Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Entwicklung eines Nagios-Plugins zur Überwachung und Auswertung von Funktionen und Fehlern in Content-Managment-Systemen

BACHELORARBEIT

für die Prüfung zum
Bachelor of Engineering

des Studienganges

Informationstechnik

an der Dualen Hochschule Karlsruhe

von

Andreas Paul

Bearbeitungszeitraum:	25.05.2009 – 23.08.2009
Matrikelnummer:	108467
Kurs:	TIT06GR
Praxissemester:	6
Ausbildungsfirma:	Forschungszentrum Karlsruhe GmbH (FZK) Steinbuch Centre for Computing Hermann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen
Betrieblicher Betreuer:	Dr. Doris Wochele
Prüfer der DHBW Karlsruhe:	Dipl-Ing. Holger Raff (BA)

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.
Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.
Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Karlsruhe, den 17. August 2009

.....

Ort, Datum (Andreas Paul)

Inhalt

1 Einleitung 1

2 Aufgabenstellung 3

3 Grundlagen 5

3.1 Überwachungssysteme 5

3.1.1 Ressourcenbelastung 6

3.1.2 Netzwerkstruktur und Abhängigkeiten 6

3.1.3 Sicherheitsaspekte 8

3.2 Dokumenten-Management-Systeme 9

3.2.1 Eingabe 13

3.2.2 Verwaltung und Archivierung 15

3.2.3 Ausgabe 16

3.3 Content-Management-Systeme 16

3.4 Service-Orientierte Architektur 19

3.5 Web-Services-Architektur 21

4 Nagios 25

4.1 Allgemein 25

4.2 Aufbau / Architektur 26

4.3 Überprüfungsmethoden 32

4.3.1 Aktive Checks 32

4.3.2 Passive Checks 32

5 Oracle UCM 38

5.1 Allgemein 38

5.2 Aufbau / Architektur 38

5.3 Konkrete Verwendung / Einsatzgebiet 40

6 Überwachungselemente 42

6.1 Statusabfragen 42

6.2 Überwachung der Funktionalität 43

6.3 Auswerten von Logdateien 44
6.4 Benutzersimulation 45
7 Umsetzung 48
7.1 Aufbau der Testumgebung 48
7.1.1 Aufsetzen eines Nagios-Test-Systems 48
7.1.2 Bilddatenbank als virtuelle Maschine 48
7.2 Übersicht Nagios-Agenten 48
7.2.1 Unix-Agenten 49
7.2.2 Windows-Agenten 51
7.2.3 Auswahl und Konfiguration des Nagios-Agenten 53
7.3 Umsetzung der Systemüberwachung 57
7.4 Umsetzung der Funktionlitätstest 58
7.5 Auswertung der Logdateien 60
7.6 Benutzersimulation 61
8 Ergebnis 69
9 Zusammenfassung und Ausblick 71

1 Einleitung

Mit dem Zusammenschluss des Forschungszentrum Karlsruhe und der Universität Karlsruhe (TH) zum Karlsruhe Insitute of Technology (KIT) ist eine Einrichtung mit 8000 Wissenschaftler und Mitarbeiter, 18000 Studierende und circa 300 externen Mitarbeitern und Gästen entstanden.

Die IT-Infrastruktur für den organisatorischen und wissenschaftlichen Betrieb liegt in der Verantwortung des Steinbuch Center für Computing (SCC), das aus der Verschmelzung des Rechenzentrums der Universität und dem Institut für Wissenschaftliches Rechnen (IWR) hervorgegangen ist.

Für alle Schichten der IT-Infrastruktur und alle angebotenen Dienstleistungen muss der Betrieb durch das Rechenzentrum überwacht werden. Die Überwachung des Dokumenten-Management-System (DMS), eines wichtigen zentralen Dienstes, war Ziel dieser Arbeit.

Unter den Aufgaben eines Dokumenten-Management-System fallen hauptsächlich die zentrale Speicherung, Bearbeitung und Verwaltung von Dokumenten. Dabei können diese Dokumente Dateien in unterschiedlicher Form sein wie Microsoft Word Dateien, Excel Tabellen, Dateien im Portable Document Format (PDF) oder auch Bilder in vielen weiteren Formaten. ~~Die wichtigsten Funktionen~~

Aufgrund der Vielzahl an angebotenen Dienstleistungen ist es schwierig herauszufinden, ob die angebotenen Dienstleistungen noch fehlerfrei arbeiten oder aus welchem Grund die Benutzer nicht mehr auf einen Dienst zugreifen können. Für diesen Zweck wurden Überwachungssysteme entwickelt die den Status der verschiedenen Komponenten und den davon abhängigen Diensten überwachen und bei Veränderungen die Verantwortlichen darüber informiert. Für einen möglichst störungsfreien Betrieb ist es notwendig, dass die Ergebnisse der Überwachung in periodischen Zeitabständen erneuert werden, damit ein auftretendes Problem schnellstmöglich erkannt und behoben werden

kann. Das Überwachungssystem soll so implementiert werden, dass Fehler erkannt werden, bevor die Nutzung der angebotenen Dienstleistungen davon beeinträchtigt werden. Dabei muss die zusätzliche Belastung der Netzwerkes und der überwachten Objekte durch die Überwachung eingeplant, die verwendete Netzwerkstruktur und die dadurch entstehende Abhängigkeit (von Netzwerkknoten) beachtet und sicherheitstechnische Aspekte einer automatischen Überwachung bedacht werden.

Im Laufe dieser Arbeit soll eine Überwachung eines Dokumenten-Management-Systems unter Berücksichtigung der Funktions- und Arbeitsweise des eingesetzten Dokumenten-Management-Systems durch eine Open Source Überwachungsanwendung realisiert werden.

2 Aufgabenstellung

Um den Mitarbeitern des Forschungszentrums Karlsruhe eine möglichst ausfallsichere Plattform für die zentrale Speicherung, Bearbeitung und Verwaltung von Dokumenten anzubieten soll eine Überwachung realisiert werden, die nicht nur die Anwendung, sondern auch den darunterliegenden Server auf seine Systemressourcen überwacht. Dabei müssen diese Elemente gefunden werden, mit deren Überprüfung der eindeutige Zustand der Anwendung festgestellt und der störungsfreie Betrieb sichergestellt werden kann.

Im Forschungszentrum Karlsruhe wird für die Verwaltung von Webseiten, Dokumenten und Bilder das Dokumenten-Management-System Oracle UCM der Firma Oracle eingesetzt. Daher muss sich für die Ermittlung der zu überwachenden Objekte mit dem Aufbau und der spezifischen Funktions- und Arbeitsweise des verwendeten Dokumenten-Management-Systems auseinandergesetzt werden.

Als Überwachungssoftware wird im Forschungszentrum Karlsruhe das Open Source-Projekt Nagios eingesetzt. Damit der fehlerfreie Betrieb von Oracle UCM als Dienst durch die Überwachung der ermittelten Überwachungselemente eindeutig festgestellt werden kann, muss sich mit dem Aufbau, der internen Funktionsweise und den verschiedenen Methoden bezüglich der Ermittlung der Statusinformationen untersucht werden. Dabei soll eine Übersicht über die unterschiedlichen Überwachungsmethoden von Nagios erstellt werden und unter Berücksichtigung des späteren Einsatzes bewertet werden. Hierbei sind für die spätere Umsetzung beispielsweise die verschlüsselte Datenübertragung zwischen Überwachungs- und Anwendungsserver ein Kriterium. Mit der durch diese Bewertung ausgewählte Methode soll die Überwachung auf verschiedenen Ebenen realisiert werden.

Die Kategorisierung der Überwachungselemente ergibt sich aus der Gewichtung der einzelnen Elemente. Essentielle Merkmale / Informationen wie die

simple Erreichbarkeit über das Netzwerk bilden die Grundlage der darüber liegende Überwachungsobjekte wie der Zustand eines Prozesses. Dabei soll die Anwendung auch reaktiv durch eine Auswertung von Logdateien auf Fehler überwacht werden.

Zum eindeutigen Erkennen von Fehlern, die während der Benutzung durch die Anwender auftreten, sollen die typischen Aktionen der Benutzer simuliert werden. Für die Realisierung dieser Benutzersimulation muss die Anwendung über eine Schnittstelle verfügen, die sich durch ein Programm über das Netzwerk ansprechen lässt. Dieses Programm soll die Benutzeraktionen automatisch/selbständig durchführen und der Überwachungssoftware Nagios die Ergebnisse der einzelnen Schritte übermitteln, damit der Fehlerzustand (möglichst) sofort erkannt und gleichzeitig seine Ursache eingegrenzt werden kann.

Dabei müssen bei der Programmentwicklung mögliche Konsequenzen aufgrund verschiedener Szenarien bedacht werden. Sollte die Anwendung bereits durch eine Vielzahl von Benutzern stark belastet sein, wird dadurch auch der Ablauf der Benutzersimulation verzögert In diesem Fall soll die Überwachungssoftware bzw. Benutzersimulation keine falsche Informationen melden.

Durch die Benutzersimulation darf die Nutzung der Anwendung durch die eigentlichen Benutzer nicht beeinträchtigt werden. Da die Ausführung der Benutzersimulation durch Nagios in kurzen Zeitabständen periodisch aufgerufen wird, müssen auch langfristige Auswirkungen wie das Überlaufen der Datenbank der Anwendung oder die Überfüllung des Festplattenspeichers des Anwendungsservers bedacht werden.

Da als Entwicklungsumgebung ein eigener Nagios-Server eingesetzt werden soll, muss die entwickelte Lösung auf den bereits vorhanden Nagios-Server exportierbar sein.

3 Grundlagen

In diesem Kapitel werden die Grundlagen von Überwachungssystemen und Dokumenten-Management-Systemen erläutert. Insbesondere wird auf Service-Orientierte Architektur (SOA) und Web-Services für die spätere Umsetzung eingegangen.

3.1 Überwachungssysteme

Überwachungssysteme wurden für den Zweck entwickelt den Status von verschiedenen Objekten meist über das Netwerk zu überwachen und im Falle einer Statusänderung diese Information an die zugewiesenen Kontaktpersonen weiterleitet.

Bei diesen Objekten kann es sich um viele verschiedene Komponenten handeln. Generell unterscheidet man zwischen der Überwachung ermöglichten zu Grunde liegenden Hardware den so genannten Hosts und den auf diesen Hardwarekomponenten aufsitzenden Diensten auch Services genannt.
Unter Hosts fallen nicht nur Server bzw. Computer, sondern auch Switches, Router oder auch explizite / dedizierte / (nur für den/einen Zweck der Überwachung halt) Überwachungshardware wie Sensoren für Temperatur, Luftfeuchtigkeit oder Rauchmelder. Die Services dieser Hosts weichen je nach Art der Hosts stark voneinander ab. Auf einem Server kann als Service ein Webserver im Betrieb sein, dessen Funktionalität sich simpel über einen Aufruf einer Webseite überprüfen lässt. Bei einem Switch können beispielsweise als Service die Übertragungsrate, der Paketverlust oder der Portzustand überwacht werden.
Sehr wichtig ist bei einem Überwachungssystem die Gewichtung der erhaltenen Überwachungsinformationen.

Vor der Einführung eines Überwachungssystems muss sich mit den folgenden Punkten auseinandergesetzt werden.

3.1.1 Ressourcenbelastung

Die Einführung einer Überwachungssoftware bringt bei größeren Serverlandschaften eine nicht zu verachtende Netzwerk- und Prozessorbelastung mit sich. Dabei unterscheidet Josephsen die anfallende Belastung in zwei unterschiedliche Arten der Überwachung[1]:

Lokale / Zentrale Bearbeitung Die Durchführung der Überprüfungen findet durch einen zentralen Überwachungsserver statt, der die Informationen über die einzelnen Hosts und Services über das Netzwerk abfragt. Diese Methode ist in der Regel vorzuziehen, da hierbei die zu überwachenden Geräte weniger belastet werden und die Konfiguration der einzelnen Kontrollschritte zentral möglich / realisierbar ist.

Entfernte / Ausgelagerte Bearbeitung Bei einer sehr hohen Anzahl von zu überwachenden Objekten ist eine zentralisierte Ausführung nicht mehr von einem einzelnen Server tragbar. In diesem Fall ist das Überwachungssystem darauf angewiesen, dass die einzelnen Hosts die kontrollierenden Überprüfungen selbständig durchführen und deren Ergebnisse an den Überwachungsserver weiterzuleiten.

3.1.2 Netzwerkstruktur und Abhängigkeiten

Die Überwachung von Hosts und Services über das Netzwerk erzeugt normalerweise immer zusätzlichen IP-Traffic. Das bedeutet, dass jede Überquerung weiterer Netzwerkknoten, die zwischen dem Überwachungsserver und den zu überwachenden Geräten liegen, eine weitere Belastung für das Netzwerk bedeutet, sowie eine Abhängigkeit zwischen Host und Server einführt.

[1] Quelle: [Jose07] S. 4

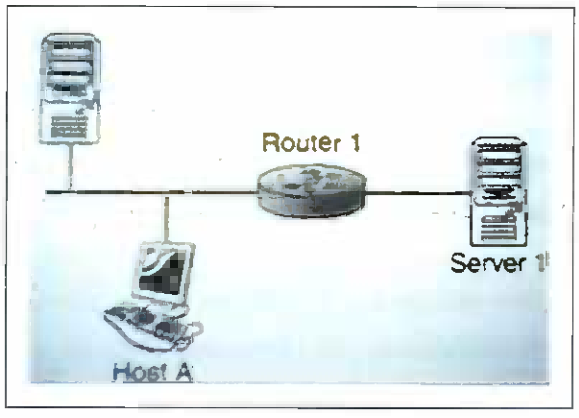


Abbildung 1: Zusätzliche Netzwerkabhängigkeit und Netzwerkbelastung[2]

In der Abbildung 1 erzeugt der Router 1 die zuvor beschriebene zusätzliche Netzwerkabhängigkeit und Netzwerkbelastung, da der Server 1 bei einem Ausfall des Routers nicht mehr durch den Überwachungsserver erreichbar ist und jede Überprüfung, die vom Überwachungsserver gesendet wird den Router mit dem Routing der Pakete belastet.
Deshalb gilt es laut [Jose07] S. 5 folgende zwei Punkte beim Erstellen eines Überwachungssystems zu beachten:

Überwachungsredundanzen vermeiden Redundante Überwachung entsteht dadurch, dass der gleiche Service durch zwei Arten ~~mit~~ in unterschiedlichen Tiefen / ~~Tiefgang~~ geprüft wird. Ein einfaches Beispiel ist die Überwachung eines Webservers auf dem Standardport 80. Eine Überwachungsmethode ist es diesen Port abzufragen und die entsprechende Rückantwort des Servers auszuwerten. Soll die auf dem Webserver laufende Webseite überwacht werden, kann die jeweilige Webseite über die Adresse nach einem bestimmten Inhalt untersucht werden.
In beiden Fällen wird getestet, ob der Webserver über das Netzwerk ansprechbar ist, jedoch sagt der zweite Test zusätzlich noch aus, dass die Webseite korrekt angezeigt wird, somit wäre der erste Test überflüssig. Jedoch muss

[2]Quelle: [Jose07] S. 5

zuvor abgewogen werden, ob eine redundante Überwachung nicht sogar hilfreich bei der Ermittlung der Fehlerursache ist. Wenn im oberen Beispiel der Inhalt der überwachten Webseite verändert wird, ist dies nur aus dem zweiten Test ersichtlich.

Minimale Netzwerkbelastung Um bereits stark belastete Netzwerkpunkte zu entlasten, bietet es sich an, die Frequenz mit der die Test über das Netzwerk gesendet werden zu verringern. Die Aufstellung des Überwachungsservers ist dadurch gerade bei größeren Serverlandschaften sehr wichtig, da durch eine effiziente Platzierung womögliche Flaschenhälse / Engstellen in Form von veralteten Switches oder ähnlichem vermieden werden können.

3.1.3 Sicherheitsaspekte

Um erweiterte Statusinformationen über einen Prozess oder über die Arbeitsspeicherauslastung auszulesen ist (meistens) zusätzliche Software auf den Hosts nötig. Diese Software benötigt einen zusätzlichen geöffneten Port auf dem zu überwachendem Rechner, die einen neuen Angriffspunkt für Angreifer darstellen kann. Außerdem erhält der Überwachungsserver Ausführungsrechte auf dem Client, so dass eine weitere potentielle Sicherheitslücke in einem (vermeintlich) zuvor sicherem System entsteht. Jeder, der die Kontrolle über den Überwachungsserver besitzt oder sich als solcher ausgibt, kontrolliert somit gleichzeitig alle anderen überwachten Hosts.

Um dies zu verhindern gibt es verschiedene Ansätze. Als ersten Ansatz sollte der Port durch den der Überwachungsserver mit dem Host kommuniziert vom Standardwert abweichen, damit nicht sofort erkennbar ist, dass sich eine (womöglich) angreifbare Überwachungssoftware auf dem Rechner befindet. Damit die über diesen Port versendeten Informationen nicht für Dritte zugänglich sind, bietet es sich an die auszutauschende Informationen mit einem Algorithmus zu verschlüsseln. Durch den Einsatz eines Verschlüsse-

lungsalgorithmus werden die Informationen nicht mehr im Klartext ausgetauscht, sondern Da die Möglichkeit einer Verschlüsselung der Datenübertragung nicht von jeder Überwachungssoftware angeboten wird, gilt diese Option als Auswahlkriterium in der späteren Umsetzung bzw. im produktivem Betrieb. (Verweis auf Windows Agenten Übersicht?) hö kommt noch . . . .
Des weiteren sollte die Erlaubnis der Abfrage der Überwachungsinformationen anhand der IP-Adresse eingeschränkt werden, so dass der Client nur Anfragen des Überwachungsservers akzeptiert. Durch diese Einschränkung kann vermieden werden, dass sensible Informationen aus den Antworten an unberechtigte Dritte übermittelt werden oder ein Denial of Service-Angriff (DoS) durch eine übermäßig hohe Anzahl an Anfragen an den Client gesendet wird, um eine Überlastung des Servers zu erreichen und diesen somit arbeitsunfähig zu machen.

3.2 Dokumenten-Management-Systeme

Um ein Dokumenten-Management-System (DMS) zu erläutern muss sich zuerst mit dem Begriff des **„Dokuments“** auseinander gesetzt werden. In [DMS08] S. 2 wird ein Dokument durch folgende Punkte definiert:

- Ein Dokument fasst inhaltlich zusammengehörende Informationen strukturiert zusammen, die nicht ohne erheblichen Bedeutungsverlust weiter unterteilt werden können.
- Die Gesamtheit der Information ist für einen gewissen Zeitraum zu erhalten.
- Ein Dokument ist als Einheit ablegbar (speicherbar) und/oder versendbar und/oder wahrnehmbar (sehen, hören, fühlen).
- Das Dokument ist eigentlich der Träger, der die Informationen speichert, egal ob das Dokument ein Stück Paper, eine Datei auf einem

Rechner, ein Videoband oder eine Tontafel etc. ist. Dies bedeutet auch, dass es keine Bindung an Papier oder ein geschriebenes Wort gibt.

Des weiteren gibt es eine Differenzierung in zwei Definitionen:

„Als **Dokument im konventionellen Sinne** werden Dokumente bezeichnet, die als körperliches Dokumente (z. B. Papier) vorliegen, ursprünglich als körperliches Dokument vorlagen oder für die Publizierung auf einem körperlichen Medium vorgesehen sind.

Die Begrifflichkeit des **Dokuments im weiteren Sinne** erweitert den Begriff des Dokuments um semantisch zusammengehörende Informationsbestände, die für die Publikation in nicht-körperlichen Medien, z.B. Webseiten, Radio, Fernsehen o. ä. vorgesehen sind. Derartige Dokumente werden oft dynamisch gestaltet und zusammengestellt."

[DMS08] S. 2

Dabei müssen auch Daten und Dokumente voneinander abgegrenzt werden. In [DMS08] S. 33 werden Daten im Allgemeinen als eher stark strukturierte Informationen gesehen, wobei Dokumente zumeist aus unstrukturierte bis zu schwach strukturierte Informationen bestehen. Eine eindeutige Klassifizierung eines vorhandenen Dokumentes lässt ist jedoch nicht immer möglich, da sich oft Mischungen beider Klassen finden (lassen). Ohne die dazugehörigen Metadaten besteht ein (graphisches) Bild aus unstrukturierten Informationen, daher auch NCI-Dokument für None-Coded Information genannt.
Die Einordnung, wann ein Dokument strukturierte oder unstrukturierte Informationen enthält, lässt an folgenden Beispielen verdeutlichen. Bei einem Bild oder Foto lassen sich die enthaltenen Informationen nicht durch Computer bestimmen. Beispielsweise, ob sich eine Person auf dem Bild befindet oder

↑ Das Beispiel ist schlecht. Es gibt Software die genau das leistet. Eher „fröhliches... traurigos...

wann und wo das Foto erstellt wurde. Daher ist ein Bild, solange keine Metadaten darüber bekannt sind, ein eindeutiges Beispiel für NCI-Dokumente mit unstrukturierten Informationen. Im Gegensatz dazu lassen sich die Werte einer Tabelle oder eines Datensatzes durch die Spaltennamen eindeutig bestimmen und durch den Computer auslesen. Solche Daten mit strukturierten Informationen werden daher auch als Dokumententyp mit Coded Information (CI) bezeichnet.

Der Anteil von strukturierten Informationen in einem Dokument nimmt von Bildern über Text zu Tabellen zu, da hier die Dokumente vollautomatisch auswertbar sind, siehe hierzu Abbildung 2.

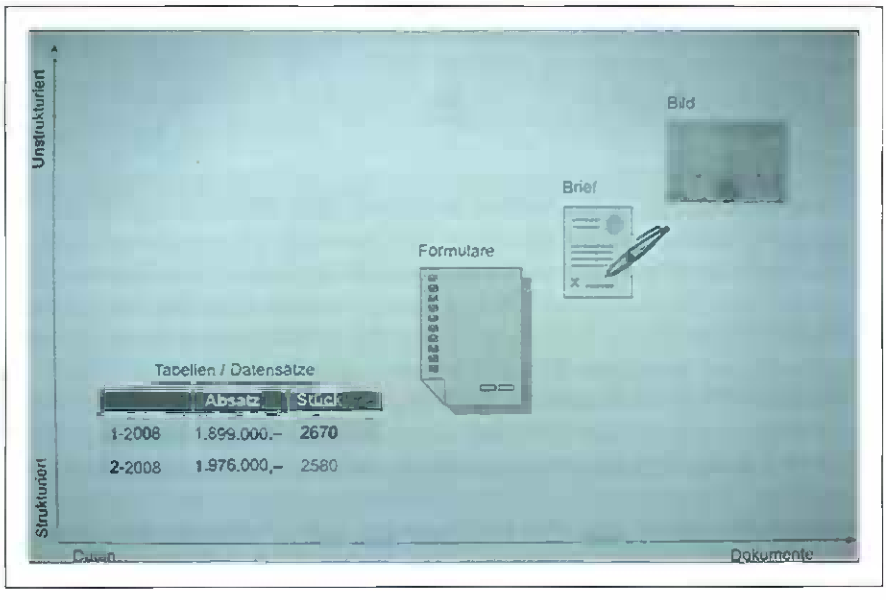


Abbildung 2: Anteil an strukturierten Informationen[3]

Unter **Dokumenten-Management** werden primär die Verwaltungsfunktionen Erfassung, Bearbeitung, Verwaltung und Speicherung von Dokumenten verstanden. [DMS08] S. 344.

Darunter fallen laut [DMS08] S. 3 folgende Punkte:

[3]Quelle: [DMS08] S. 33

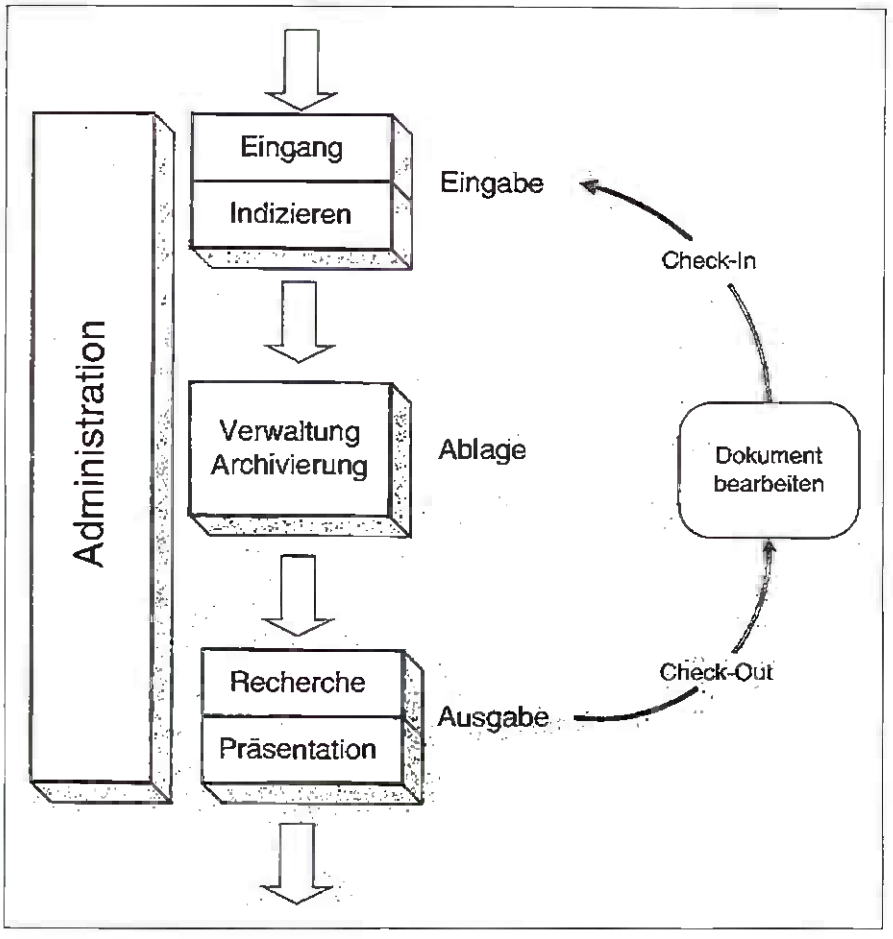
- Kennzeichnung und Beschreibung von Dokumenten (auch Metadaten des Dokuments genannt)
- Fortschreibung, Versionierung und Historienverwaltung von Dokumenten
- Ablage und Archivierung von Dokumenten
- Verteilung und Umlauf von Dokumenten
- Suche nach Dokumenten bzw. Dokumenteninhalten
- Schutz der Dokumente vor Verfälschung, Missbrauch und Vernichtung
- Langfristiger Zugriff auf die Dokumente und Lesbarkeit der Dokumente
- Lebenslauf und Vernichtung von Dokumenten
- Regelung von Verantwortlichkeiten für Inhalt und Verwaltung von Dokumenten

Der Begriff **„Dokumenten-Management-System"** muss auch in zwei verschiedene Sichtweisen differenziert werden:

> „Bei **Dokumenten-Management-Systemen im engeren Sinne** geht es um die Logik der Verwaltung von Dokumenten, deren Status, Struktur, Lebenszyklus und Inhalt. Dokumente werden beschrieben, klassifiziert und in einer bestimmten logischen Struktur eingeordnet, damit sie einfach wieder gefunden werden können. Dokumente entstehen, werden verändert und (irgendwann) vernichtet.
>
> Den **Dokumenten-Management-Systemen im weiteren Sinne** ordnet man auch noch weitere Funktionalitäten zu, wie z. B. Schrifterkennung, automatische Indizierung, [...], Publizierung. Hier lassen sich die Grenzen nicht mehr genau bestimmten!"

[DMS08] S. 5

Die Grundstruktur eines Dokumenten-Management-Systemes kann man dadurch grob in folgender Abbildung zusammenfassen:



Bischen groß

Abbildung 3: Aufgabenbereiche eines Dokumenten-Management-Systems[4]

Dabei wird ein DMS-System in drei verschiedene Teilbereiche aufgegliedert:

3.2.1 Eingabe

Unabhängig des Ursprungs oder der Art des Dokumentes besitzt der Funktionsbereich Eingabe die Aufgabe diese Dokumente dem Dokumenten-Management-System zuzuführen.

Laut [DMS08] S. 40fff fallen in diesen Bereich zwei Funktionen:

[4]Quelle: [DMS08] S. 38

Dokumenteneingang Hier wird ~~die~~ Zuspielung der Dokumente in das DMS-System durch verschiedene Methoden behandelt / realisiert. Als mögliche Eingabe von Dokumenten kann sowohl das Einscannen von Textdokumente oder Bilder als auch der elektronische Eingang von Dokumenten durch E-Mail oder externen Anwendungen fungieren.

Auch hier gilt zu unterscheiden, dass durch den Einscannvorgang erstellte Dokumente als NCI-Dokument abgelegt werden und bereits digitalisierte Dokumente sich zur Umwandlung zu CI-Dokumenten anbieten. Sobald der Inhalt von eingescannten Dokumenten zur weiteren Verarbeitung ausgelesen bzw. ausgewertet werden soll, müssen die Dokumente in ein CI-Format transformiert werden. Dies wird häufig durch eine OCR-Software realisiert, die beispielsweise das Bild eines eingescannten Briefes in (bearbeitbaren) Text umwandelt.

Bereits im CI-Format vorliegende Dokumente müssen nicht transformiert werden, jedoch werden die Dokumente oft in anderen Formaten zusätzlich abgespeichert. Ein Beispiel ist die Umwandlung eines Microsoft Word-Dokumentes in ein PDF-Dokument oder von ~~unterschiedlichen Bildformaten in ein einheitliches~~ Format.

Indizierung Bei der Indizierung werden Dokumente zur eindeutigen Identifikation mit Attributen versehen. Diese Attribute werden teilweise automatisch durch das DMS-System anhand einer hoch zählenden Identifikationsnummer oder manuell durch den Benutzer beim Einstellen des Dokumentes hinzugefügt. Solche Attribute werden auch als Metadaten des Dokumentes bezeichnet und meist als zusätzliche Suchkriterien angeboten.

Dabei werden in [DMS08] S. 44 zwei verschiedene Methoden zur automatischen Klassifizierung genannt. Beim wissensbasiertem Ansatz wird mittels umfangreichem Wissen über das Umfeld der Dokumente und dadurch abgeleitete Regeln dem System ermöglicht diese Dokumente automatisch einzu-

ordnen und zu indizieren. Eine weitere Möglichkeit eröffnet sich durch das Verwenden von neuronalen Netzen. Hierbei wird durch die Vorarbeit eines Menschen Beispiele geschaffen anhand welcher sich das System selbstständig (Auswahl)Kriterien erzeugt. Je mehr korrekte Beispiele vorgegeben werden, desto besser und zuverlässiger arbeitet die automatische Klassifizierung.

3.2.2 Verwaltung und Archivierung

Bei der **Verwaltung** werden die Probleme beim *Check-in* (Einspielen des Dokumentes), Bearbeitung und *Check-out* (Signalisieren der Weiterbearbeitung) behandelt, siehe auch Abbildung [DMS08] S. 38 ???. Wie auch bei einer Datenbank müssen Dokumente, die gerade bearbeitet werden, für andere Benutzer für Änderungen gesperrt werden, damit keine Inkonsistenzen auftreten können. Nach einer Bearbeitung und dem Check-in des abgeänderten Dokumentes muss die Versionsverwaltung des DMS-Systems beide Versionen beibehalten und (dabei) die ursprüngliche Version als veraltet und die neue Version als solche kennzeichnen. Zusätzlich muss die Wiederherstellung einer älteren Revision als aktuelles Dokument unterstützt werden.
Die **Archivierung** befasst sich mit der Sicherung und Wiederherstellung von Dokumenten und deren Metadaten. Im Zusammenhang mit DMS-Systemen springt man auch von einer revisionssicheren Archivierung. Dabei müssen laut [DMS08] S. 288 unter anderem bestimmte Punkte beachtet / eingehalten werden:

- Jedes Dokument muss unveränderbar archiviert werden.
- Es darf kein Dokument auf dem Weg ins Archiv oder im Archiv selbst verloren gehen.
- Kein Dokument darf während seiner vorgesehenen Lebenszeit zerstört werden können.

- Jedes Dokument muss in genau der gleichen Form, wie es erfasst wurde, wieder angezeigt und gedruckt werden können.

3.2.3 Ausgabe

Wie die Eingabe besteht die Ausgabe aus zwei Funktionen:

Recherche Die Recherche ist die Suche nach einem Dokument entweder durch eine strukturierte Suche anhand von zuvor eingetragenen Attributen (Autor, Erstellungsdatum, Speichergröße usw.) oder durch eine Volltextsuche.
Die **strukturierte Suche** ist nur bei einer qualitativ hochwertigen Indizierung effizient, bietet dafür auch mit guter zeitlichen Performanz die besten / genauesten Ergebnisse, sofern die Indizierung entsprechend aufgebaut / eingehalten wurde.
Die **Volltextsuche** besteht aus einer ordinären Suche durch den Inhalt der Dokumente nach den eingegebenen Suchbegriffen. Daher ist die Qualität der Suchergebnisse unabhängig von der Qualität der Indizierung. Jedoch können nur CI-Dokumente, deren Informationen auch durch den Computer auslesbar und interpretierbar sind, durchsucht werden. NCI-Dokumente wie Bilder oder Videos können ohne Metadaten durch die Volltextsuche nicht gefunden werden.

Reproduktion In diesem Teilbereich können die gespeicherten Dokumente wieder vom Benutzer abgerufen werden. Dies ist durch eine einfach Anzeige im Webbrowser, eine Weiterleitung per E-Mail oder eine Sendung als Druckauftrag möglich.

3.3 Content-Management-Systeme

Bei einem Content-Management-System (CMS) steht nicht mehr das eigentliche Dokument im Vordergrund, sondern vielmehr der enthaltene Informati-

onsgehalt des Dokuments. Der Unterschied zwischen einem DMS und einem CMS besteht laut [DMS08] S. 114 im/in Folgenden/m:

> „Ein DMS hat als kleinstes Objekt der Betrachtung eines einzelnen Dokument. [...] Content-Management ist auf logische Informationseinheiten ausgerichtet. Es ist z.B. das Ziel des Content-Managements, Inhalte, die auf mehrere Quellen verteilt sind, neue zusammenzustellen und daraus z.B. ein neues Dokument zu generieren."
>
> [DMS08] S. 114f

Die folgende Abbildung soll den (charakteristischen) Unterschied zwischen CMS-Systemen und DMS-Systemen verdeutlichen.

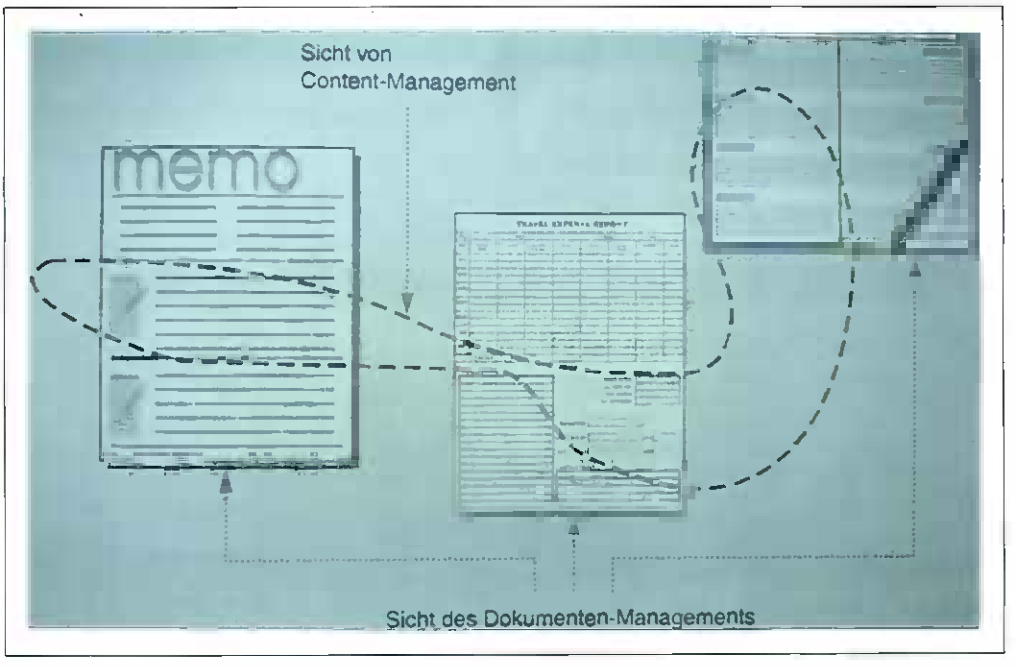


Abbildung 4: Sichtweise CMS gegenüber DMS[5]

Wie zuvor beschrieben ist die Sichtweise eines DMS nur auf die einzelnen Dokumente beschränkt, während ein CMS einzelne Elemente / Informationen aus den Dokumenten extrahieren und ggf. zu einem neuen Dokument

[5]Quelle: [DMS08] S. 115

verschmelzen kann. Die Sichtweise des CMS wird durch das gestrichelte Polygon dargestellt, welches hier dokumentenübergreifend abgebildet ist.
tiefer Sinn ?
Der (theoretische/beabsichtigte) Zweck, weshalb ein CMS-System eingesetzt wird, ist laut Oracle folgendermaßen definiert:

> „The key to a successful content management implementation is unlocking the value of content by making it as easy as possible for it to be consumed. This means that any piece of content must be available to any consumer, no matter what their method of access."
>
> [UCM07] S. 12

Ein CMS soll die Informationen jedes/jedwedem (Inhalts) extrahieren/aufnehmen und jedes Einzelteil / Element dieser Information den Benutzern zugänglich machen, unabhängig von der Art des Zugriffs. Dieses Konzept soll in Abbildung 5 verdeutlicht werden.

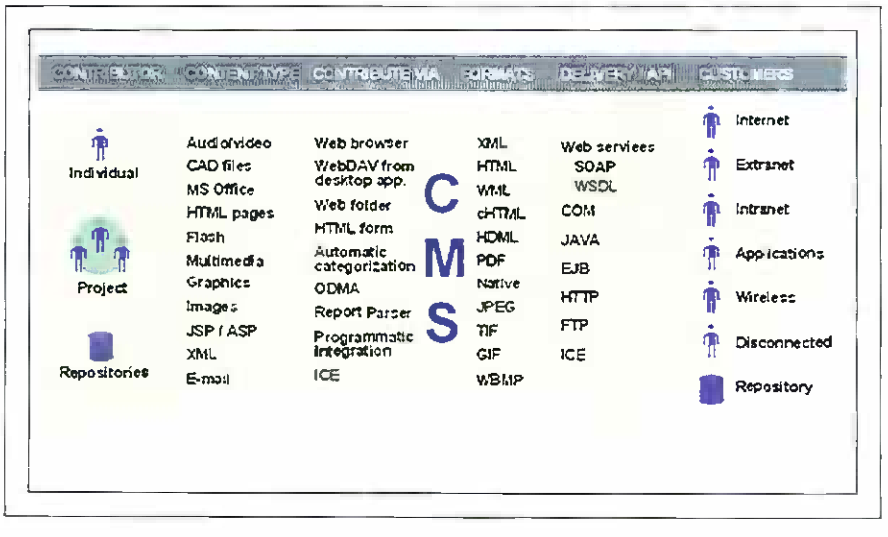


Abbildung 5: „any-to-any" Content-Management Konzept[6]

[6]Quelle: [UCM07] S. 12

Das CMS steht hier in der Mitte der Abbildung als Medium zwischen den verschiedenen Inhalten, eingestellt von den *Contributors* (links), und den Anwendern, die auf transformierte Versionen der Inhalte durch unterschiedliche Arten zugreifen (rechts).

3.4 Service-Orientierte Architektur

Eine eindeutige und einheitliche Definition einer Service-Orientierter Architektur (SOA) existiert nicht. Einen Versuch einer Definition wird in [SOA07] beschrieben:

> „[...] a service oriented architecture is an architecture for building business applications as a set of loosely coupled black-box components orchestrated to deliver a well-defined level of service by linking together business processes."
>
> [SOA07] S. 27

SOA ist ein Ansatz im Bereich der Informationstechnik um Anwendungen oder einzelne Dienste aus verschiedenen Geschäftsprozessen zu bilden.

Melzer bietet eine ausführlichere Definition:

> „Unter einer SOA versteht man eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht. "
>
> [Melzer08] S. 13

Zur Verdeutlichung einer SOA kann ein beispielhafter und vereinfachter Aufbau eines Online-Shops verwendet werden.

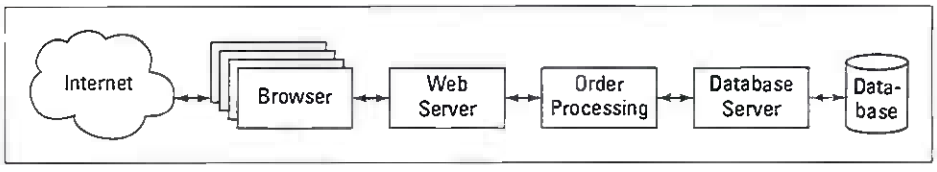


Abbildung 6: Simple Software Architektur eines Webshops[7]

Durch den gewöhnlichen Browser können Benutzer auf die Webseite des Webservers zugreifen um dort auf die eigentliche Anwendung des Webshops *Order Processing* zuzugreifen. Dabei werden durch einen Datenbankserver die Informationen in einer Datenbank gespeichert oder von dort der Webshop-Anwendung zugänglich gemacht. Welche Funktion die Anwendung *Order Processing* ausführt hängt von den Aufforderungen des Benutzers durch den Browser ab.
Dieser Struktur wird nun ein Service-Orientierte Komponente *Credit Checking* hinzugefügt, siehe Abbildung 7.

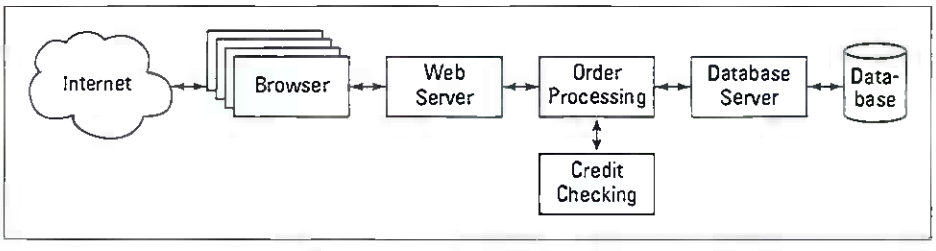


Abbildung 7: Hinzugefügte Service-orientierte Komponente[8]

Dabei hat die eigentliche Anwendung des Webshops keine Kenntnis wie die Komponente *Credit Checking* intern abläuft, sondern übergibt nur die essentiellen Informationen, in diesem Fall die Kreditkartendaten, an die Komponente. Für die Anwendung ist irrelevant, ob diese Komponenten eine externe Datenbank oder Webseite nach der Kreditwürdigkeit des Benutzers befragen,

[7] Quelle: [SOA07] S. 18
[8] Quelle: [SOA07] S. 20

solange die Komponente auswertbare Informationen (zahlungsfähig ja/nein) an die Webshop Anwendung liefert. Für die Anwendung *Order Processing* ist die Komponente *Credit Checking* eine so genannte **black box**.
Die komplexen Berechnungen und Algorithmen zur Bestimmung der Kreditwürdigkeit des Benutzers werden komplett verdeckt, so dass nur die Kreditkarteninformationen der Komponente zu übergeben sind.
Die Komponente *Credit Checking* steht der Webshop Anwendung als **abstrahierter Dienst bzw. Service** zur Verfügung.

3.5 Web-Services-Architektur

Wie bei dem Begriff SOA gibt es für Web Services keine allgemein gültige Definition, jedoch überlappen sich Definitionsvorschläge in verschiedenen Gesichtspunkten. Laut Melzer ([Melzer08] S. 55) bietet das World Wide Web Consortium (W3C) den konkretesten Ansatz einer passenden Definition.

> „A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."
>
> [W3WS04] S. 7

Ein Web Service ist so aufgebaut, dass ein Zusammenspiel zwischen Rechner über ein Netzwerk möglich ist. Dabei ist Schnittstelle des Web Services in einem maschinell interpretierbaren Format gehalten, so dass andere Systeme auf diese Schnittstelle zugreifen können. Dieser Zugriff findet durch das Simple Object Access Protocol (SOAP) statt, welches üblicherweise über das

Hypertext Transfer Protocol (HTTP) versendet wird. Die SOAP-Nachrichten sind nach dem XML-Schema zusammen mit anderen Web-Standards aufgebaut. Dadurch können die Nachrichten von beiden Seiten (Client und Server) interpretiert werden.
Als Beispiel verschickt der Client zwei Zahlenwerte, die vom Server addiert werden sollen:

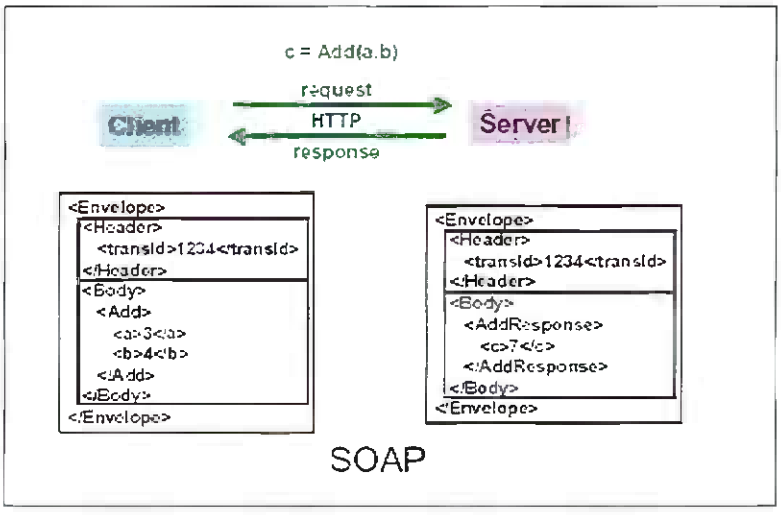


Abbildung 8: Kommunikationprotokoll SOAP[9]

Der Server entpackt die SOAP-Nachricht und führt mit den zwei Zahlenwerte die Addition aus. Das Ergebnis der Rechnung wird im Anschluss wieder als Nachricht im SOAP-Format an den Client zurückgesendet.
Daraus leitet Melzer folgende Spezifikationen für eine Web-Services-Architektur ab:

SOAP beschreibt das XML-basierte Nachrichtenformat der Kommunikation und dessen Einbettung in ein Transportprotokoll.

WSDL ist eine - ebenfalls XML-basierte - Beschreibungssprache, um Web Services (Dienste) zu beschreiben.

[9] `http://www.devarticles.com/c/a/PHP/Building-XML-Web-Services-with-PHP-NuSOAP/1/`

UDDI beschreibt einen Verzeichnisdienst für Web Services. UDDI (Universal Description, Discovery and Integration protocol) spezifiziert eine standardisierte Verzeichnisstruktur für die Verwaltung von Web-Services-Metadaten. Zu den Metadaten zählen allgemeine Anforderungen, Web-Services-Eigenschaften oder die benötigten Informationen zum Auffinden von Web Services.

[Melzer08] S. 55

Dabei erwähnt Metzer ([Melzer08] S. 56), dass ein Verzeichnisdienst keine Notwendigkeit für die Verwendung eines Web Services ist, „sondern vielmehr die Infrastruktur zum Auffinden von geeigneten Web Services beschreibt." Der Ablauf der Benutzung eines Web Services soll durch Abbildung 9 verdeutlicht werden.

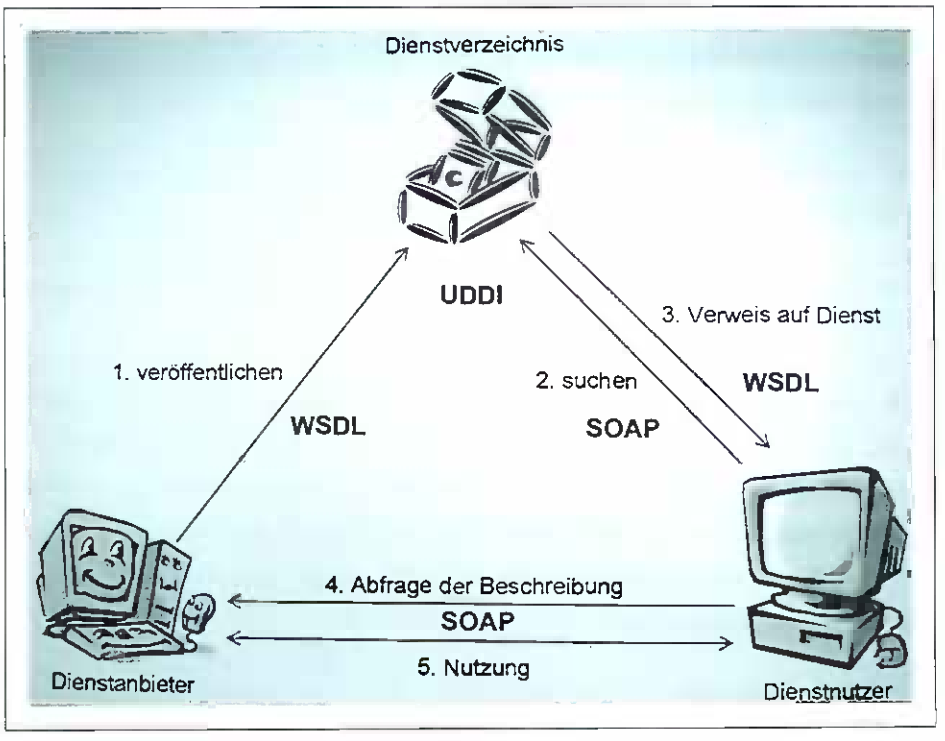


Abbildung 9: Ablauf einer Web Service-Benutzung[10]

1. Der Anbieter des Web Services muss seinen Dienst durch eine WSDL-Datei in Form einer XML-Datei dem Dienstverzeichnis bekannt geben.

[10]Quelle: [Melzer08] S. 56

2. Erst dann können mögliche Nutzer dieses Dienstes den Web Service im UDDI-basiertem Dienstverzeichnis finden. Die Suchanfrage findet über eine SOAP-Schnittstelle statt.

3. Ein Verweis auf den Dienst in Form einer WSDL-Datei wird an den Dienstbenutzer als Antwort der Suchanfrage gesendet.

4. Durch diesen Verweis erfährt der Benutzer die Adresse des Dienstanbieters und kann die Beschreibung des Web Services abfragen.

5. Nach Erhalt dieser Beschreibung kann der eigentliche Webdienst mittels SOAP verwendet werden.

Wirklich UDDI ist gestorben! Les' das noch einmal online nach.