten verwendet.
Dokumentenmanagementsystem	Kurz: DMS. Dient der „datenbankgestützten Verwaltung elektronischer Dokumente“ [http://de.wikipedia.org/wiki/Dokumentenmanagement] (3.4.2006).
Domänenontologie	Ontologie eines Sach- oder Wissensgebietes.
Epistemologie	Die Erkenntnistheorie; eine zentrale Disziplin der Philosophie neben der Ethik und der Logik [http://de.wikipedia.org/wiki/Epistemologie] (3.4.2006).
Ereignisroutine	Eng <i>event handler</i> . Bei Ereignissen (zB eine Benutzereingabe,

Begriff	Erläuterung
Freigegebener Arbeitsbereich	ein bestimmter Zeitpunkt) automatisch ausgeführter Programmcode. Synonym von gemeinsamer Arbeitsbereich, ist Titel einer einblendbaren Benutzungsoberfläche in MS-Office-2003-Produkten.
Gemeinsamer Arbeitsbereich	Ein für eine Arbeitsgruppe logisch abgegrenzter Bereich in einem (um die WSS-Anwendung erweiterten) virtuellen Server; realisiert mit Hilfe einer WSS-Webseite.
Generalisierung	Beziehung zwischen generellen und speziellen Klassen. Spezielle Klassen sind „vollständig konsistent“ mit der generellen Klasse, können aber zusätzliche Informationen haben [Balzert 2005:52].
Hypertext Transfer Protocol	Kurz: HTTP; Netzwerk-Protokoll, das der Übertragung von Daten dient.
IP-Adresse	Numerische Adressierung von Rechnern in IP-Netzwerken.
IP, Internet Protocol	Ein Netzwerkprotokoll; <i>Vermittlungsschicht</i> im ISO/OSI-Referenzmodell (darunter nur noch Sicherungsschicht und Bitübertragungsschicht).
Klasse	In der Objektorientierung: Ein statisches Muster, von dem konkrete Ausprägungen (Objekte oder Instanzen) erzeugt werden.
Kompilierung	Übersetzung von programmiertem Quelltext in eine Zielsprache (meist ausführbarer Maschinencode) mit Hilfe eines Compilers.
Konsistenz	„Widerspruchsfreiheit in einem [...] System“, etwas ist „in sich stimmig“, „Benennungen [sind] richtig, einheitlich“ [http://de.wikipedia.org/wiki/Konsistenz] (5.4.2006)]. Konsistenz zwischen verteilten Modellen: gleiche Dinge werden in separaten Modellen gleich benannt.
Konstruktor	In der Softwareentwicklung: Methode einer Klasse oder Variable, die immer bei der Initialisierung ausgeführt wird. Sie versetzt das Objekt in einen Anfangszustand [http://de.wikipedia.org/wiki/Konstruktor] (5.4.2006)]. Mittels der <i>Überladung</i> ist die Einrichtung mehrerer Konstruktoren möglich. Sie unterscheiden sich durch ihre Signatur (Art und Anzahl der zu übergebenden Parameter) voneinander.
Liste	Im Begriffsraum von SharePoint: In ASP.NET entwickeltes Oberflächenelement, das Daten in Abhängigkeit einer Sicht anzeigt. Beides, die angezeigten Daten wie die Sicht, kann angepasst werden.
Lokalisierung	Anpassung eines Programms an die sprachlichen Gegebenheiten eines Kulturkreises.
Microsoft Office System	Produktpalette von Büroanwendungen der Firma Microsoft, bestehend aus den clientseitigen Standard-Anwendungen

Begriff	Erläuterung
Microsoft Office System Integration	und Zusatz-Anwendungen sowie Server und Services [http://www.microsoft.com/germany/office/ueberblick/erklaerung.mspx] (5.4.2006). In einer MS-Office-Anwendung eingebettete Funktionen anderer MS-Office-Anwendungen.
Modell	Grafische Darstellung eines Sachverhalts. Siehe auch SemTalk-Modell.
Namensraum	Bezeichnet den Kontext, dh den inhaltlichen Zusammenhang, in dem der Name eines Objektes Verwendung findet [http://de.wikipedia.org/wiki/Namensraum] (18.4.2006). Der Namensraum wird dem Klassennamen durch einen Punkt getrennt vorangestellt.
Objekt	In SemTalk: Alle Elemente eines Modells; siehe auch Objektorientierung.
Objektorientierung	Ein Paradigma, in dem alle Dinge Objekte sind, die ein Verhalten (Methoden) ausweisen und Eigenschaften (Attribute) besitzen. Siehe Klasse.
Ontologie	Konzeptionelles Schema (einer Domäne) [Gruber].
Ontologie-Nutzer	Benutzer einer Ontologie (idR die Mitarbeiter eines Unternehmens), eng <i>ontology user</i>
Ontologiekonstrukteur proprietär	Experte im Erstellen von Ontologien, eng <i>ontology engineer</i> Entweder urheberrechtlich geschütztes Material, nicht frei programmierbare Dateiformate oder Dateiformate, die nicht anerkannten Standards entsprechen. Ausführlichere Erläuterungen unter: http://de.wikipedia.org/wiki/Propriet%C3%A4r (5.4.2006).
Referenzmodell	grafische Informationsmodelle für allgemeine Daten, Organisationen, Prozesse und Funktionen, die Ausgangspunkt für konkrete Anwendungsmodelle sind.
Referenzvorgehensmodell	Dokumentation von typischen Abläufen eines Projekts.
SemTalk-Modell	Ein mit SemTalk erzeugtes Modell, das aus Objekten besteht, die auf Diagrammen angeordnet sind. Das SemTalk-Modell wird in einem MS-Office-Visio-Dokument gespeichert.
SemTalk-Objekt	Jedes Element eines SemTalk-Modells (zB: Klassen, Instanzen, Diagramme, Beziehungen)
Server	Anwendung, die auf Anfragen eines Clients wartet und daraufhin mit ihm Nachrichten austauscht [http://de.wikipedia.org/wiki/Server] (3.4.2006). Wird häufig als Synonym für einen Rechner verwendet, der vornehmlich Dienste anbietet.
Spezialisierung	Erzeugung von Spezialfällen einer generellen Klasse, die alle Eigenschaften und das Verhalten der generellen Klasse erben (= Vererbung). Diese Unterklassen können zudem eigene Eigenschaften und eigenes Verhalten besitzen.

Begriff	Erläuterung
Starker Name	Eng <i>strong name</i> . Besteht aus Bezeichnung und Version der Assemblierung, dem öffentlichen Schlüssel sowie der Angabe des Kulturkreises. Wird bei der Signierung einer Assemblierung mit dem Public-Private-Key-Schlüsselpaar erzeugt.
Taxonomie	Eine Hierarchie von Begriffen.
Thesaurus	Taxonomie einer Domäne mit Relationen "Ähnlichkeit" und "Synonym" zwischen den Begriffen.
Topic Map	Abstraktes Modell zur Formulierung von Wissensstrukturen, bestehend aus Themen, Assoziationen, Ausprägungen (in Dokumenten).
UML	Unified Modeling Language; Methoden zur objekt-orientierten Analyse und Entwurf (= Notation); von der Object Management Group (OMG) standardisiert und aktuell in der Version 2.0 verfügbar.
URL	Uniform Resource Locator (URL) dient der Adressierung einer Ressource, zB einer Webseite oder eines Dokuments im Internet.
Usenet	Im Internet verfügbares <i>schwarzes Brett</i> ; Bereitstellung asynchroner Nachrichten; Diskussionen finden themenbezogen statt [Borghoff; Schlichter 1998:119].
Versionsverwaltung	Zentrales System zur Bereitstellung von Dateien, das Veränderungen erfasst und Versionsstände erzeugt [http://de.wikipedia.org/wiki/Versionsverwaltung] (5.4.2006).
Virtueller Server	Verhält sich aus Anwendersicht wie ein physikalischer Server. Wird von einem physikalischen Server bereitgestellt. Im IIS: „Virtuelle Server auf Basis der Trennung von Diensten“ [http://de.wikipedia.org/wiki/Virtueller_Server] (5.4.2006). Das heißt, mehrere virtuelle Webserver werden von einem physikalischen Webserver (IIS) bereitgestellt.
Vorgehensmodell	siehe Referenzvorgehensmodell
Webdienst	Programm, das von einem Application Server ausgeführt und einem Webserver veröffentlicht wird. Dateneingabe und -ausgabe erfolgt mittels standardisierter Internetprotokolle.
Webpart	Oberflächenelement in SharePoint-Produkten, in dem Anwendungslogik implementiert sein kann und das sich in einen gemeinsamen Arbeitsbereich einfach hinzufügen und entfernen lässt. In anderen Portalen als <i>Portlet</i> bezeichnet.
Webservice	Siehe Webdienst.
Wissenskonstrukteur	Eng <i>knowledge engineer</i> . Experte im Erfassen, Formalisieren, Verarbeiten und Darstellen von Wissen [http://de.wikipedia.org/wiki/Wissensmanagement] (5.4.2006),
Wissensträger	Eng <i>domain expert</i> . Person mit Expertenwissen eines Fachbereichs.

Begriff	Erläuterung
Workflow	Arbeitsablauf rechnergestützter Aktivitäten; häufig durch ein Workflow-Management-System (WfMS) und Einbindung von externen Anwendungen unterstützt [http://de.wikipedia.org/wiki/Workflow] (5.4.2006)].
XML	Offenes, dh durch einfaches Einsenden der Datei lesbares Dateiformat, das auf dem SGML-Standard beruht. Es dient der Auszeichnung von Daten. Seine Struktur kann frei festgelegt werden.

Literatur

Gedrucktes Schrifttum

- [Balzert 2005] Balzert, Heide. 2005. *Lehrbuch der Objektmodellierung. Analyse und Entwurf mit der UML 2*. 2. Aufl. München: Spektrum Akademischer Verlag.
- [Bauer 2001] Bauer, Herbert. 2001. *Unternehmensportale: Geschäftsmodelle, Design, Technologien*. Bonn: Galileo Press.
- [Becker; Niehaves; Knackstedt 2004]. Becker, Jörg; Björn Niehaves; Ralf Knackstedt. 2004. „Bezugsrahmen zur epistemologischen Positionierung der Referenzmodellierung“. In: Becker, Jörg; Patrick Delfmann (Hgg.). *Referenzmodellierung. Grundlagen, Techniken und domänenbezogene Anwendungen*. S.1-17. Heidelberg: Physica.
- [Beimborn; Mintert; Weitzel 2002] Beimborn, Daniel; Stefan Mintert; Tim Weitzel. 2002. „Web Services und ebXML“. In: *Wirtschaftsinformatik 44* (2002) Heft 3. S. 277-280. Braunschweig, Wiesbaden: Vieweg.
- [Borghoff; Schlichter 1998] Borghoff, Uwe M.; Johann H. Schlichter. *Rechnergestützte Gruppenarbeit. Eine Einführung in Verteilte Anwendungen*. 2. Aufl. Berlin, Heidelberg: Springer.
- [Burghardt 1997] Burghardt, Manfred. 1997. *Projektmanagement: Leitfaden für die Planung, Überwachung und Steuerung von Entwicklungsprojekten*. 4. Aufl. Erlangen: Publicis Corporate Publishing.
- [Buxmann; Wüstner 2002] Buxmann, Peter; Erik Wüstner. 2002. „SOAP für Web Services“. In: Hartmann-Wendels, Thomas; Rainer Thome; Artur Woll (Hgg.). *WISU. Das Wirtschaftsstudium*. Ausgabe 11/02. S. 1434-1441. Düsseldorf: Lange.
- [De Bono 1993] De Bono, Edward. 1993. *Der klügere gibt nicht nach [I am right you are wrong, deutsch]*. Übersetzt von Till R. Lohmeyer. Düsseldorf, Wien: ECON.
- [Hillier 2005] Hillier, Scot P. 2005. *Advanced SharePoint Services Solutions*. New York: Apress.
- [Jung 2001] Jung, Hans. 2001. *Allgemeine Betriebswirtschaftslehre*. 7. Aufl. München, Wien: Oldenbourg.
- [Klein 1996] Klein, Stefan. 1996. *Interorganisationssysteme und Unternehmensnetzwerke*. Wiesbaden: Dt. Univ.-Verlag.
- [Krallmann ua 1999] Krallmann, Herrmann ua 1999. *Systemanalyse im Unternehmen*. 3.Aufl. München: Oldenbourg.
- [Gierhake 2000] Gierhake, Olaf. *Integriertes Geschäftsprozessmanagement. Effektive Organisationsgestaltung mit Workflow-, Workgroup- und Dokumentenmanagement-Systemen*. 3.Aufl. Braunschweig, Wiesbaden: Vieweg.
- [Gurzki; Hinderer 2003] Gurzki, Thorsten; Henning Hinderer. 2003. „Eine Referenzarchitektur für Software zur Realisierung von Unternehmensportalen“. In: *WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen, Beiträge der 2. Konferenz Professionelles Wissensmanagement, 2.-4. April 2003*. S. 157-160. Bonn: Köller.

- [Mertens ua 2004] Mertens, Peter; Freimut Bodendorf; Wolfgang König; Arnold Picot; Matthias Schumann; Thomas Hess. 2004. *Grundzüge der Wirtschaftsinformatik*. 8.Aufl. Berlin, Heidelberg: Springer.
- [Munkvold 2003] Munkvold, Bjørn Erik. 2003. *Implementing Collaboration Technologies in Industry*. London: Springer.
- [Prosise 2002] Prosise, Jeff. 2002. *Microsoft .NET Entwicklerhandbuch*. Unterschleißheim: Microsoft Press Deutschland.
- [Schwichtenberg 2006a] Schwichtenberg, Holger. 2006. *Bestandteile der .Net-Entwicklungsumgebung. Schichtenmodell*. In: iX spezial 1/2006. S. 6-11.
- [Schwichtenberg 2006b] Schwichtenberg, Holger. 2006. *Sprachen, Werkzeuge und Komponenten für .Net*. In: iX spezial 1/2006. S. 15-28.
- [Simon 2004] Simon, Frank. 2004. *Ein Leitfaden zur verteilten Modellierung von Prozessen mit SemTalk*. Brandenburg: zugängliche Diplomarbeit [Dipl-WInf].
- [Stahlknecht; Hasenkamp 2005] Stahlknecht, Peter; Ulrich Hasenkamp. 2005. *Einführung in die Wirtschaftsinformatik*. 11.Aufl. Berlin, Heidelberg: Springer.
- [Sure ua 2005] Sure, York ua. 2005. „The DILIGENT knowledge processes“. In: *Journal of Knowledge Management*. Vol.9 No.5 2005. S. 85-96.
- [Thomas 2005] Thomas, Oliver. 2005. „Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation“. In: Scheer, August-Wilhelm (Hg.). *Veröffentlichungen des Instituts für Wirtschaftsinformatik*. Heft 184.
- [Vom Brocke 2003a] Vom Brocke, Jan. 2003. *Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen*. Berlin: zugängliche Dissertation.
- [Vom Brocke; Buddendick 2004] Vom Brocke, Jan; Christian Buddendick. 2004. „Konstruktionstechniken für die Referenzmodellierung – Systematisierung, Sprachgestaltung und Werkzeugunterstützung“. In: Becker, Jörg; Patrick Delfmann (Hgg.). *Referenzmodellierung. Grundlagen, Techniken und domänenbezogene Anwendungen*. S.19-49. Heidelberg: Physica.

Ungedrucktes Schrifttum

- [Bearing Point 2003a] Bearing Point GmbH. 2003. „Softwareauswahl und -anpassung. Wirtschaftlichkeitsanalyse“. Präsentation an der FH Brandenburg für den SG Wirtschaftsinformatik im WS 2003/2004.
- [Bearing Point 2003b] Bearing Point GmbH. 2003. „Softwareauswahl und -anpassung. Auswahlprozess“. Präsentation an der FH Brandenburg für den SG Wirtschaftsinformatik im WS 2003/2004.
- [Ceccaroni; Kendall 2002] Ceccaroni, Luigi; Elisa Kendall. 7.11.2002. „A Semantically-Rich, Graphical Environment for Collaborative Ontology Development in Agentcities“.

- <<http://portal.acm.org/citation.cfm?id=860744&coll=GUIDE&dl=ACM&CFID=45696324&CFTOKEN=31928583>> (25.7.2005).
- [Dowler 2003] Dowler, Anthony. 11.12.2003. „Gut, besser, am besten: Die Integration von Windows SharePoint Services mit Microsoft Office“. Microsoft Coporation. <http://download.microsoft.com/download/9/d/6/9d6a5377-3ac0-4f4d-9e60-7543166681c7/office_sharepoint.doc> (25.7.2005).
- [empolis 2003] empolis GmbH. 2003. „The Topic Maps Handbook“. Whitepaper. <http://www.empolis.com/downloads/empolis_TopicMaps_Whitepaper20030206.pdf> (14.3.2006)
- [Hodgson 2005] Hodgson, Ralph. TopQuadrant (Hg). 03/2005. „Semantic Technology and Model-Based Life-Cycle Support for NASA Space Vehicle Engineering.“ <[http://www.topquadrant.com/documents/talks/TQ%20Semantic%20Technology%20for%20ModelBased%20LifeCycle%20Support%20\(v4\).pps](http://www.topquadrant.com/documents/talks/TQ%20Semantic%20Technology%20for%20ModelBased%20LifeCycle%20Support%20(v4).pps)> (30.07.2005)
- [Kirchhof ua 2004] Kirchhof, Anja; Thorsten Gurzki; Henning Hinderer; Joannis Vlachakis. 06/2004. „Was ist ein Portal? Definition und Einsatz von Unternehmensportalen.“ <http://www.ebi.iao.fraunhofer.de/Whitepaper%20Was%20ist%20ein%20Portal_mit%20Logo.pdf> (3.8.2005)
- [Gruber] Gruber, Tom. oJ. „What is an Ontology?“. <<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>> (8.8.2005)
- [Gurzki 2002] Gurzki, Thorsten. 2002. „Portale und E-Business Infrastruktur“. <http://gurzki.de/vortraege/baw2002/Fraunhofer_IAO_Gurzki_Portale.pdf> (3.4.2006)
- [McGuinness 2004] McGuinness, Deborah. 21.4.2004. „The Evolving Semantic Web.“ <http://www.topquadrant.com/conferences/apr21_WCST/materials/Deborah_McGuinness.pdf> (3.2.2006)
- [Microsoft 2003a] Microsoft Corporation. 20.10.2003. „Implementing Rich Collaboration Infrastructure Using Windows SharePoint Services and SharePoint Portal Server 2003.“ <<http://download.microsoft.com/download/7/8/5/785f2f99-5bc8-4c49-a79a-fefe38003f72/SharePointEvaluate.doc>> (2.2.2006)
- [Microsoft 2003b] Microsoft Corporation. 1.11.2003. „Programming the Microsoft SharePoint Object Model, Web Services and Events.“ <http://www.microsoft.com/seminar/shared/asp/view.asp?url=/Seminar/en/20031101SharePoint_T302/manifest.xml&rate=1> (12.2.2006)
- [Microsoft 2003c] Microsoft Corporation. 2003. „Microsoft Windows SharePoint Services 2.0 Architecture“. <<http://www.microsoft.com/>>

- resources/documentation/wss/2/all/adminguide/en-us/stsb02.msp?mfr=true>
(4.4.2006)
- [Microsoft 2004] Microsoft Corporation. 10.12.2004. „Die Architektur von Windows SharePoint Services.“
<<http://www.microsoft.com/germany/msdn/library/servers/windowsserver/DieArchitekturVonWindowsSharePointServices.msp>> (2.2.2006)
- [Naumenko ua 2005] Naumenko, Anton, Sergiy Nikitin, Vagan Terziyan, Andriy Zharko. 04/2005. „Strategic Industrial Alliances in Paper Industry: XML- vs. Ontology-Based Integration Platforms.“ Whitepaper.
<<http://www.cs.jyu.fi/ai/papers/TLO-2005.pdf>> (8.8.2005)
- [Ontopia] Ontopia AS. o.j. „The TAO in Topic Maps – Finding the Way in the Age of Info-glut“. <<http://www.ontopia.net/topicmaps/materials/tao.html>> (14.3.2006)
- [ontoprise 2003] Ullrich, Mike; Andreas Maier; Jürgen Angele. 07/2003. „Taxonomie, Thesaurus, Topic Map, Ontologie – ein Vergleich. v.1.3“ <http://www.ontoprise2003.de/content/e5/e69/e208/TaxonomieThesaurusTopicMapOntologiev13_ger.pdf>
(9.8.2005)
- [SemanticWeb] Berners-Lee, Tim; James Hendler; Ora Lassila. 2001. „The Semantic Web“. <http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21> (8.8.2005)
- [Semtation 2005] Semtation GmbH, Potsdam. 2005. „Tutorial SemTalk Version 2.1 KSA Edition“. <<http://www.semtalk.com/pub/tutsem21ksag.pdf>> (13.3.2006)
- [Sure ua 2002] Sure, York ua. 06/2002. „OntoEdit: Collaborative Ontology Development for the Semantic Web“. In: I. Horrocks; J. Hendler (Hgg). 2002. *The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002. Proceedings*. Berlin: Springer. S. 221-235. <http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2002_iswc_ontoedit.pdf> (26.7.2005)
- [Uni Münster 2003] Universität Münster. Institut für Wirtschaftsinformatik. „Forschungsschwerpunkte 2001-2002: Verteilte Referenzmodellierung.“
<<http://www.uni-muenster.de/Rektorat/Forschungsberichte-2001-2002/fo04fb09.htm>> (8.1.2006)
- [Uni Saarland] Universität des Saarlandes. Fachrichtung Informationswissenschaft. 2.5.2005. „Virtuelles Handbuch Informationswissenschaft. Einführung in die (Informations-) Systemanalyse.“ <http://is.uni-sb.de/studium/handbuch/system/systemanalyse_infsys.html> (10.3.2006)
- [Uschold; Gruninger 1996] Uschold, Mike; Michael Gruninger. 1996. „Ontologies: Principles, Methods and Applications“. In: Knowledge Engineering Review. Vol 11. No 2. S. 93-155. Artikel im PDF-Format unter <<http://citeseer.ist.psu.edu/uschold96ontologie.html>> (14.3.2006)

- [Vom Brocke 2003b] Vom Brocke, Jan. 2003. „Verteilte Referenzmodellierung. Gestaltung multipersoneller Konstruktionsprozesse.“ <<http://www.wi.uni-muenster.de/is/Tagung/Ref2003/vrm-vombrocke.pdf>> (11.6.2005)
- [Weichhardt 2005] Weichhardt, Frauke. 2005. „Redaktionsprozess für verteilte Modellierung“. Potsdam [unveröffentlicht].
- [Wortschatz] Projekt Deutscher Wortschatz. Institut für Informatik der Universität Leipzig. <<http://wortschatz.uni-leipzig.de>>
- [XMLidP] Behme, Henning. Stefan Mintert. 2004. „XML in der Praxis.“ <<http://www.linkwerk.com/pub/xmlidp/2000/buch.html>> (10.09.2005)
- [XPath] World Wide Web Consortium (W3C). 16.11.1999. „XML Path Language (XPath) Version 1.0“. <<http://www.w3.org/TR/xpath>> (13.09.2005)

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Die wörtlich oder inhaltlich den im Literaturverzeichnis aufgeführten Quellen und Hilfsmitteln entnommenen Stellen sind in der Arbeit als Zitat bzw. Paraphrase kenntlich gemacht.

Diese Diplomarbeit ist noch nicht veröffentlicht worden. Sie ist somit weder anderen Interessenten zugänglich gemacht, noch einer anderen Prüfungsbehörde vorgelegt worden.

Brandenburg an der Havel, 6.5.2006

Index

- A
- Anforderung.....
- Methodenbezogen..... 2, 11, 25
 - Modellbezogen..... 2, 11, 13
 - Organisationsbezogen..... 1, 11, 25
 - Technologiebezogen..... 2, 50
- Anwendungspool..... 41, 107, 130
- Application Server..... 43, 50, 74, 76, 130, 133
- Application-Server..... 43, 71, 76
- Assemblierung..... 77, 103ff., 130, 133
- C
- Client... 34, 40, 45ff., 51, 53, 64, 73ff., 78ff., 82, 87, 101, 130ff.
- Clientseitig..... 34, 45f., 51, 73f., 78, 82, 87, 130f.
- Codezugriffssicherheit.....
- Personifizierung..... 106f.
- D
- Diagramm..... 14ff., 19ff., 32, 53, 67ff., 89, 94ff., 125ff., 130, 132
- Dokumentbibliothek..... 40f., 43, 47, 51, 64ff., 76f., 79ff., 83ff., 95, 97, 101ff., 130
- E
- Ereignisroutine... iii, 72, 76ff., 87, 89f., 103ff., 114, 130
- Externe
- Verfeinerung..... 65
- Externe Referenz... iii, 14, 16ff., 21, 24, 66ff., 82ff., 88f., 96f., 102f.
- Externe Verfeinerung..... iii, 14ff., 19f., 67, 69, 71, 102f., 112, 122
- F
- Freigegebener Arbeitsbereich..... iii, 46, 100, 131
- G
- Gemeinsamer Arbeitsbereich..... 34, 130f.
- Dokumentbibliothek..... iii, 40f., 43, 47, 51, 64ff., 76f., 79ff., 83ff., 95, 97, 101ff., 130
 - Liste... iii, 36, 39ff., 46, 51, 63ff., 74, 76ff., 89f., 94, 98, 102, 105f., 130f.
- K
- Konsistenz..... 13, 15ff., 20, 25, 27, 29, 32, 54f., 59, 62, 65f., 71, 82, 84, 93f., 112, 131
- M
- Microsoft Office System Integration.i, 44f., 51, 87, 100, 132
- Modell.....
- Diagrammi, iii, 14ff., 19ff., 32, 53, 67ff., 89, 94ff., 125ff., 130, 132
- Konsistenz..... iii, 13, 15ff., 20, 25, 27, 29, 32, 54f., 59, 62, 65f., 71, 82, 84, 93f., 112, 131
- Prozesselement..... 15f., 19f., 25, 87
- Modellierung.....
- Iv. Verteilte Modellierung..... 122
 - Modellierungsmethode 7, 12f., 15, 59, 61, 87, 100, 115
 - Modellierungswerkzeug. 7, 11, 13, 26, 31f., 61, 82, 114f.
 - Verteilte Modelle..... 10, 13, 86
 - Verteilte Modellierung. iff., 8, 10ff., 26, 28f., 50f., 53, 59ff., 64, 66, 70f., 100, 108, 110, 112ff., 121ff., 139
 - Verteilte Modellierung 115
- O
- Objekt.i, iii, 8, 13ff., 43, 52f., 55, 58f., 62, 65ff., 71, 73, 82ff., 87f., 90ff., 102f., 106, 112, 122, 131ff., 135
- Objektorientierung.....
- Klasse.. iii, 13, 20f., 23f., 52f., 58f., 72ff., 79, 89ff., 98, 104, 106f., 131f.
 - Objektorientiert..... 13, 52f., 99, 133
 - Objektorientierung..... 13, 52f., 99, 133
- Objekts..... 25
- Ontologie... 2, 8ff., 12, 16, 26ff., 54f., 58, 130, 132, 138
- Ontologiebasiert..... 2
- Ontologien..... 8
- R
- Referenzmodell... 1, 22ff., 62, 65, 68, 71, 97, 102f., 112, 122, 130ff., 135f., 138f.
- Referenzmodellierung..... 1, 23f., 135f., 138f.
- Referenzobjekt. 14, 16ff., 24f., 55, 59, 62, 65ff., 71, 83ff., 87, 92, 96f., 103, 112
- S
- Semantic Web..... 1, 9, 17f., 32, 137f.
- SemTalk. 1ff., 11ff., 25, 28f., 50ff., 54ff., 64ff., 70f., 82ff., 87ff., 95, 100ff., 105ff., 110ff., 121f., 132, 136, 138
- SemTalk und ist größtenteils in der Praxis entstanden..... 29
- Semtalk-Webdienst..... iii, 87ff., 95, 105ff.
- Server.....
- Application Server..... 43, 50, 74, 76, 130, 133
 - Application-Server..... 43, 71, 76
 - Virtuelle Server..... 41, 86, 133
- U

UML.....	i, 52f., 57, 89, 94, 125, 133, 135	Anwendungspool.....	41, 107, 130
UML.....		Virtueller Server.....	104, 133
Klassendiagramm.....	iii, 94	W	
Sequenzdiagramm.....	i, 53, 89, 94ff., 125ff.	Webdienst.....	i, iii, 39, 72, 74ff., 82, 85, 87ff., 95f., 104ff., 114, 133
V		Webpart.....	iii, 38ff., 51, 63f., 72, 77ff., 112, 133
Virtuelle Server.....	41, 86, 133		