

Fakultät für Informatik
der Technischen Universität München



Forschung- und Lehrereinheit Informatik III
Prof. R. Bayer Ph.D., Prof. Dr. D. Kossmann

Hauptseminar im Sommersemester 2003

Web Services

Ausarbeitung für den Vortrag

Workflow

Referentin: Bc. Hana Stetinova
Betreuer: Dipl.-Inf. Peter Fischer
Abgabetermin: 15. Mai 2003
Votragsdatum: 22. Mai 2003

Inhaltsverzeichnis

1. Einführung

- 1.1 Workflow
 - 1.1.1 Beispiel
- 1.2 Workflow Management System (WFMS)
 - 1.2.1 Historie und Überblick
 - 1.2.2 Beispiele von WFMS

2. Standardisierung von WFMS

- 2.1 Workflow Management Consortium
- 2.2 Object Management Group

3. Modellierung von Workflows

- 3.1 Statecharts
 - 3.1.1 E-Commerce Beispiel

4. Architektur von WFMS

- 4.1 Zentralisierte versus Verteilt
 - 4.1.1 Zentralisierte Architektur
 - 4.1.2 Verteilte Architektur
 - 4.1.3 Bewertung

5. Implementierung

- 5.1 CORBA (klassischer Ansatz)
- 5.2 Java-Basierte
- 5.2 XML-Basierte
- 5.3 Bewertung

6. Ausblick auf Webservices

- 6.1 WSFL
- 6.2 Xlang
- 6.3 BPEL4WS
- 6.4 Bewertung

7. Zusammenfassung

8. Literaturverzeichnis

Anhang A: Glossar

1. Einführung

Der Wettbewerb und ökonomischer Druck zwingt moderne Business Firmen nach neuen Informationstechnologien zu suchen, um ihre Geschäft Prozesse zu unterstützen.[MSKW] Besonders im Bereich von Web Services und E-Commerce, wo sich die Komplexität der Prozesse ständig erhöht und das Management immer anspruchsvoller wird, besteht ein Bedarf nach Informatik Systemen, die solche Prozesse automatisieren, steuern und überwachen. Die Workflow Technologie bietet einen Model für Geschäftsprozesse und die Grundlage, auf der die Lösungen für das Management gebaut werden können. Dies ist der Hauptgrund, warum die Workflow Management Systeme schon in den letzten Jahren viel Aufmerksamkeit erregt haben und zu einem wichtigen Bestandteil von Web-Service Systemen werden.

1.1 Was is Workflow

Workflow kann man vereinfacht definieren als Satz von Aufgaben (tasks), auch genannt Aktivitäten (activities) oder Schritte (Steps), die zusammen kooperieren, um einen Geschäftsprozess umzusetzen. Eine gute Workflow Technologie kann auch einen Weg bieten, die bestehende Systeme zu integrieren und unterstützt die wichtigen Änderungen (Evolution) von Organisation und Technologie.

Ein Workflow besteht aus eine Sammlung von koordinierten Aufgaben, die interagieren miteinander und dazu entworfen sind, einen wohl-definierten, komplexen Prozess durchzuführen. Typische Anwendungsbeispiele umfassen Online-Auktionen, Katalog-Bestellung, Reise Planung, oder Unterstützung von Geschäfts- und Herstellungsprozessen in eine Firma.

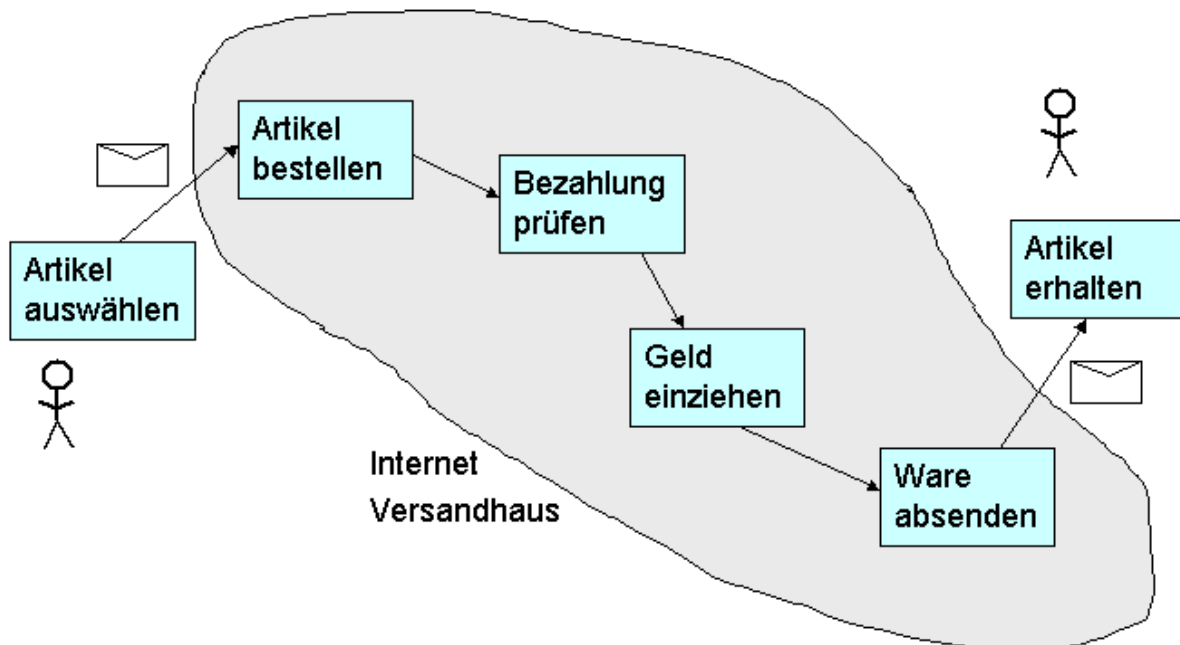


Abbildung 1a: Einfacher Workflow

Die **WFMC (Workflow Management Coalition)** definiert einen **Workflow** wie folgt [WFM99]:

Definition 1.1.1 (Workflow)

[A Workflow is] „the computerised facilitation or automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.“

Übersetzung: „Die computergestützte Unterstützung oder Automatisierung eines Geschäftsprozesses im Ganzen oder in Teilen davon, während deren Dokumente, Informationen oder Aufgaben zur Bearbeitung, entsprechend einer Reihe von prozeduralen Regeln, von einem Beteiligten an den nächsten weitergereicht werden.“

Der abzubildende Geschäftsprozess wird mittels des Prozessdefinitions-Tools von der realen Welt in eine durch den Computer verarbeitbare Form gebracht. Das Ergebnis dieses Schrittes wird als Prozess, oder Workflowmodell bezeichnet.

Ein Modell besteht aus einem Netz von (verteilten) Aktivitäten und den Beziehungen zwischen diesen.

Zur Ausführung einer konkreten Aktivität werden dieser ein oder mehrere Workflow-Bearbeitern zugeordnet. Hierbei handelt es um eine Person oder eventuell auch um eine maschinenbasierte Ressource, wie etwa einen Intelligenzen Agenten. Eine Aktivität wird dem Bearbeiter im allgemeinen in Form von einem oder mehreren Aufgaben (Work-Items) in einer Arbeitsliste (ToDo-List) repräsentiert, welche die durchzuführenden Arbeitsschritte festlegen.

1.2 Workflow Management System (WFMS)

Hauptaufgabe eines Workflowsystems ist die Koordination von Arbeitsschritten zwischen Benutzern und Systemkomponenten. Diese erfolgt entsprechend einer zuvor definierten Ausführungsanweisung und umfasst die Zuordnung von einzelnen Arbeitsschritten/Programmen, Daten und Benutzern. Ziel ist es dabei, nicht nur die Abarbeitung der einzelnen Arbeitsschritte, sondern vor allem auch die Koordinierung und Überprüfung gemäß Qualitätskriterien rechnerunterstützt zu optimieren.

In [WFM99] wird ein Workflow-Management-System als ein aus Software bestehendes System bezeichnet, das die Erzeugung und Ausführung von Arbeitsabläufen unterstützt, mit den an der Durchführung der Arbeitsabläufe beteiligten Personen (Workflow-Participant) interagiert und, falls notwendig, den Aufruf von Applikationen veranlasst.

Dazu muss das System in der Lage sein, maschinenlesbare Beschreibungen der Arbeitsabläufe (Workflowmodelle) zu speichern, zu interpretieren und daraus einzelne Ablauf-Instanzen (Workflow-Instanzen) ableiten zu können. Neben dem Begriff des Workflow-Management-Systems sind auch die Begriffe Workflow-Automation-System, Workflow-Manager und Workflow-Computing-System gebräuchlich.

Workflow
specification

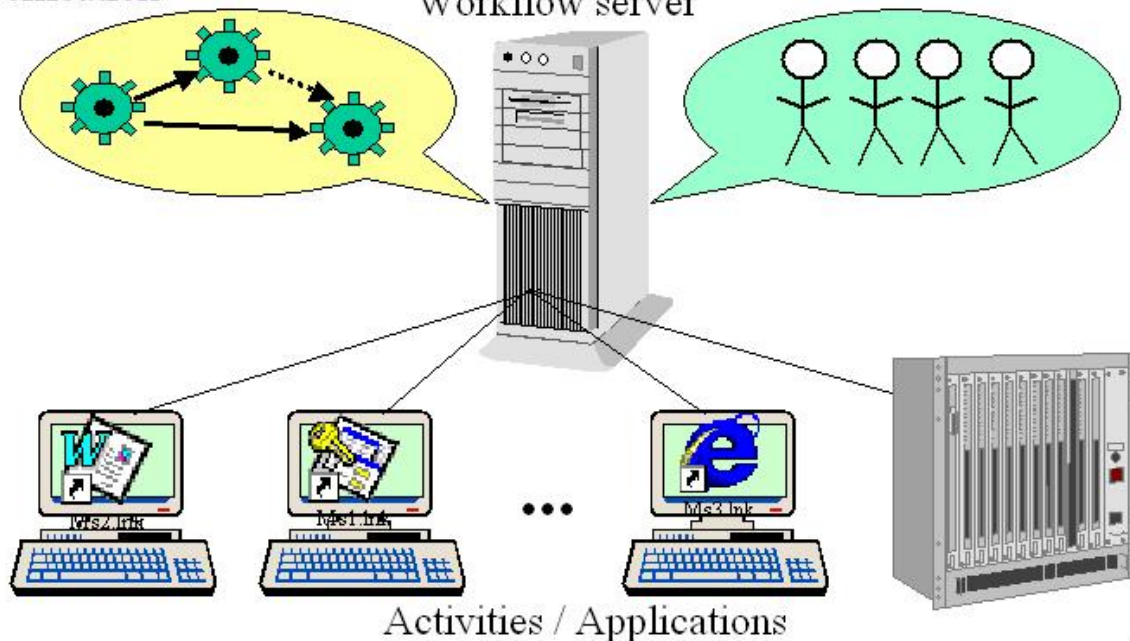


Abbildung: Workflow Management System Architektur [WEI]

Ein Workflow-Management-System benötigt zur Ausführung von Workflow-Instanzen Ressourcen. Bei Ressourcen kann es sich sowohl um Personen, die an der Bearbeitung eines Workflows beteiligt sind, und Maschinen, welche z. B. im Rahmen eines Produktions-Workflows eingesetzt werden, handeln. Daneben stellen auch die benötigten Daten, (z. B. ein CAD-Entwurf oder ein entwickeltes Simulationsmodell) und Programme (z. B. die Verfügbarkeit einer Laufzeitlizenz für ein bestimmtes Programm) Ressourcen dar, die zur Ausführung einer Workflow-Instanz notwendig sind.

Wie funktioniert das ganze?

Nach der Initialisierung durch den Benutzer wird eine Workflow Spezifikation instantiiert und interpretiert von der Workflow Engine. Eine oder mehrere Engines formieren einen so genannten Workflow Enactment Service. Während der Ausführung werden Aktivitäten erzeugt. Diese Aktivitäten korrespondieren entweder mit Workflow Client Applikationen oder mit aufgerufenen (invoked) Applikationen. Die Workflow Client Applikationen sind typischerweise interaktive Aktivitäten, wie z.B. intellektuelle Entscheidungen, aufgerufene (Invoked) Applikationen sind automatisierte Aktivitäten wie z.B. die Host Applikationen. Workflows können auch weitere Subworkflows enthalten, die von einem anderen, externen Workflow Engine kontrolliert werden. Um den Workflow zu analysieren und um ein Feedback zurück zu erhalten, werden Administration und Monitoring Tools eingesetzt. [SGW]

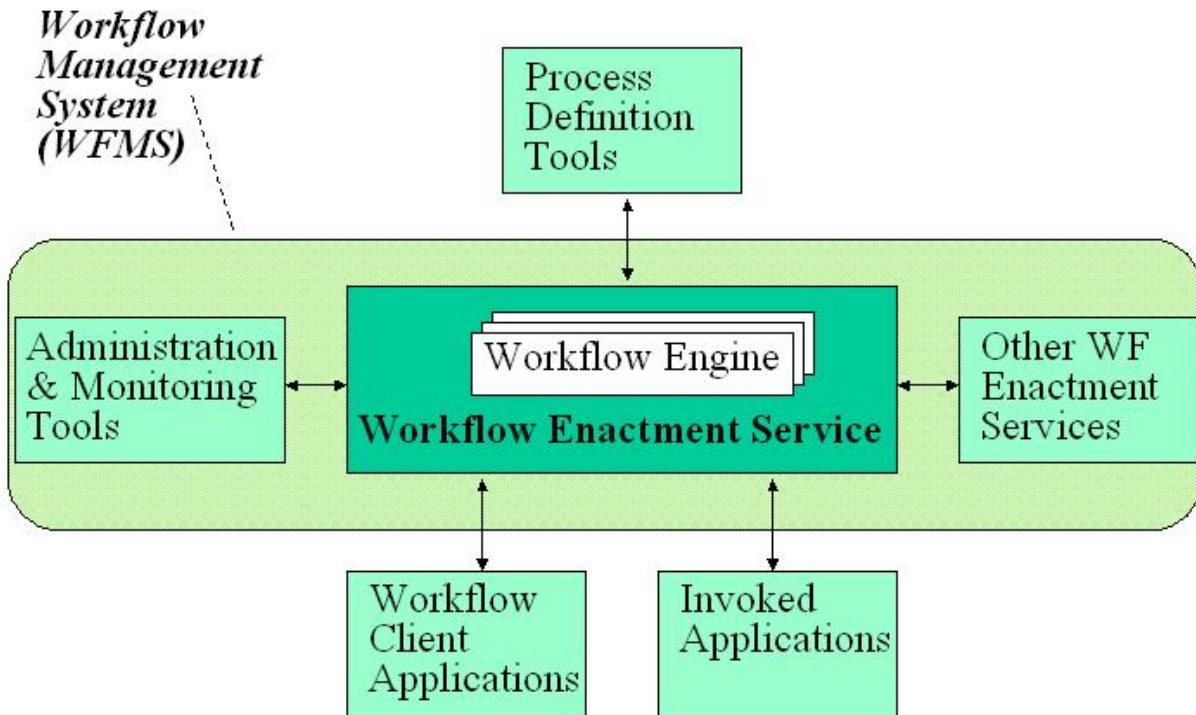


Abbildung 1c: WfMC Reference Architektur [WFMC]

Im folgenden wird zuerst ein kurzer Abriss über die Historie und die Herkunft von Workflow-Management-Systemen gegeben. Anschließend werden die zugrunde liegende Architektur und das Workflow-Modell vorgestellt.

1.2.1 Historie und Überblick

Workflow Management Systeme haben in letzten Jahren viel Aufmerksamkeit erhalten, und es ist viel getan worden für ihre Entwicklung und Standardisierung, aber das bedeutet nicht, dass es die Workflowsysteme erst seit kurzem existieren.

Die Ursprünge von Workflowsystemen können in den Bereichen Büroautomatisierung, Dokumentenverwaltung und Groupware Produkten gesehen werden, schon in den frühen 80ern Jahren [JAB]. Jeder dieser Bereiche enthält Funktionalitäten, die sich heute in den meisten Workflowsystemen wieder finden. Im Bereich Büroautomatisierung sind das u. a. Arbeitsplanung, die Funktionsintegration, die Aufgabenverwaltung und die Assistenz durch das System. Eng damit verwandt ist auch der Bereich der elektronischen Dokumentenverwaltung. Aufgaben, wie sie auch im Bereich Workflow-Management vorkommen, sind die Verteilung von Dokumenten an unterschiedliche Personen-Gruppen, die Verwaltung unterschiedlicher Zugriffsrechte und damit verbunden die Anwendung bestimmter Applikationen auf diese Dokumente. Aber auch erweiterte e-Mail Systeme weisen Funktionen auf, die in Workflowsystemen zu finden sind.

Die Möglichkeit, Informationen über weitere Bearbeiter, an die z.B. eine Nachricht oder ein Dokument nach der Bearbeitung geschickt werden soll, zu integrieren, erlaubt es, Ausführungssequenzen, wie sie auch bei Workflowsystemen üblich sind, zu definieren. Weitere Ursprünge liegen in Groupware Systemen, welche die Fähigkeit aufweisen, die Zusammenarbeit

von verschiedenen Gruppen oder Individuen zu unterstützen. Dies ist auch eine gewünschte Eigenschaft von Workflowsystemen.

Workflowsysteme sind bei gleicher Spezialität der Anwendung und Strukturierung der Arbeitsabläufe, in ihrem Einsatz weitaus flexibler sind als hardcodierte Systeme. Analog unterscheiden sich die Workflowsysteme von den Groupwareprodukten hauptsächlich durch die stärkere Strukturierung der Abläufe, bei einer ähnlich starke Aufgabenteilung und hoher Flexibilität.

Hauptmerkmal eines Workflowsystems ist die Trennung von Applikation und Ablaufsteuerung. Diese Separation kann analog zur Trennung zwischen Applikation und Datenhaltungskomponente in Datenbankanwendungen gesehen werden und erlaubt bei der Entwicklung von Anwendungen die Konzentration auf die Applikationsaspekte [JAB].

2. Standardisierung von WFMS

Workflow-Management-Systeme (WFMS) werden von vielen Experten als sehr Erfolg versprechender Markt mit enorm hohem Wachstumspotential angesehen [JAB]. Aus diesem Grund ist es auch nicht verwunderlich, dass in den letzten Jahren eine Vielzahl von Systemen auf dem internationalen Markt erschienen sind, die für sich alle den Begriff „Workflow“ in Anspruch nehmen. Ein Hauptproblem der erhältlichen Systeme ist, dass sie alle zueinander inkompatibel sind. Dies führte bei potentiellen Anwendern zu der Sorge, bei ihrer Entscheidung, ein Workflowsystem einzusetzen, möglicherweise aufs „falsche Pferd“ zu setzen und sich in eine Sackgasse bzw. in eine enge Abhängigkeit von einem Hersteller zu begeben. Insbesondere durch die nicht unerheblichen Investitionen, die mit der Einführung von Workflow-Management-Systemen verbunden sind, führte dies dazu, dass viele Firmen auf den Einsatz von Workflowsystemen verzichteten, auch wenn die Vorteile eines solchen Systems klar auf der Hand lagen. Dies führte 1993 zur Gründung eines Interessenverbands, der sich mit der Standardisierung von Workflow-Systemen beschäftigt.

2.1 Workflow Management Coalition

Die WorkFlow Management Coalition (WFMC) wurde im August 1993 als non-Profit Interessenverband von Workflow-Management-System Anbieter und Anwender gegründet. Ziel der WFMC ist, einen Standard im Bereich Workflowsysteme zu etablieren, der die Interoperabilität zwischen Produkten verschiedener Hersteller ermöglicht und die Verbreitung der Workflow Technologie fördern soll [WFMC].

Dies soll durch die Einführung einer Reihe von Schnittstellen erreicht werden, die es den Anwendern erlauben, bestimmte Module eines Herstellers durch die Module anderer Hersteller zu ersetzen oder zu ergänzen.

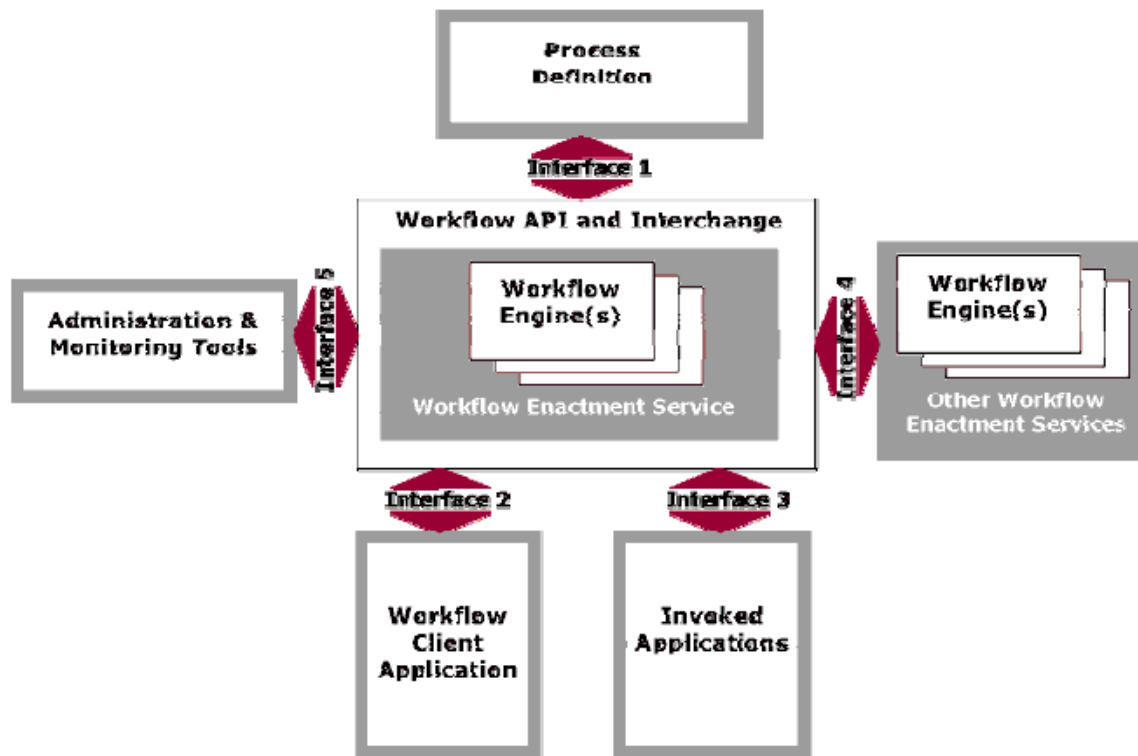


Abbildung 2a: WFMC Workflow Referenz Modell [WFMC]: Komponenten und Schnittstellen

Abbildung 2a zeigt fünf von der WFMC definierte Schnittstellen, die jeweils zwischen der zentralen Komponente eines Workflowsystems, der eigentlichen Engine, und den anderen Komponenten definiert sind, und die im folgenden kurz vorgestellt werden:

Interface 1: (Workflow Process Definition Read/Write Interface)

Schnittstelle zum Austausch von Prozessdefinitionen. Dazu wird in [WFM98c] eine Grammatik vorgeschlagen, welche die WPDL2 beschreibt. Die WPDL soll die Formulierung der Arbeitsabläufe erlauben.

Interface 2: (Workflow Client Application Programming Interface)

Hierbei handelt es sich um die Schnittstelle zu Workflow Client Anwendungen, die die Realisierung von speziellen Benutzerfrontends erlauben soll [WFM98a].

Interface 3: (Invoked Applications)

Diese Schnittstelle soll eine Standardisierung beim Applikationsaufruf realisieren. In neueren Dokumenten der WFMC ist das Interface 3 mit in das Interface 2 integriert worden [WFM98a].

Interface 4: (Interoperability)

Interface 4 regelt die Interoperabilität zwischen verschiedenen WFMS. Dies beinhaltet primär, dass konforme Workflowsysteme in der Lage sind, die Prozessdefinitionen auszutauschen und zu interpretieren [WFM98d]. Im Zusatzdokument [WFM98b] wird ein Mechanismus zum Nachrichtenaustausch auf MIME Basis definiert.

Interface 5: (Draft Audit Specification)

Die Draft Audit Specification standardisiert den Zugriff auf Informationen zur Administration von Workflows, um beliebige Administrations- und Monitoringtools einsetzen zu können.

2.2 Object Management Group (OMG)

Neben der WFMC gibt es noch ein zweites Konsortium, das sich mit der Standardisierung von Workflows auseinandersetzt. Die Object Management Group (OMG) hat sich zum Ziel gesetzt, die Verbreitung der Objekttechnologie zu fördern und hat mit CORBA (Common Object Request Broker Architecture) eine Referenzarchitektur [OMG95] vorgestellt. CORBA stellt eine plattform- und sprachunabhängige Infrastruktur für verteilte Objekte bereit. Die OMG hat sich anschließend entschlossen, ihre Standardisierungsbemühungen auf weitere Gebiete auszuweiten und mit den CORBA Facilities.

Standardisierungsbestrebungen sind für eine Reihe von weiteren Teilgebieten vorgeschlagen, die als Dienste bezeichnet werden. Es werden hierbei horizontale (Anwendungsgebiet unabhängige) und vertikale (auf Anwendungsgebiete spezialisierte) Dienste unterschieden. Allen diesen Facilities ist gemeinsam, dass sie auf der von CORBA bereitgestellten Objekttechnologie aufsetzen. Einer der vorgeschlagenen anwendungsabhängigen Dienste ist die Workflow Facility (hierbei handelt es sich um einen Teil der Task Management Dienste), für die Anfang 1997 eine Arbeitsgruppe eingerichtet wurde.

Die Gruppe veröffentlichte im Mai 1997 ein entsprechendes Request for Proposal (RFP), als dessen Ziel „die Definition einer Schnittstelle und deren Semantik für die Ausführung und Manipulation von verteilten Workflow Objekten und deren Metadaten“ angegeben wird. Die Workflow Facility soll hierbei als Plattform zum Aufbau flexibler Management Anwendungen dienen, welche die Verbindung von Objekten und existierenden Applikationen realisiert. Im Juli 1998 wurde daraufhin als Reaktion auf das RFP ein überarbeiteter Vorschlag, basierend auf Standards der WFMC, veröffentlicht [OMG98]. Dieser Vorschlag wird von einer Reihe führender Unternehmen getragen und stellt eine Basis für die Integration der Workflow Technologie in das OMG Architekturmodell dar.

3. Modellierung von Workflows

Ein Workflow besteht in der Regel aus mehreren, z. T. hierarchisch angeordneten Aktivitäten. Andere Bezeichnungen für Aktivitäten sind beispielsweise Step, Node, Task, Work-Element, Process-Element, Operation und Instruktion.

Ein Modell besteht aus einem Netz von (verteilten) Aktivitäten und den Beziehungen zwischen diesen Komponenten.

Ein Workflow Modell kann dazu verwendet werden, automatisierte oder halbautomatisierte Lösungen für bestimmte Geschäftsprozesse abzubilden, die innerhalb einer Firma vorkommen, oder sich über mehrere Firmen hinausstrecken. Workflow Modelle tendieren mehr dazu, mehr computer-orientiert zu sein als die traditionelle Geschäftsprozess Modelle.

3.1 Statecharts

Zur Modellierung und Beschreibung von Workflow Prozessen können unterschiedliche Darstellungsmodelle verwendet werden.

Als Standard werden so genannte Statecharts und Aktivitätcharts eingesetzt, die auch in UML Industrie-Standard inkludiert sind (Unified Modeling Language, OMG 1997). Damit wurde für die breite Anerkennung und Standardisierung gesorgt.

Alternativ können die Workflow Prozesse auch mit Hilfe von Petri-Netz Varianten beschrieben werden oder durch temporale Logik, Prozess Algebra, oder Skript Sprachen.

3.1.1 E-Commerce Beispiel

Da die Statecharts einen Standard darstellen, werden sie anhand eines Beispiels vorgestellt.

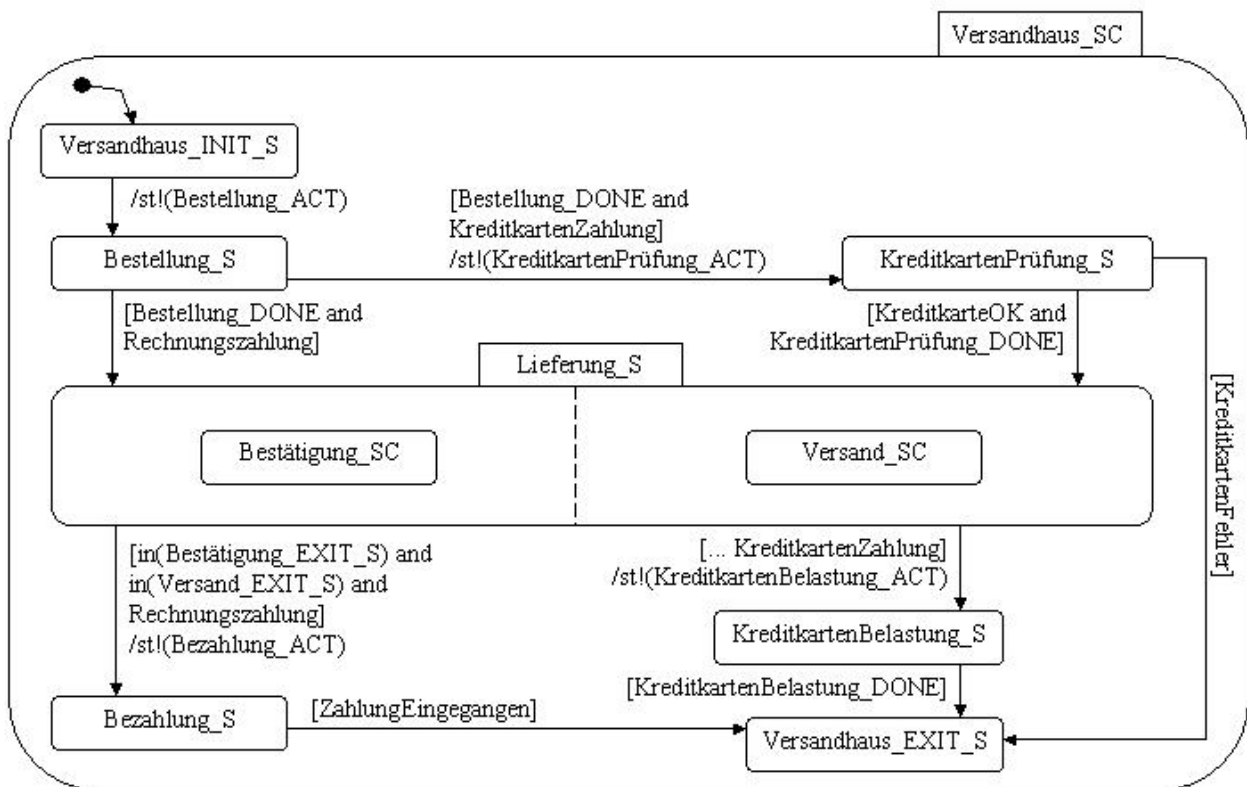


Abbildung 3a: Beispiel aus E-Commerce – Statechart von einem Versandhaus [WEI]

Damit die Statechart übersichtlich bleibt und die komplexe Funktionalität bis zum Details trotzdem modelliert werden kann, können auch verfeinerte Statecharts definiert werden.

Folgendes Beispiel (Abbildung 3b) stellt diese detaillierte Statechart dar.

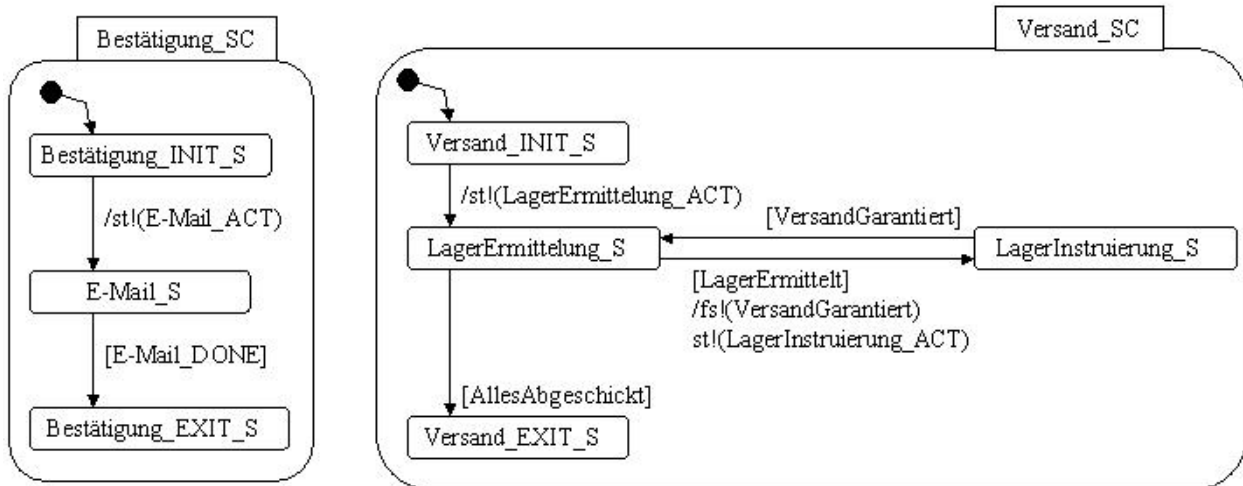


Abbildung 3b: Verfeinerte Statechart, Fortsetzung von Abbildung 3a.[WEI]

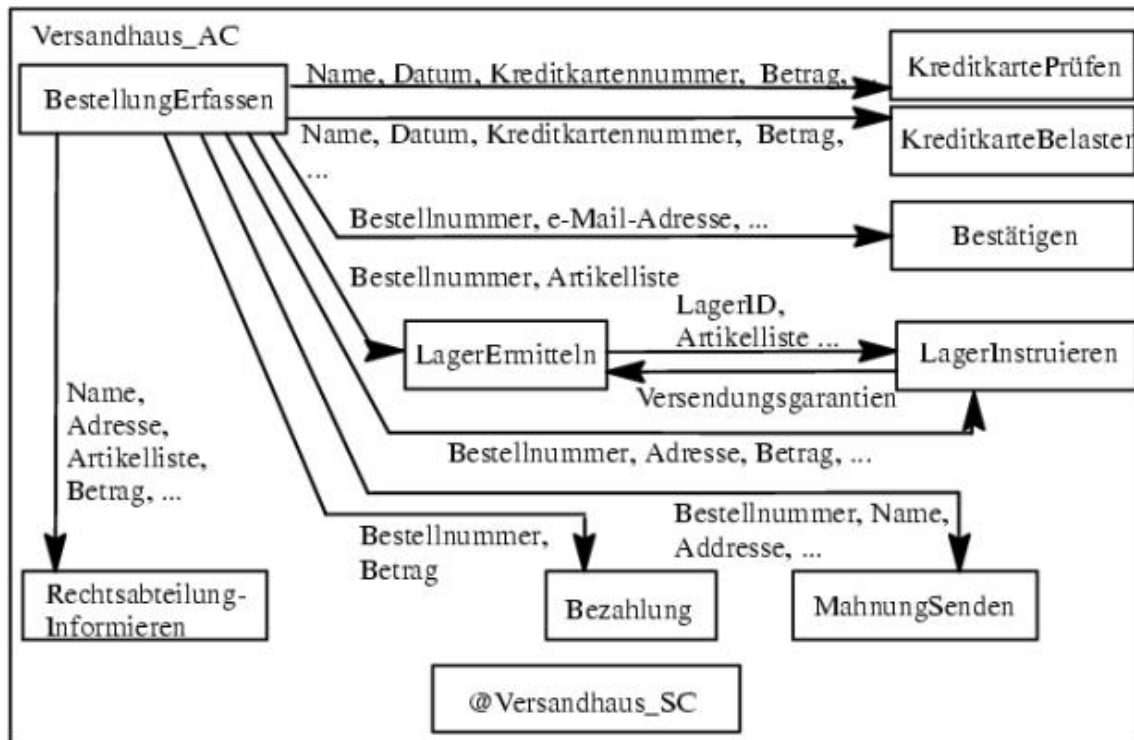


Abbildung 3c: Activity Chart

Workflow Charts (Statechart + Aktivitätschart) sind high-level visuelle Sprachen, abgeleitet von Geschäftsprozess Modellierung- und Definitionstools wie z.B. Aris Toolset. Dies folgt aus der geschichtliche Entwicklung (siehe 1.2.1 Historie und Überblick)

4. Architektur von WFMS

Prinzipiell lässt sich ein WFMS in zwei große funktionale Einheiten unterteilen. Zum einen ist

dies die Buildtime Komponente, auch Prozessdefinitions-Tool (Abbildung 2a oben) genannt, die zur Erstellung des aus der Realität abzubildenden Prozessmodells in eine durch einen Computer interpretierbare Beschreibung eingesetzt wird.

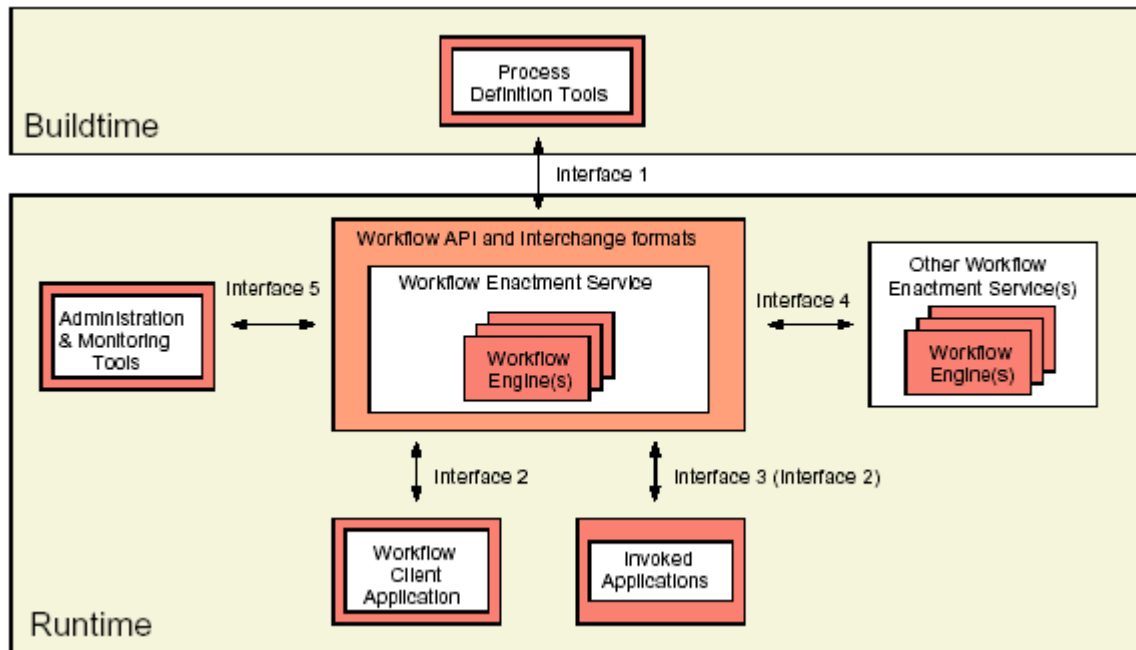


Abbildung 2a: WfMC Workflow Referenz Modell [HOL]: Komponenten und Schnittstellen

Die zweite Einheit wird durch die Runtime Komponente gebildet (Abbildung 2a unterer Block), welche die Abarbeitung der, mittels des Prozessdefinitions-Tools definierten, Workflows durchführt und überwacht. Hierzu zählen beispielsweise das Anlegen neuer Workflows, der Aufruf von Applikationen, die Interaktion mit dem Benutzer und das Verwalten der Arbeitslisten. Die Runtime Komponente lässt sich, wie aus Abbildung 2a ersichtlich, noch in Laufzeitkontrolle (mitte) und Laufzeitschnittstellen (unten) unterteilen. Erstere enthält die Workflow-Engine, welche für die Ablaufsteuerung sowie die persistente Speicherung der Daten zuständig ist. Die Laufzeitschnittstelle gliedert sich in die Teilbereiche Benutzerschnittstelle (unten, links), Aufrufschnittstelle für Programme (unten, rechts), Administrationschnittstelle (links) und das Interface zur Kommunikation mit anderen Workflowsystemen (rechts).

4.1 Zentralisierte versus Verteilte Architektur

In den folgenden Abschnitten werden die Unterschiede zwischen hoch-zentralisierter und verteilter Runtime Architektur von Workflow Management Systemen behandelt.

Es gibt auch weitere System-Architekturen [MSKW], z.B. synchrone zentralisierte Architektur, asynchrone zentralisierte und halb-verteilte Architekturen, die „Hybriden“ zwischen der hoch-zentralisierter und einer völlig verteilten Architektur sind.

4.1.1 Zentralisierte Architektur

Die zentralisierte Architektur (highly centralized architecture) ist in meist kommerziellen Workflow Management Systemen zu finden. Die Hauptkomponenten sind Workflow Scheduler, Task Manager (TM) und Tasks.

Es handelt sich um geschlossene Systeme, wobei alle Komponenten mitintegriert sind.

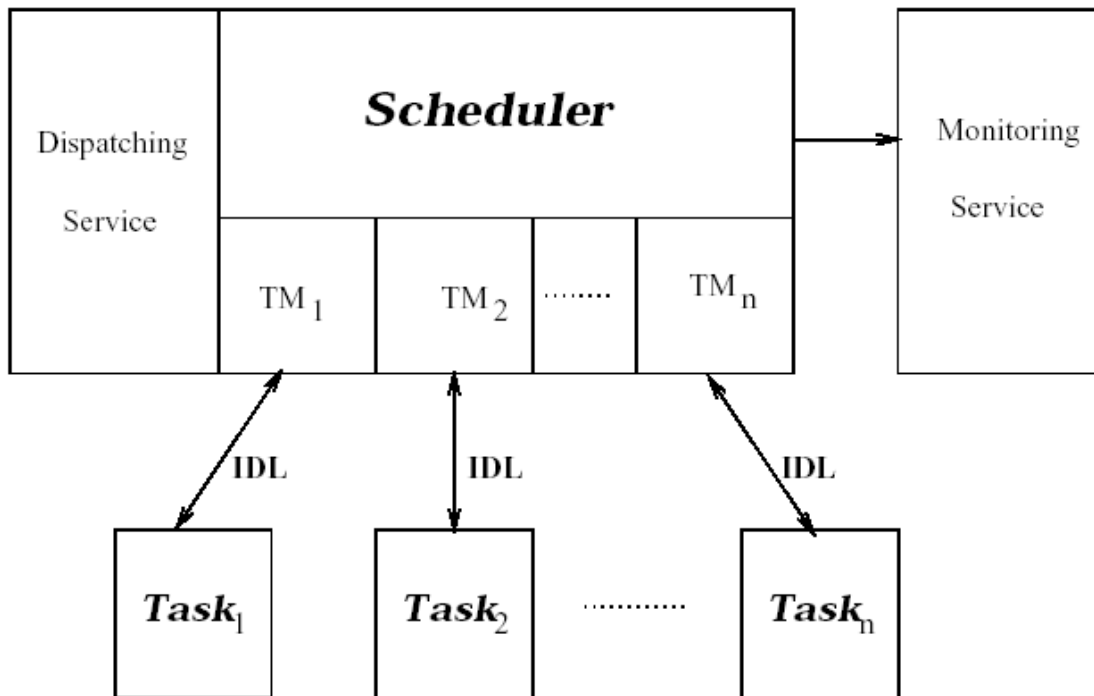


Abbildung 4a : Hochzentralisierte Architektur (Highly Centralized Architecture) [MSKW]

In der hochzentralisierte System Architektur sind die Task Managers (TM₁ bis TM_n) in dem Scheduler Prozess direkt mitintegriert. Dieser Prozess ist multi-threaded und hat jeweils einen Thread für den Scheduler's Dispatcher und die einzelnen Task Managers. Die Task Managers (TM) kommunizieren mit den Tasks durch eine CORBA IDL (Interface Definition Language) Schnittstelle. Diese Architektur ist auf der Abbildung 4a veranschaulicht, wo jede Box einen Prozess repräsentiert, die Unterteilung innerhalb eine Box repräsentiert einen Thread (eine. leichtgewichtiger Prozess).

4.1.2 Verteilte Architektur

In der völlig verteilten Architektur (fully distributed architecture), wie auf der Abbildung 4b gezeigt, hat keinen zentralisierten Scheduler.

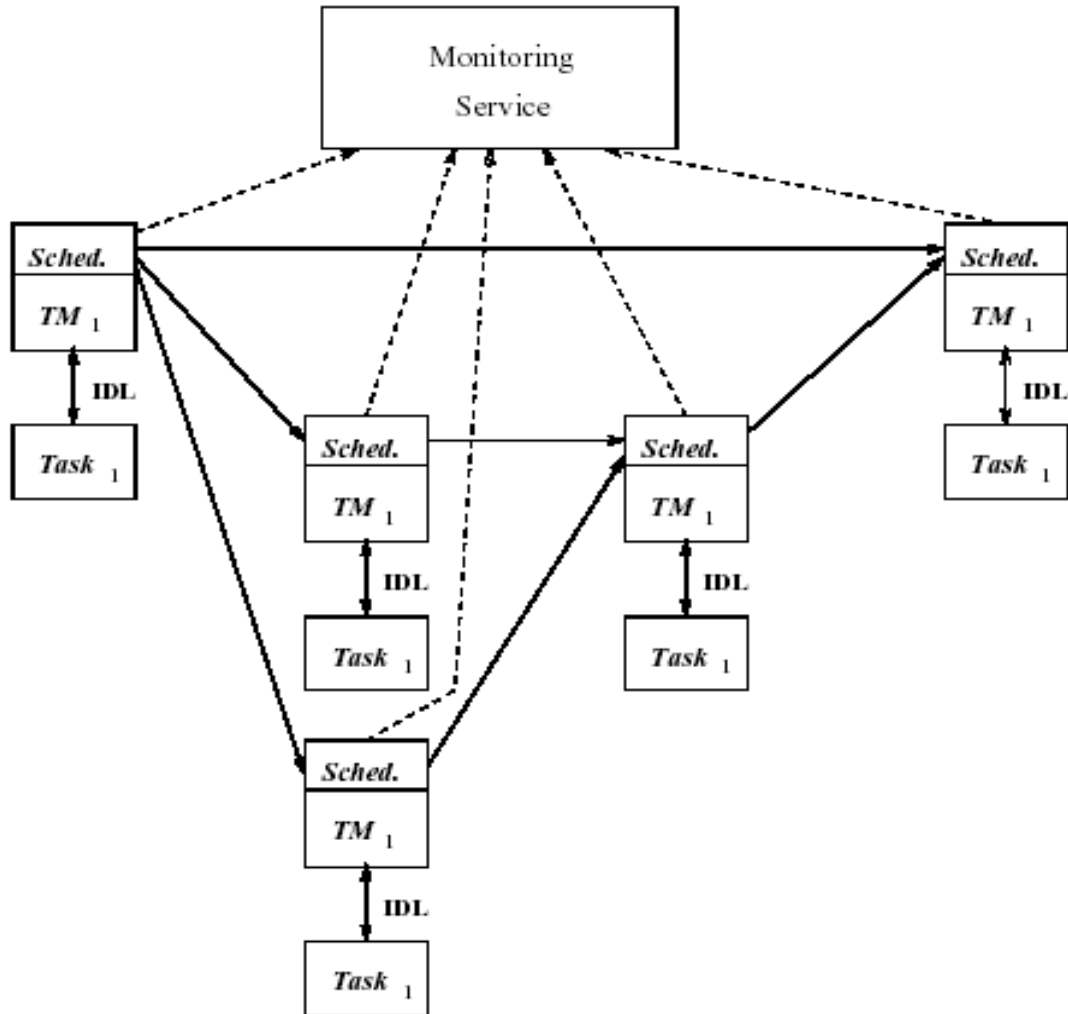


Abbildung 4b: Völlig Verteilte Architektur (fully distributed architecture) [MSKW]

Jeder Task Manager (TM) ist mit einem Kodefragment ausgerüstet das bestimmt ob und wann die zugehörige Task ausgeführt werden sollte. Seit ein Task von der Ausführung von anderen Tasks abhängen kann (entweder AND oder OR), ist ein AND-OR Baum Bestandteil von Scheduler. Die Individuellen Task Managers kennen nur die nachfolgende Tasks. Sie kommunizieren mit anderen Task Managers durch eine asynchrone IDL Schnittstelle. Im Rahmen der Kommunikation kann ein Task ID dazu verwendet werden, die Task Manager zu identifizieren, die die Nachricht geschickt haben.

Der Monitoring Dienst (monitoring service) überwacht die Ausführung des Workflow. Die Einzelnen Task Managers kommunizieren mit dem Monitor und teilen ihm sowohl ihren internem Zustand, als auch die Referenzen auf den Data Objekt mit, für den Fall von Recovery.

Die Kommunikation zwischen Task managers und den Monitor erfolgt asynchron, mittels ORB (Object Request Broker). Die verteilte Architektur passt sehr gut zu dem verteilten Charakter von Workflow. Es eliminiert den Flaschenhals von Task Managern, die mit einem entfernten zentralisierten Scheduler kommunizieren müssen während der Workflow-Ausführung. Anderer Vorteil dieser Architektur ist die Flexibilität bei einem Zusammenbruch (failure). Wenn ein Knoten zusammenbricht, ist nur ein Teil der Workflow ist davon betroffen.

4.1.3 Bewertung

Ein Hauptkritikpunkt an vielen kommerziellen Workflowsystemen ist die mangelnde Skalierbarkeit. Dies ist ein Hauptgrund für Performance-Probleme bei steigender Last sowie für mangelnde Verfügbarkeit und verhindert somit den Einsatz von Workflowsystemen in größerem Rahmen.

Viele Prototypen und kommerzielle Workflowsysteme basieren auf der standard client-server Architektur, definiert von WFMC. Die Workflow Engine, der Kern eines Workflow Management System, ist ein Server, der speichert typischerweise sowohl die Applikationsdaten (die Daten, die verwendet und generiert werden während jede Aktivität in Workflow) als auch die Workflowdaten (die Kontrolldaten mit Definitionen, Zustand, Ausführungsgechichte und andere Daten, die für die Ausführung relevant sind).[MSKW]

Es wird angenommen, dass die zentralisierte Workflow Engine eines Workflow Management Systems ist der Flaschenhals, weil die zentral gespeicherten Workflow-Daten im Fall eines Server-Zusammenbruches nicht mehr verfügbar sind und das ganzen Geschäftssystem paralysieren können.

Im Vergleich dazu bietet ein verteiltes Workflow Management System, wo einzelne Komponenten im Netz verteilt sind und zusätzlich einzelne Komponenten parallel auf mehreren Rechnern eingesetzt werden können, ein hohes Maß an Skalierbarkeit und Verfügbarkeit, infolge von Partitionierung und/oder Replikation. Und da ein Workflow von Natur aus verteilt ist - es kooperieren geografisch verteilte Anwendungen, scheint es auch natürlich das Workflow Management System als verteiltes System zu realisieren.

Trotzdem kann aber die Frage nach dem besseren Weg nicht eindeutig beantwortet werden, da wenn der zentralisierte Server ein leistungsstarken multiprozessor Rechner ist, die Beschränkung der zentralisierte Architektur nicht mehr von erheblicher Bedeutung ist. Ein anderer Vorteil der zentralisierte Architektur ist, dass dieses System leichter zu testen und zu administrieren ist.

Eine Studie [MSKW] kommt zu dem Ergebnis, dass die hoch zentralisierte Architektur besser geeignet ist, wenn die Tasks sehr rechenintensiv sind. Die mehr verteilte Architektur ist wiederum besser geeignet, wenn die CPU-Anforderungen der Task Manager deutlicher größer sind, als die der Tasks.

5. Implementierung von WFMS

5.1 CORBA-basiert

CORBA (Common Object Request Broker Architecture) stellt den klassischen Ansatz bei der Realisierung von Workflow Management Systemen dar.

CORBA wurde von der OMG (Object Management Group) entwickelt und die erste Spezifikation wurde schon in Oktober 1991 veröffentlicht. Es gehört inzwischen zum Standart und es sind zahlreiche kommerzielle ORB oder CORBA-basierte Produkte auf dem Markt.

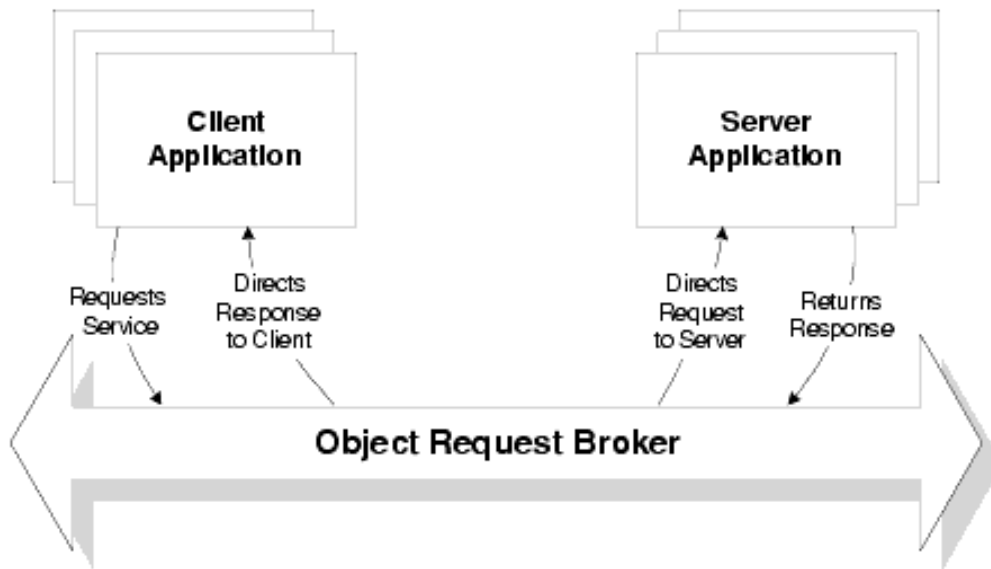


Abbildung 5a: CORBA Modell [BEA]

CORBA ermöglicht eine Kommunikation zwischen Heterogenen und Verteilten Systemen, indem es eine Schnittstelle in relativ einfache Beschreibungssprache IDL (Interface Definition Language) definiert. CORBA ist eine Erweiterung von Client-Server Architektur, Es trennt formal den Client- und den Serverteil der Applikation und separiert es als Objekte, die können bestimmte Operationen (Tasks) durchführen.

Der CORBA Client weiß nur wie er nach eine Task fragen sollte und der CORBA Server weiß nur, wie er die angefragte Tasks erledigt (ausführt). CORBA definiert den ORB (Object Request Broker) als den Mediator zwischen den Client and Server Applikationen.

Das hat natürlich Vorteile indem es den erlaubt die Server Implementation zu ändern, ohne die Clients zu beeinflussen und umgekehrt. [BEA]

Als Vereinfachung kann CORBA als RPC (Remote Procedure Call) gesehen werden, die auch Methodenaufrufe auf entfernten Objekten erlaubt (via ORB – Object Request Broker).

5.2 JAVA-basiert

Anderer Ansatz stellt die Java-basierte Implementierung dar. Dabei wird ein Teil der Schnittstellen zwar immer noch mit CORBA realisiert, aber die anderen Schnittstellen werden mit Hilfe von Java Technologie umgesetzt. Ein Beispiel für eine CORBA- und Java-basierte Workflow Management System Architektur ist auf der Abbildung 5b zu sehen.

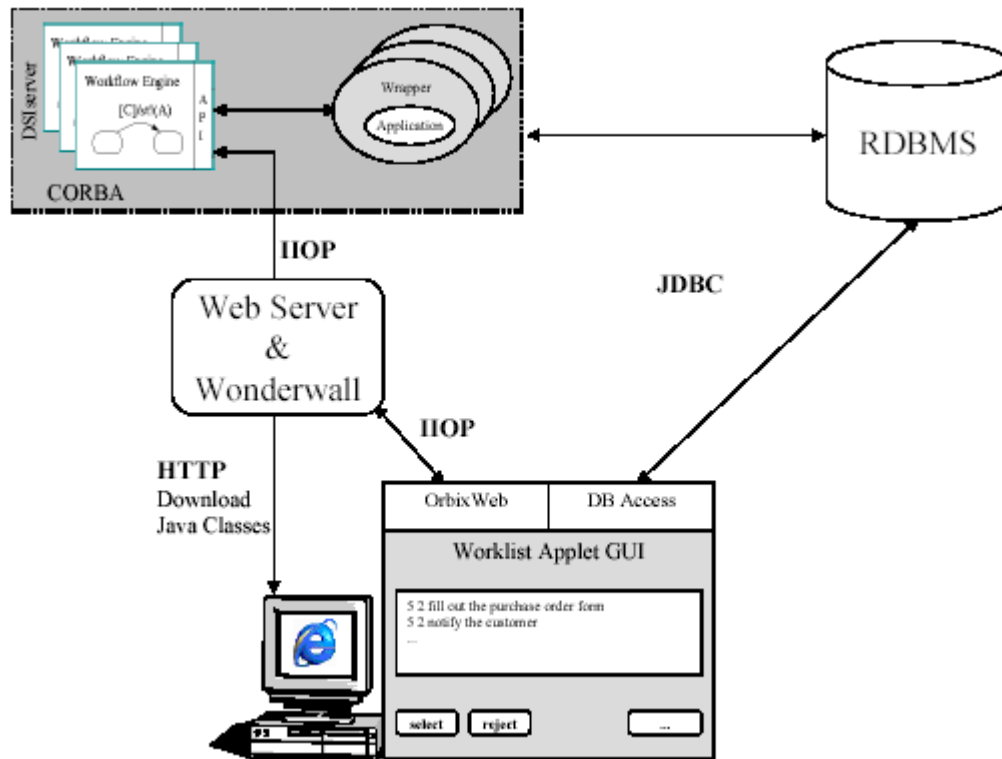


Abbildung 5b: Architektur von Java-basierte Mentor-lite Version [SGW]

Der Workflow Engine läuft als CORBA Applikation Server, ausgeprägt als DSI-Server (Distributed Statechart Interpreter), oder als eine Sammlung von mehreren solchen Servern, die auf unterschiedlichen Rechnern verteilt sind. Der DSI-Server verwendet eine Datenbank, um den Workflow Zustand und Kontext Daten abzuspeichern und auch um Workflow Ablaufsdaten zu verwalten. CORBA Technologie wird eingesetzt für die Schnittstelle API3 (Invoked Host Applikationen) und API4 (andere Workflow Engines), hier CORBA-orientierter IIOP (Internet InterORB Protocol).

Für die API2 (Client Applikation) und API5 (Administrierung und Monitoring Tools) werden in unserem Beispiel Java Applets verwendet. Der initialisierte Applet (Abbildung 5b rechts unten) wird von einer sicheren (trusted) Seite heruntergeladen und kontaktiert dann den DSI-Server für „work-to-do“, implementiert als http Aufruf, gehandelt durch IIOP-basierten OrbixWeb Service (installiert auf der Client Seite). Die Applets greifen danach direkt auf die Datenbank zu, über JDBC. Eine alternative Möglichkeit wäre, den Datenbank Zugriff immer indirekt, über den DSI-Server zu realisieren.

Die Nachteile der Java-basierte Architektur sind:

- Das Laden von Java Applets von einem sicherem Server ist sehr zeitraubend
- JDBC Aufrufe sind es auch, insbesondere eine JDBC Session zu etablieren
- Jeder Client etabliert eigene JDBC Session, was einen Skalierbarkeit Problem hervorrufen kann
- Für eine Workflow, die über mehrere Organisationen und Heterogenne Workflow Engines verteilt ist, müssten die Clients noch mehrere Sessions beibehalten, um den unterschiedlichen Servern zu genügen.

5.2. XML-basiert

Um die Nachteile der Java-basierten Architektur zu vermeiden, kann XML als Mediator dienen. Derzeit sind noch keine kommerzielle XML-basierte WFMS auf dem Markt, aber es existieren bereits die ersten Forschungsprototypen (z.B. XML-basierter Mentor-lite, entwickelt an der Universität Saarland). [SGW]

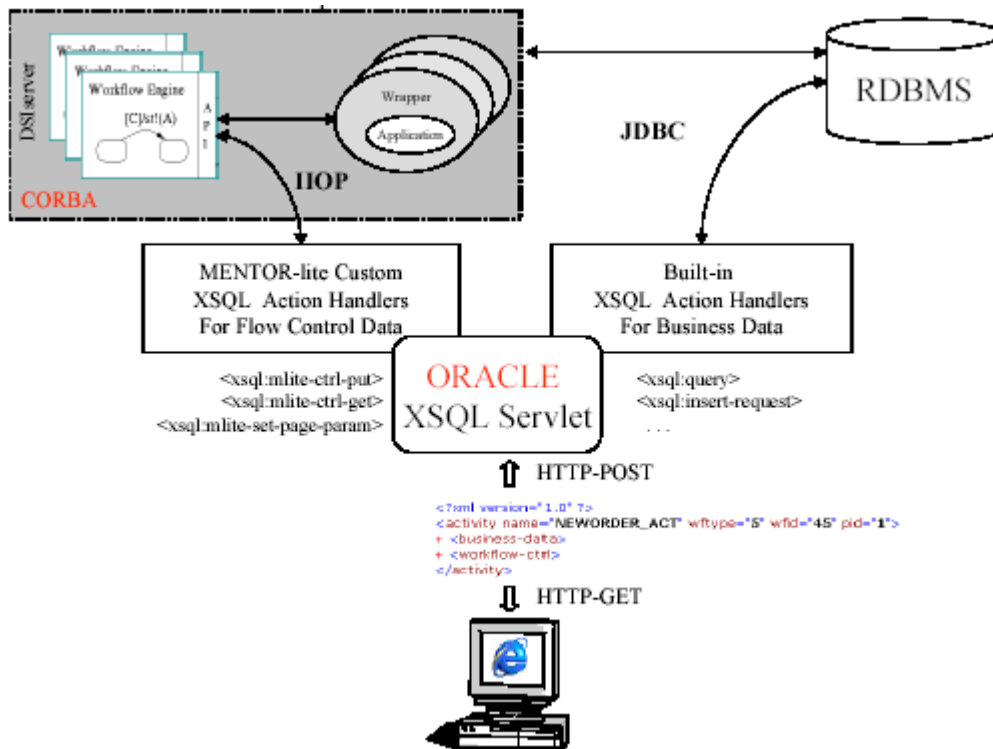


Abbildung 5c: Architektur von XML-basierter Mentor-lite Version [SGW]

In diesem System (Abbildung 5c) sind die Schnittstellen API2 (Client Applikation) und API3 (Invoked Applikation) modifiziert worden. Die Nachrichten von und zu Client Applikation sind in XML Format und enthalten Aktivitätsname und Workflow ID als XML Element.

Der modifizierte Client kann die Daten filtern und sortieren mit Hilfe eines XSLT Stylesheets, DHTML wird für die Datenrepräsentation und in eine benutzerfreundliche grafische Oberfläche eingesetzt .

Um die Performance Probleme von Java-basierter Architektur auszugleichen, interagiert ein XML-Mediator zwischen Clients und Workflow Engines und auch dem Datenbank Server (z.B. Oracle8i DB Server). Dieser Mediator wird als Servlet implementiert, der auch viele permanent geöffnete JDBC Sessions mit dem DB Server verwaltet, was das Performance Problem eliminiert.

Der XML Mediator vermittelt eine Abstraktion auf hohem Niveau, das die WFMS Interoperabilität in globalem, verteilten Workflow Management erhöht. Es bietet eine Internet-basierte, von überall zugreifbare Schnittstelle, um die Workflows und Aktivitäten aufzurufen und die Ergebnisse zu der richtige Stelle zu delegieren.

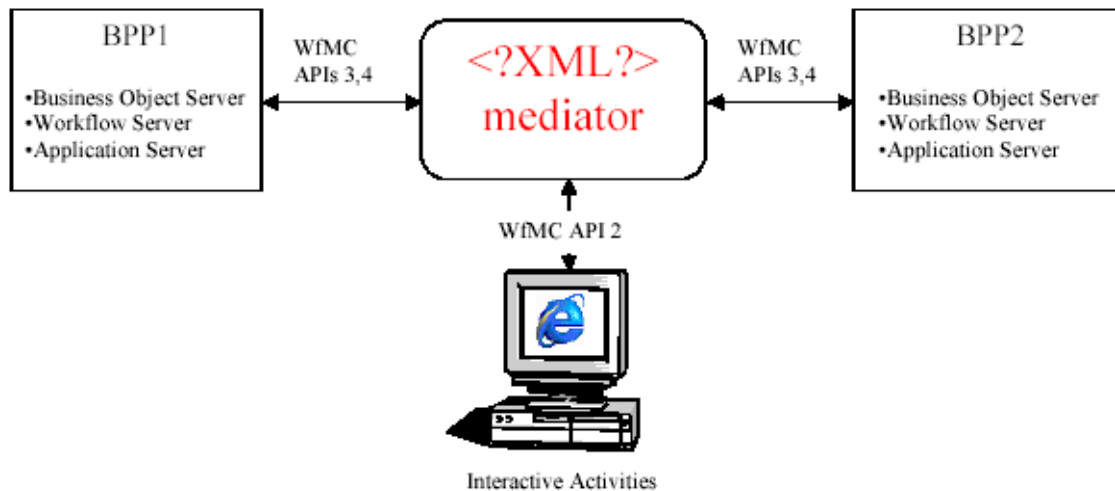


Abbildung 5d: Rolle von dem XML mediator [SGW]

Ein Szenario (angezeigt auf der Abbildung 5d) stellt zwei Geschäftsteilnehmer dar, die in globalen Geschäftsprozess involviert sind, z.B. in einem Web-Service, der personalisierte multimediale Nachrichten für angemeldete Benutzer bietet. Beide BPP (Business Process Participants), BPP1 als Business Portal Betreiber für diesen Web Service und BPP2 als Inhaltsanbieter, haben eigene spezifische IT Lösungen. Der XML Mediator kümmert sich um die Zusammenarbeit zwischen diesen zwei Informationssystemen. Dadurch werden auch globale Workflows ermöglicht, z.B. Konfigurieren von dem Service für neu angemeldeten Benutzer.

Der XML Mediator versteckt die Implementierungsdetails (hinterlegende Middleware, Firewalls etc.) von BPP1 und BPP2 für jede der beteiligten Seite. Besondere XML Nachrichten, die nur Business- und Kontrolldaten enthalten, werden via dem XML Mediator ausgetauscht. Der Mediator ist verantwortlich für die Lieferung der Nachricht, nach dem Empfänger Identifikation und den Tags in der XML Nachricht.

Ein Beispiel von einem Aktivitätsaufruf [SGW]:

```
<?xml version='1.0' ?>
<activity name='some name' wftype='x' wfid='y' pid='z'>
<business-data>
<relation1><row1><attr1>...</attr2></row1> ... </relation1>
...
</business-data>
<workflow-ctrl>
<variable name='var-name1' value='var-value1' />
...
</workflow-ctrl>
</activity>
```

Nachdem die Aktivität zu Ende ist, wird eine ähnlich strukturierte XML Nachricht mit dem Ergebnis zu dem XML Mediator geschickt, der dann diese Nachricht parst und die nötigen Schritte für folgende Aktionen (z.B. Speichern oder Modifizieren von Business Daten oder senden von Workflow Kontrolldaten) an den Workflow Engine leitet.

5.4 Bewertung

XML scheint das beste Austauschformat zu sein, da die Firewalls meistens alle TCP Ports bis auf den http (Port 80) blockieren. Die Länge des XML Nachrichtenformats kann mit der passenden Komprimierungstechnik kompensiert werden.

Der XML Mediator benutzt eine standard Technik der Wiederverwendung von Server Verbindungen, z.B. JDBC Sessions (er verwendet einen Pool von Verbindungen anstatt jedes Mal eine neue Verbindung zu öffnen). Dies erhöht natürlich die Skalierbarkeit.

Der Mediator ist Zustandlos, er braucht sich Nichts über die Clients oder Server Applikationen zu merken, was über eine einfache http get oder post Methode hinausgeht. Hieraus resultiert sehr wenig Overhead und hoher Mediator Durchsatz. Ein weiterer großer Vorteil ist, dass der Mediator leicht repliziert werden kann, was die hohe Verfügbarkeit des Mediator Service versichert.

Darum scheint der XML-basierter Ansatz im Vergleich zum Java-basierten mehr Performance und Skalierbarkeit zu bringen, was insbesondere für komplexe und anspruchsvolle Workflow Management Systeme wichtig ist.

6. Ausblick auf Webservices

Für Web Services gibt es derzeit noch keinen Standard des W3C Konsortiums, der das Protokoll für Workflow festlegen würde. Es gibt aber zahlreiche konkurrierende Vorschläge, die von unterschiedlichen Unternehmen entwickelt worden sind und warten auf die Anerkennung durch W3C, zum Beispiel WSCI (Web Services Choreography Interface), WSFL und XLANG. WSCI wurde vom Sun, SAP und BEA entwickelt.

Die wichtigsten „Pre-Standards“ sind die WSFL von IBM, und XLANG von Microsoft. Am Anfang gegeneinander stehenden Großunternehmen haben sich aber letztendlich gereinigt und gemeinsam die BPEL4WS Sprache entwickelt, die eigentlich aus WSFL und XLANG besteht. Nach der Vorstellung von IBM und Microsoft sollte der Standard von Web Services wie auf der Abbildung 6a aussehen.

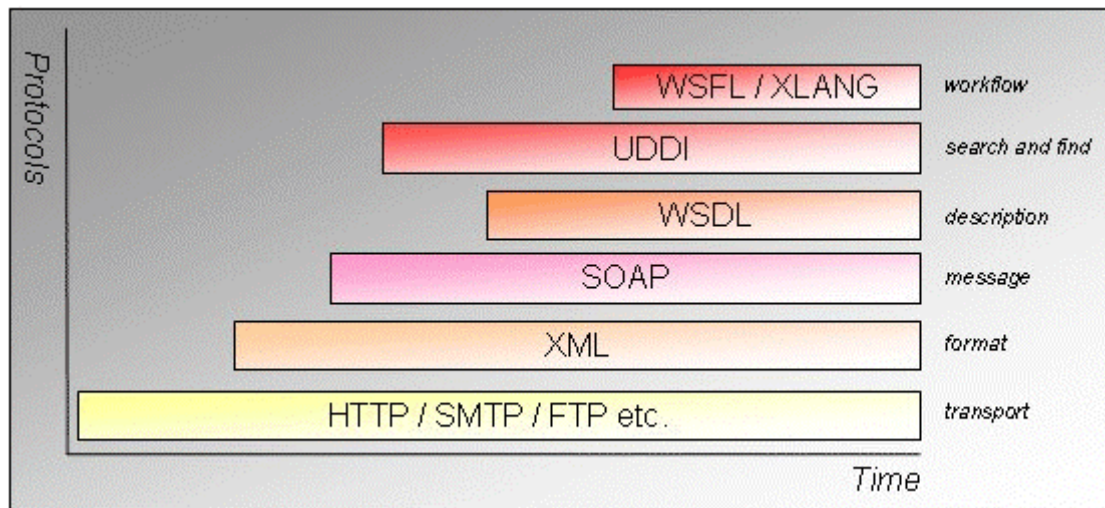


Abbildung 6a: IBM Präsentation [WSA]

6.1 Web Services Flow Language (WSFL)

Die Web Services Flow Language (WSFL) ist ein von IBM vorgeschlagener Standard zur Beschreibung von Geschäftsprozessen auf Basis von Web Services. WSFL basiert auf dem Modell eines gerichteten Graphen. Dabei stellen die Knoten im Graphen Aktivitäten dar, die von Web Services implementiert werden. Die Aktivitäten sind untereinander durch einen Kontrollfluss (durchgezogene Linie) und einen Datenfluss (gestrichelte Linie) verbunden.

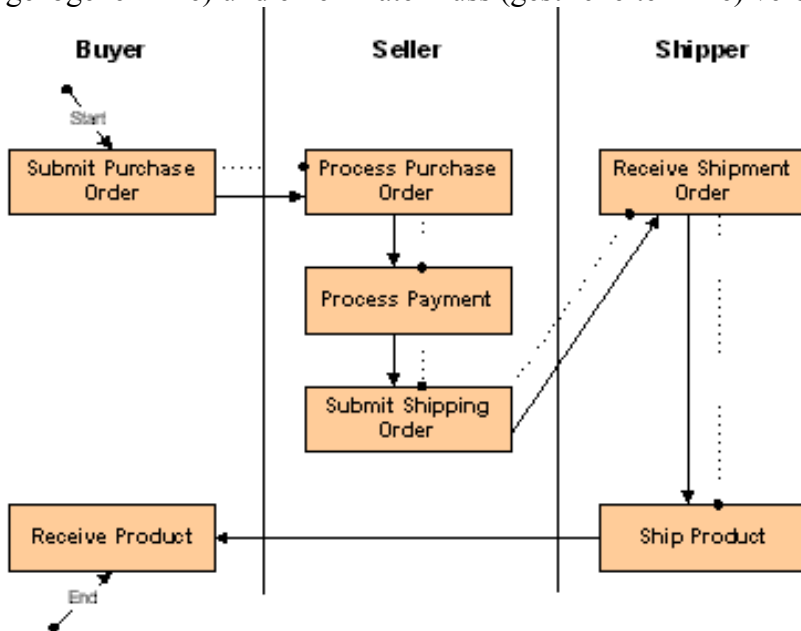


Abbildung 6a: Beispiel Geschäftsprozess-Modell

WSFL erlaubt solche Graphen mit Hilfe von XML zu beschreiben. Eine Workflow-Engine kann die beschriebenen Aktivitäten des in WSFL beschriebenen Geschäftsprozesses abarbeiten. Dabei kann es sich um eine beliebige Workflow-Engine handeln, welche sich an die WSFL Spezifikation hält. WSFL beschreibt hierzu nur die zu verwendenden Protokolle und Standards. Ein mit WSFL erstellter Geschäftsprozess, welcher mehrere Web Services miteinander verbindet, stellt selbst einen Web Service dar und kann entsprechend in einem UDDI Verzeichnis registriert werden.

Codebeispiel 6b zeigt einen Ausschnitt aus dem WSFL Dokument, das den oben gezeigten Beispiel-Prozess modelliert. Beschrieben werden die Aktivitäten „Versenden des Lieferauftrages“ und „Erhalt des Lieferauftrages“ sowie ihre Verbindung miteinander.

Jede der beschriebenen Aktivitäten ist dabei ein durch ein WSDL Dokument beschriebener Web Service, der von einem Service Provider zur Verfügung gestellt wird [WSFL].

WSFL wurde bereits in der Plattform WebSphere von IBM eingebaut (MQSeries Workflow bekannt als WebSphere Process Manager).

```

<flowModel name="totalSupplyFlow" serviceProviderType="totalSupply">

  <serviceProvider name="seller" type="seller" />
  <serviceProvider name="shipper" type="shipper" />

  <activity name="submitShippingOrder">
    <performedBy serviceProvider="seller" />
    <implement>
      <export>
        <target portType="totalSupplyPT" operation="submitShippingOrder" />
      </export>
    </implement>
  </activity>

  <activity name="receiveShippingOrder">
    <performedBy serviceProvider="shipper" />
    <implement>
      <export>
        <target portType="totalSupplyPT" operation="receiveShippingOrder" />
      </export>
    </implement>
  </activity>

  <controlLink source="submitShippingOrder" target="receiveShippingOrder" />
  <dataLink source="submitShippingOrder" target="receiveShippingOrder">
    <map sourceMessage="shippingOrder" targetMessage="shippingOrder" />
  </dataLink>

</flowModel>

```

Codebeispiel 6b [WSFL]

6.2 Web Services for Business Process Design (XLANG)

XLANG kann als Erweiterung von WSDL betrachtet werden. Eine XLANG-Servicebeschreibung entspricht einer WSDL-Servicebeschreibung mit einem Element, welches das Verhalten eines Services als Teil eines Geschäftsprozesses regelt.

Das folgende Script enthält eine XLANG-Servicebeschreibung. Bei den markierten Elementen handelt es sich um die Erweiterungen bezüglich der WSDL Spezifikation.

Codebeispiel aus [XLA]:

```

<?xml version="1.0"?>
<definitions name="StockQuoteProvider"
  targetNamespace="http://example.com/stockquote/provider"
  xmlns:tns="http://example.com/stockquote/provider"
  xmlns:xlang="http://schemas.microsoft.com/biztalk/xlang/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" ... >

  <portType name="RequestReceivePortType">
    <operation name="AskLastTradePrice">
      <input message="..."/>
    </operation>
  </portType>

```

```

<portType name="ResponseSendPortType">
  <operation name="SendLastTradePrice">
    <output message="..."/>
  </operation>
</portType>

<binding name="RequestReceivePortBinding"
  type="tns:RequestReceivePortType">
  <!-- details omitted -->
</binding>

<binding name="ResponseSendPortBinding"
  type="tns:ResponseSendPortType">
  <!-- details omitted -->
</binding>

<service name="StockQuoteProviderService">
  <port name="pGetRequest" binding="tns:RequestReceivePortBinding">
    <soap:address location="mailto:quote@example1.com"/>
  </port>

  <port name="pSendResponse" binding="tns:ResponseSendPortBinding">
    <soap:address location="mailto:response@example2.com"/>
  </port>

```

```

<xlang:behavior>
  <xlang:body>
    <xlang:sequence>
      <xlang:action operation="AskLastTradePrice"
        port="pGetRequest" activation="true"/>
      <xlang:action operation="SendLastTradePrice"
        port="pSendResponse"/>
    </xlang:sequence>
  </xlang:body>
</xlang:behavior>

```

```

</service>
</definitions>

```

XLANG bildet die Basis für eine automatisierte Workflow-Engine, die den Zustand eines Geschäftsprozesses und der beteiligten Web Services überwacht und die Korrektheit des Ablaufes beim Nachrichtenaustausch sichert. Eine solche Workflow-Engine stellt Microsoft mit dem BizTalk Server zu Verfügung, welcher XLANG als Geschäftsprozess-Automatisierungssprache verwendet. Im Moment lassen sich SOAP Web Services nur mit zusätzlicher Anpassung als Teile eines BizTalk Servers nutzen. In dem Vortrag von Roger Riff am 10.7.2003 wird BizTalk eingehend besprochen.

6.3 BPEL4WS (Business Process Execution Language for Web Services)

BPEL4WS ist aus der Zusammenarbeit von Microsoft, IBM und BEA entstanden. Als Grundlage für BPEL4WS dienen die beiden Spezifikationen XLANG und WSFL, welche bereits weiter oben

beschrieben worden sind. Zusammen mit WS-Coordination und WS-Transaction ermöglicht BPEL4WS die Erstellung und Verbindung mehrerer Geschäftsprozesse in einem Web Services Umfeld. Diese Standards unterstützen die Koordination von Geschäftsprozessen und Transaktionen innerhalb einer Firma und über heterogene Systeme hinaus.

Ein Geschäftsprozess beschreibt die Abfolge der Aufgaben, die gemeinsam benutzten Datentypen und die Einbindung anderer Partner. BPEL4WS erlaubt Firmen, ihre Geschäftsprozesse zu beschreiben, welche über mehrere Web Services abgewickelt werden können, und ermöglicht einen standardisierten Nachrichtenaustausch zu internen Services und zwischen Partnern. WS-Coordination und WS-Transaction unterstützen eine verlässliche und beständige Interaktion mehrerer Web Services, unabhängig von der zugrunde liegenden Infrastruktur. Zusätzlich skizzieren sie das Zusammenspiel einer Sammlung von Web Services von verschiedenen Partnern und koordinieren die Ergebnisse der entsprechenden Aktivitäten.

BPEL4WS wird auf mehrere XML Spezifikationen aufgesetzt. WSDL Nachrichten und XML Schema Typdefinitionen liefern das Datenmodell, welches für Prozesse verwendet wird. XPath unterstützt die Datenmanipulation. Alle externen Ressourcen und Partner werden als WSDL Services repräsentiert.

BPEL4WS wird ausführlicher in dem Vortrag von Sergei Chevtsov am 26.6.2003 behandelt.

6.4 Bewertung

Die Auseinandersetzung um einheitliche Standards für den Datenaustausch zwischen Web Services ist vor kurzem eskaliert und Microsoft ist am 25. März 2003 aus der Choreography Group des World Wide Web Consortium (W3C) ausgestiegen.

Als Zeitrahmen hat man ein Jahr für die Entwicklung der Spezifikationen und ein weiteres Jahr für Tests vorgegeben. Mit dem Ausscheren von Microsoft hat sich jedoch die Hoffnung auf industrieweit gültige Vereinbarungen und Techniken für Web Services zunächst zerschlagen. Eine Fragmentierung auf dem Gebiet der Web Services kann zu einer deutlich langsameren Akzeptanz von Standards führen. Ohne einheitliche Standards müssen Anwendungen immer aufs Neue aufeinander abgestimmt werden, was natürlich nur mit entsprechendem Arbeits- und Zeitaufwand zu bewerkstelligen ist.

7. Zusammenfassung

Workflow ist ein Satz von Aufgaben (Tasks, Aktivitäten, Schritten), die zusammen kooperieren, um einen (Geschäfts)Prozess umzusetzen.

Workflow Management System (WFMS) ist ein Software System, das Erzeugung und automatische Durchführung von Workflows unterstützt und koordiniert. Ansätze von WFMS gab es schon in der Büroautomatisierung der 80er Jahre.

Hauptkomponenten eines WFMS sind: Workflow Engine – der Kern, Prozess Definitionstool, Administration und Monitoring Tools, weitere, externe Workflow Engines, Client Applikationen, Invoked Applikationen. Die Workflow Engine kommuniziert mit anderen Komponenten und ist für die Koordination von Workflow zuständig. Es speichert und verwaltet die Workflow Daten typischerweise in eine Datenbank.

Es gibt einen Industrie Standard, von Workflow Management Consortium (WFMC), der die wichtigsten Komponenten und Schnittstellen von WFMS definiert. Insbesondere werden 5 Schnittstellen (API1-5) zwischen der Workflow Engine und den anderen WFMS Komponenten beschrieben.

Zur Modellierung (Spezifikation) von Workflows werden überwiegend Statecharts und Aktivitätscharts verwendet, die Petri-Netze ähnlich sind. Die Statecharts sind in UML Industrie Standard von OMG inkludiert.

Die Architektur von WFMS kann zentralisiert oder verteilt sein, wobei es auch verschiedene Mischformen zwischen den Beiden Architekturen gibt. Welche Architektur am besten ist, ist diskutabel.

Die Implementierung von kommerziellen WFMS beruht (derzeit) auf CORBA (auch ein Standard von OMG) und wird als Java-basiert oder XML-basiert (mit XML Mediator) umgesetzt. Für die Web Services scheint die XML-basierte Architektur besser zu sein, wartet aber noch auf kommerzielle Implementierung.

Was der Standardisierung von Protokollen für Datenaustausch angeht, wurde bisher kein Standard von W3C gewählt. Als wichtigsten Pre-Standard steht WSFL von IBM und XLANG von Microsoft dar (zusammen formen WSFL und XLANG den BPEL4WS).

Über den Standard kann zwar schon zur Zeit meines Vortrags entschieden werden, aber nachdem Microsoft vor kurzem aus der Choreography Gruppe von W3C ausgeschieden ist, halte ich dies für sehr unwahrscheinlich.

8. Literatur

[BEA] BEA Systems, The CORBA Programming Model, <http://edocs.bea.com/wle/techarti/corba.htm>, 2000

[HOL] D. Hollingsworth: WfMC: The Workow Reference Model. Technischer Bericht WFMC-TC-1003 V 1.1, The Workow Management Coalition, November 1994.

[JAB] S. Jablonski, C. Bussler: Workflow Management - Modeling Concepts, Architecture and Implementation, Thomson, 1996

[MSKW] J.A. Miller, A.P. Sheth, K.J. Kochut, X. Wang: CORBA Based Run-Time Architectures for Workflow Management System, Paper von University of Georgia

[OMG95] CORBA: Architecture and Specification. Object Management Group, Inc., 1995.

[OMG98] OMG Submission Workow Management Facility. Technischer Bericht bom/98-06-07, Object Management Group, Inc., Juli 1998.

[SGW] G. Shegalov, M. Gillmann, G. Weikum: XML-enabled Workflow Management for E-Services across Heterogeneous Platforms, Paper von University of the Saarland, Germany, 2001

[WEI] Gerhard Weikum: Workflow Technology, Paper von University of the Saarland, Germany

[WFMC] Workflow Management Coalition, <http://www.wfmc.org>

[WFMC98] Workflow Management Coalition (WFMC): Workflow and Internet: Catalysts for Radical Change. White Paper, June 1998. <http://www.wfmc.org>

[WFM98a] Workow Client Application Programming Interface (WAPI) Naming Conventions V 2.0. Technischer Bericht WFMC-TC-1009, WFMC, 1998

[WFM98b] WFMC:Interoperability e-mail MIME Binding V 1.1. Technischer Bericht WFMC-TC-1018, The Workow Management Coalition, Oktober 1998

[WFM98c] WFMC: Process Definition Interchange V 1.0 Final. Technischer Bericht WFMC-TC-1016-P, The Workow Management Coalition, November 1998

[WFM98d] WFMC: Reference Model Workow Interoperability, the Key to E-Commerce and to Process Scalability, Oktober 1998. <http://www.aiim.org/wfmc/if4article.pdf>

[WFM99] Workflow Management Coalition (WFMC): Terminology & Glossary. Technischer Bericht WFMC-TC-1011, The Workow Management Coalition, Februar 1999.

[WSA] Sander Duivestijn: Web Services and Workflow, Web Services Architect, September 2001, <http://www.webservicesarchitect.com/>

[WSFL] <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>

[XLA] http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm

Anhang A: Glossar

API: (Application Programming Interface)

Schnittstelle für Anwendungsprogrammierung. Eine API ist eine Gruppe von Routinen, die von einem Programm aufgerufen werden können, um bestimmte Aufgaben zu erledigen.

BPEL4WS (Business Process Execution Language for Web Services)

CORBA: (Common Object Request Broker Architecture)

Spezifikation, die die Kommunikation von Objekten (Programmteilen) gleicher oder unterschiedlicher Programme beschreibt und 1991 für den Einsatz in objektorientierten Umgebungen entwickelt wurde.

DDL: (Database Description Language), Datenbank-Beschreibungssprache.

DSI (Distributed Statechart Interpreter), DSI Server – ein CORBA Applikation Server

IDL (Interface Definition Language), Verwendet für Definition von CORBA Schnittstelle

IOP (Internet Inter-ORB Protocol), CORBA-orientierter IOP Internet Protokol

Middleware: Software mit Schnittstellencharakter, die als Zwischenschicht zwischen Server- und Client-Komponenten eines verteilten Systems sitzt. Für den Anwender ist sie in der Regel unsichtbar, wenn verschiedene Anwendungen, Computer- oder Betriebssysteme z. B. mit Servertechniken verbunden werden.

ORB (Object Request Broker)

OMG: (Object Management Group)

Die Object Management Group hat sich zum Ziel gesetzt, die Verbreitung der Objekttechnologie zu fördern und hat mit CORBA1 eine Referenzarchitektur [OMG95] vorgestellt. CORBA stellt eine plattform- und sprachunabhängige Infrastruktur für verteilte Objekte bereit.

RPC: (Remote Procedure Call) Programm- oder Prozeduraufruf über ein Netzwerk.

WFMC: (Workflow Management Coalition)

Non-Profit Interessenverband der von Workflow-Management-System Anbietern und Anwendern gegründet wurde. Ziel der WFMC ist es, einen Standard im Bereich Workflowsysteme zu etablieren, der die Interoperabilität zwischen Produkten verschiedener Hersteller ermöglicht und die Verbreitung der Workflow Technologie fördern soll [WFMC].

WPDL: (Workflow Process Definition Language)

Die WPDL soll die standardisierte Formulierung von Arbeitsabläufe erlauben

WSDL: (Web Services Definition Language)

WSFL: (Web Services Flow Language) entwickelt von IBM

XLANG: (Web Services for Business Process Design) entwickelt von Microsoft, basiert auf WSDL