Proposal zur Diplomarbeit

Konzeption und Realisierung einer verteilten Workflow-Steuerung für mobile Endgeräte

Christoph Haase

Universität Dortmund
Fachbereich Informatik
Lehrstuhl für Softwaretechnologie

Version 2.0
Dortmund, 18.03.2003

Inhaltsverzeichnis

1. Einleitung

Die Diplomarbeit ist aus dem Kontext des Forschungsprojekts Mobile Spedition im Web (SpiW) hervorgegangen. Ein Ziel dieses Projekts ist die Kommunikation zwischen Spediteuren, Disponenten und Fahrern zu verbessern. Dazu werden bei den Fahrern mobile Endgeräte eingesetzt, die mittels mobiler Datenübertragungstechniken (z.B. GSM, GPRS, UMTS) mit der Zentrale der Spedition kommunizieren.

Mit Hilfe dieser Kommunikationstechniken soll die rechnergestützte Steuerung unterschiedlicher Arbeitsvorgänge ermöglicht werden. Wenn beispielsweise ein Kunde die Annahme einer Lieferung verweigert, dann meldet der Fahrer das über seinen mobilen Client dem Disponenten. Dieser erzeugt dann einen neuen Auftrag mit dem neuen Zielort der abgelehnten Lieferung und überträgt das an den mobilen Client des Fahrers. Dieser muss den neuen Auftrag bestätigen.

Es ist nicht sinnvoll für jeden einzelnen Arbeitsvorgang eine eigene Routine („hard-wired“) zu erstellen, die die Steuerung des Vorgangs vornimmt. Dieses Verfahren wäre viel zu unflexibel, da die Entwicklung von Vorgängen evolutionär ist. So verändern sich mit der Zeit Arbeitsvorgänge oder es entstehen ganz neue Arbeitsvorgänge. Eine jeweilige Anpassung der Routine wäre viel zu aufwändig. Außerdem sollten Arbeitsvorgänge auch variiert ausgeführt werden können, denn beispielsweise benötigt ein erfahrener Anwender weniger Unterstützung als ein Anfänger. Auch sind vom Vorgangs-modellierer nicht unbedingt Programmiersprachenkenntnisse zu verlangen.

Deshalb ist die Verwendung eines System zur flexiblen Steuerung von Arbeitsvorgängen sinnvoll. Ein solches System wird als Workflow-Management-System (WfMS) bezeichnet und von ihm unterstützte Arbeitsvorgänge werden Workflows genannt. Ein WfMS stellt eine Sprache zur Verfügung, mit der Arbeitsvorgänge modelliert werden können. Auf Grundlage eines solchen Modells übernimmt das WfMS die Steuerung und Ausführung des Arbeitsvorgangs.

In der Diplomarbeit ist ein Konzept für ein WfMS zu entwickeln, das die Steuerung mobiler Clients ermöglicht. Konzepte für WfMS sind in der wissenschaftlichen Literatur bereits zahlreich entwickelt worden. Die Besonderheit dieser Diplomarbeit resultiert also aus den Restriktionen der Anbindung der mobilen Clients:

- Es besteht nicht ständig eine Kommunikationsverbindung zwischen Client und Server. D.h. es sind Mechanismen zu entwickeln, die auch im Offline-Betrieb die Ausführung bzw. Teilausführung von Workflows ermöglichen.
  Unter diesen Aspekt ist insbesondere die Verteilung der Workflow-Ausführung zu analysieren. D.h. es sollte untersucht werden, ob eine client-seitige Steuerung einer server-seitigen Steuerung (wie es in den meisten herkömmlichen WfMS angewendet wird) oder eine Kombination aus beiden vorzuziehen ist.
- Außerdem sind Kommunikationskosten bei mobilen Übertragungstechniken vergleichsweise hoch und die Übertragungsraten gering, so dass eine Minimierung der Kommunikation zwischen den Clients und dem Server erreicht werden sollte.
- Der Lösung zur Reduzierung der Kommunikationskosten, möglichst viele Daten auf den Clients unterzubringen, stehen die geringen Ressourcen (insbesondere bzgl. Speicherkapazität) der mobilen Clients entgegen. Es gilt hier, einen sinnvollen Kompromiss zu finden.

2. Einordnung in die Literatur

Mit dem Thema Workflow-Management beschäftigt sich die wissenschaftliche Literatur bereits seit Jahren. Entsprechend vielfältig sind auch die Definitionen der verwendeten Begriffe. Deshalb ist der Versuch der Workflow Management Coalition (WfMC), eine Vereinigung von Workflow-Management-System-Herstellern und -Anwendern, besonders sinnvoll, eine einheitliche Definition der Begriffe in einem Glossar vorzunehmen ([WMC99]).

Das Ziel der Diplomarbeit ist die Konzeption eines WfMS. Dazu ist zunächst die Kenntnis über Komponenten eines WfMS sinnvoll. In [WMC95] und [JBS99] werden zwei unterschiedliche Modelle vorgestellt.

In [WMC95] entwickelt die WfMC ein Referenzmodell eines Workflow-Management-Systems. Dieses Referenzmodell besteht aus folgenden Komponenten: Workflow Enactment Service (Workflow-Engines), Process Definition Tools, Administration & Monitoring Tools, Workflow Client Applications (Worklist Handler) und Invoked Applications.
Zur Kommunikation des Workflow Enactment Services mit den anderen Komponenten bzw. mit anderen Workflow Enactment Services sind Schnittstellen definiert. Beispielsweise ist in [WMC02] für den Austausch des mittels der Process Definition Tools modellierten Workflows mit dem Workflow Enactment Service die XML-basierte Prozessdefinitionssprache XPDL spezifiziert worden. Oder in [WMC01] wird der Austausch zwischen Workflow Enactment Services mittels XML spezifiziert. Die WfMC verfolgt mit der Standardisierung der Schnittstellen das Ziel, dass Workflow-Management-Systeme unterschiedlichster Hersteller miteinander kommunizieren können.
Die Object Management Group (OMG) hat anlehnend an das Referenzmodell der Workflow Management Coalition die CORBA-basierte Workflow Management Facility ([OMG00]) spezifiziert. In dieser Spezifikation werden ein Objektmodell und die Schnittstellen der Objekte konstruiert, über die Laufzeitaktivitäten des Referenzmodells realisiert werden. D.h. die Workflow-Modellierung ist nicht Bestandteil der Workflow Management Facility.

In [JBS99] und [Sch97] wird ein aspektbezogener Aufbau eines Workflow-Management-Systems propagiert. Dabei wird das WfMS in folgende Komponenten unterteilt: Funktionsaspekt, Verhaltensaspekt, Informationsaspekt, Organisationsaspekt und Operationsaspekt. Der Funktionsaspekt beschreibt den strukturellen Aufbau eines Ablaufs. Der Verhaltensaspekt beschreibt die zeitliche und logische Reihenfolge eines Ablaufs. Der Informationsaspekt beschreibt den Datenfluss in Abläufen. Der Organisationsaspekt umfasst die Aufbauorganisation und Zuordnung der ausführenden Agenten zu den Abläufen. Der Operationsaspekt ist für die Einbindung der im Ablauf verwendeten Applikationen zuständig. Diese Aspekte bilden zusammen mit einer Koordinationskomponente das Workflow-Management-Kernsystem. Dieses Kernsystem wird durch ein Werkzeug-System ergänzt, das beispielsweise eine Arbeitsliste oder einen Übersetzer enthält. Eine Erweiterung um zusätzliche Aspekte ist explizit vorgesehen.

Ein Schwerpunkt der Diplomarbeit wird die Betrachtung der Verteilung der Workflow-Steuerung sein. Diesen Betrachtungen werden die in [Sch97] und [BD99] vorgestellten Konzepte der Verteilung der Ausführung von Workflows auf verschiedene Server zugrunde liegen. Diese Konzepte werden insbesondere um Überlegungen hinsichtlich der Besonderheiten von mobilen Clients ergänzt.

In [Sch97] werden vier verschiedene Kategorien der Verteilung der Workflow-Steuerung vorgestellt: zentraler Server, Workflow-Instanz-Migration, Workflow-Partitionierung und Subworkflow-Verteilung. In [BD99] wird unterschieden zwischen WfMS mit zentralem Server, WfMS mit mehreren Servern und voll verteilte WfMS. WfMS mit mehreren Servern werden noch unterteilt in WfMS mit zufällig gewählten Servern, WfMS mit Servern nahe bei Bearbeitern und Servern nahe der Anwendung.
In [BRD01] werden Lösungen diskutiert, wie möglichst effizient Workflow-Daten zwischen den Servern ausgetauscht werden können.

Eine aus der Verwendung mobiler Clients resultierende Restriktion der Entwicklung eines WfMS ist, dass nicht immer eine Kommunikationsverbindung zum Server besteht. In [AG95] wird eine Lösung zur Handhabung dieses Problems vorgestellt und auf das WfMS FlowMark von IBM angewendet . Es wird ein Mechanismus entwickelt, bei dem der Benutzer die Aktivitäten eines Workflows, die er Offline durchführen möchte, auswählt. Dann werden alle erforderlichen Daten dieser Aktivitäten auf den Client übertragen. Diese Aktivitäten werden im WfMS gesperrt, so dass diese nicht von anderen Benutzern parallel ausgeführt werden können. Im Offline-Modus werden dann die Aktivitäten auf dem Client bearbeitet werden. Wenn der Client wieder mit dem Server verbunden ist, erfolgt ein Abgleich der Daten.

3. Aufgabenstellung

Die Aufgabe der Diplomarbeit ist die Realisierung eines verteilten Workflow-Management-Systems. Bezogen auf das Referenzmodells der WfMC wird der Schwerpunkt auf der Entwicklung des Workflow Enactment Service (Workflow-Engines) liegen. D.h. die Administration & Monitoring Tools werden komplett vernachlässigt und die Process Definition Tools werden durch eine Standardsoftware (z.B. Microsoft Visio oder Leu Smart von adesso) realisiert.

Zum Austausch der Informationen zwischen dem Process Definition Tool und dem Workflow-Enactment-Service ist eine Workflow-Sprache zu definieren.
Diese Sprache sollte mächtig genug sein, um Reihenfolgen, Verzweigungen und Zusammenführungen von Vorgängen zu realisieren. Es sollten Datenflüsse unterstützt und Bedingungen für Ausführungen von Vorgängen modelliert werden können.
Diese Workflow-Sprache muss von der Workflow-Engine (Workflow Enactment Service) interpretiert und der modellierte Workflow ausgeführt werden können.

Eine weitere Funktionalität des WfMS ist die Möglichkeit der verteilten Ausführung der Workflows. Ob nur einzelne Aktivitäten oder ganze (Sub-) Workflows verteilt werden sollen und ob eine server- oder eine clientseitige Steuerung des Workflows oder eine Kombination aus beiden sinnvoll ist, ist Untersuchungsgegenstand der Diplomarbeit.

Das WfMS muss die Ansteuerung der Komponenten des SPIW-Projekts ermöglichen. Die Komponenten werden mit Hilfe des Microsoft .NET-Frameworks erstellt, so dass ein Zugriff auf die .NET-Klassen wünschenswert ist.

Außerdem sollte eine flexible Ausführung der Workflows ermöglicht werden. So soll beispielsweise der Anfänger Schritt für Schritt durch den Ablauf geführt werden, währenddessen der erfahrene

Anwender selbst entscheiden kann, welche Schritte er zuerst ausführt. Für diese Flexibilität sind Profile, wie z.B. Benutzer- oder Standortprofile, zu entwerfen, die die Ausführung der Workflows beeinflussen.

Das WfMS sollte dem Anwender eine Aufgabenliste zur Verfügung stellen, aus der er die nächsten zu erledigenden Aufgaben entnehmen kann. Diese Aufgabenliste sollte dem Benutzer ggf. Hilfetexte für die auszuführenden Aktivitäten zur Verfügung stellen.

4. Lösungsansatz

In der Diplomarbeit sind zunächst die theoretischen Grundlagen zu schaffen. Dafür wird die Definition grundlegender Begriffe vorgenommen und Konzepte des Workflow-Managements vorgestellt. Im Focus dieser Betrachtungen wird insbesondere auch die Referenzarchitektur der WfMC stehen.

Die Entwicklung des WfMS beginnt ausgehend von der Anforderungsanalyse, in der die funktionalen Anforderungen an das zu konzipierende WfMS aus den zu unterstützenden Workflows abgeleitet werden. Das setzt zunächst eine Kategorisierung der zu unterstützenden Workflows voraus.
Die Ermittlung der nicht-funktionalen Anforderungen basieren auf allgemeinen Anforderungen an WfMS, wie sie beispielsweise in [JBS99] zu finden sind. Diese allgemeinen Betrachtungen werden ergänzt durch Betrachtungen der Besonderheiten, die durch den Einsatz mobiler Endgeräte entstehen. Beispielhaft sind hier die Ressourceneinschränkung der mobilen Endgeräte oder die hohen Kommunikationskosten zu nennen.
Eine Einteilung in Muss- und Kann-Anforderungen ist für die Auswahl der verwendeten Technologien hilfreich. Denn durch die Muss-Anforderungen können Lösungsalternativen ausgeschlossen werden.

Ausgehend von diesen Anforderungen ist ein Konzept der Verteilung der Workflow-Ausführung zu entwickeln. Es gilt zu untersuchen, ob eine client-seitige oder server-seitige Steuerung der Workflows oder eine Kombination aus beiden am Besten den Anforderungen genügt.

Die Entwicklung einer Workflow-Sprache ist ebenfalls ein wesentlicher Bestandteil der Diplomarbeit. Dabei sind zwei Ebenen zu berücksichtigen: Es sollte eine graphische Modellierung der Workflows ermöglicht werden. Dieses graphische Model sollte dann in eine textuelle Sprache umgewandelt werden, die von der Workflow-Engine interpretiert werden kann. Die Definition der Workflow-Sprache hängt entscheidend von der Auswahl der Lösungsalternative ab.

Schließlich ist ausgehend von den Anforderungen und dem Konzept der Verteilung der Workflow-Ausführung eine Workflow-Engine zu entwickeln. Zur Realisierung werden in der Diplomarbeit zwei Alternativen betrachtet:

1. Microsoft Workflow Designer
2. Eigenentwicklung einer Workflow-Steuerung.

In den folgenden Kapiteln (Kapitel 4.1 und Kapitel 4.2) werden kurz die Konzepte der beiden Alternativen dargestellt und die Möglichkeit der Umsetzung wesentlicher allgemeiner Anforderungen diskutiert. Zu diesen allgemeinen Anforderungen, die bereits in Kapitel 3 angeführt wurden, gehören die Mächtigkeit der Workflow-Sprache, Möglichkeiten der Unterstützung der Verteilung, Umsetzung der Ansteuerung externer Schnittstellen und Offenheit des Systems.

Die folgende Konzeptions- und der Entwurfsphase umfasst die Entwicklung eines Klassenmodells der Workflow-Engine und eines Klassenmodells für den Import des Workflow-Modells. Ebenfalls sollte die Ansteuerung der SPIW-Komponenten und die Realisierung der Verteilung der Workflow-Ausführung konzipiert werden. Schließlich ist ein Datenbankschema zur Speicherung aller Workflow-relevanten Daten zu entwickeln. Das Ziel dieser Entwurfsphase ist die Erstellung der Systemarchitektur.

Anschließend ist die (prototypische) Implementierung der Workflow-Steuerung vorzunehmen, die mit einer Analyse der Implementierungsphase abschließt.

4.1. Microsoft Workflow Designer

Sowohl der Microsoft Exchange Server als auch der Microsoft SQL Server bieten dem Entwickler Workflow-Dienste an. Diese Dienste können über den MS Workflow Designer genutzt werden.

Der den beiden Server-Lösungen zu Grunde liegende Mechanismus ist der gleiche: Es werden die Elemente „*Item*“, „*State*“, „*Transition*“ und „*Event*“ verwendet (siehe [MWY00]). Ein *Item* ist ein Element, das in der Workflow-Ausführung bearbeitet werden soll. Ein *Item* befindet sich immer in einem *State*. Ein *State* hat mindestens zwei *Transitions*: eine *Transition*, die zu dem *State* führt, und eine *Transition,* die das *Item* in einen neuen *State* überführt. Eine *Transition* wird ausgeführt, wenn ein *Event* ausgelöst worden ist und die zur *Transition* gehörige Bedingung erfüllt ist. Ein *Event* wiederum wird ausgelöst, wenn ein *Item* verändert wurde.
Beim Exchange Server ist das *Item* ein Dokument (z.B. ein Word-Dokument oder eine Textdatei), währenddessen beim SQL Server das *Item* ein Tabelleneintrag ist.

Deshalb kann die Lösung basierend auf dem Exchange Server als dokumentenorientierte Workflow-Steuerung bezeichnet werden (vgl. Kategorisierung von WfMS in [Ro96]).

Der Workflow-Mechanismus des MS SQL Server basiert auf dem Konzept aktiver Datenbanken (vgl. [EG96]). Dabei werden Einfügen-, Löschen- und Ändern-Trigger verwendet. Eine Änderung in der dem Workflow zugrunde liegenden Tabelle führt zur Ausführung des Triggers, der wiederum eine Stored Procedure aufruft. Diese Stored Procedure setzen die Anweisungen um, die vom Workflow-Modellierer in Ereignisskripten definiert wurden.

Die Elemente beider Lösungen und damit auch die Möglichkeiten der umzusetzenden Workflows sind identisch, so dass die Betrachtung eines Systems ausreichend ist. Die Installation des Exchange Servers war wesentlich umständlicher und ressourcenaufwändiger. Deshalb wird im Folgenden die Lösung auf Basis des Microsoft SQL Servers betrachtet.

4.1.1. Workflow-Sprache

Die Workflow-Sprache besteht aus den Elementen Ereignis (Event), Status (State), Transition und Zugriffsrecht (Permission).

Ein *Ereignis* wird ausgelöst, wenn Änderungen in der Tabelle (Erstellen, Löschen oder Ändern eines Datensatzes) vorgenommen werden oder wenn ein Zeitlimit abläuft. Mit Auslösung eines Events ist in

der Regel ein Statuswechsel verbunden. Dann werden folgende Ereignis-Skripte, die vom Workflow-Modellierer angepasst werden können, aufgerufen:

1. bisheriger Status: OnExit
2. Transition von bisheriger Status nach neuer Status
3. neuer Status: OnEnter.

Jedes dieser Ereignisskripte hat ein zugehöriges Validierungsskript, das entscheidet ob, das Ereignis-Skript überhaupt ausgeführt werden darf. Auch dieses kann vom Workflow-Modellierer angepasst werden.

Ein *Status* repräsentiert den Zustand des Workflows. Der Workflow beginnt mit dem Status „Element erstellt“ und endet mit dem Zustand „Element gelöscht“. Im Workflow-Diagramm wird ein Status durch ein Rechteck symbolisiert. Jedem Status werden zwei Ereignis-Skripte zugeordnet: das OnEnter-Skript und das OnExit-Skript mit jeweils den Validierungsskripten.

Eine *Transition* (auch als Aktion bezeichnet) überführt den aktuellen Zustand eines Workflows in einen neuen Zustand. Im Workflow-Diagramm wird eine Transition durch einen Pfeil dargestellt. Jeder Transition wird ein Ereignis-Skript OnCreate zugeordnet, in dem die Aktionen zum Statuswechsel vom Workflow-Modellierer implementiert werden können. Auch diesem Skript ist ein Validierungsskript zugeordnet, das auswertet, ob die Aktion überhaupt durchgeführt werden soll.

Mit Hilfe von *Zugriffsrechten* kann die Ausführung von Transitionen nur durch bestimmte Benutzer erlaubt werden. Dazu wird das im MS SQL Server verwendete Rollenkonzept in den Workflow-Mechanismus integriert.

Die Workflow-Engine hat dann folgende Funktionen:

1. Überprüfung der Gültigkeit der Änderung: findet ein Wechsel in einen gültigen Status statt?
2. Überprüfung der Zugriffsrechte: ist der Benutzer befugt zur Ausführung der Transition?
3. Evaluierung des Validierungsskripts und ggf. Ausführung des Skripts

4.1.2. Vorgehensweise zur Entwicklung einer Workflow-Anwendung

Die Entwicklung einer Anwendung mit Hilfe des Microsoft Workflow Designers erfolgt in drei Schritten:

1. Erstellung einer Datenbank
2. Erstellung eines Workflows
3. Erstellung einer Benutzeroberfläche

Zunächst muss eine SQL Server Datenbank und eine Tabelle erstellt werden, in die die Workflow relevanten Daten gespeichert werden. Der zu entwickelnde Workflow beginnt mit Einfügen eines neuen Eintrags in diese Tabelle und endet mit dem Löschen dieses Eintrags.
Anschließend sind die Rollen anzulegen, die in dem zu definierenden Workflow benötigt werden und diesen Rollen sind die beteiligten Anwender zuzuordnen. Alle diese Einstellungen werden innerhalb des MS SQL Servers vorgenommen (beispielsweise durch den „Enterprise Manager“).

Mit Hilfe des Microsoft Workflow Designers wird dann der Workflow modelliert. Dazu werden zunächst die erforderlichen Stati eingefügt, die dann mittels Transitionen verbunden werden. Den

Transitionen werden dann die Rollen zugeordnet, die zur Ausführung berechtigt sind. Schließlich werden in den Ereignisskripten die erforderlichen Anweisungen implementiert.

Es muss anschließend eine Benutzeroberfläche erstellt werden, die den Benutzer durch den Ablauf des Workflows führt. Diese kann mit jeder beliebigen Entwicklungsumgebung erstellt werden, die den Zugriff auf eine MS SQL Server Datenbank ermöglicht. Microsoft bietet eine einfache Realisierung in Form von Data Access Pages an. Dabei handelt es sich um HTML-Seiten, die die Dateneingabe eines Datensatzes ermöglicht. Die Steuerung des Workflows wird durch Verwendung der ActiveX-Komponente „Workflow-Toolbar“ ermöglicht. Die HTML-Seiten werden dem Workflow-Anwender über einen Internet-Server zur Verfügung gestellt.

4.1.3. Diskussion: Realisierung allgemeiner Anforderungen

Die vom Workflow Designer angebotene Workflow-Sprache ist mächtig genug Reihenfolgen, Verzweigungen und Zusammenführungen zu realisieren. Problematisch erscheint aber die Umsetzung alternativer Verzweigungen. Diese können zwar durch den Ausgang mehrerer Transitionen aus einem Status und entsprechende Auswertungen in den Validierungsskripten realisiert werden. Aber eine Auswertung eines Validierungsskripts erfolgt erst unmittelbar vor Ausführung der Transition, so dass nicht vorher entschieden werden kann, welche Transitionen gültig sind. Es kann erst im Nachhinein entschieden werden, dass ein Transition gültig bzw. ungültig war.
Ein Rollenkonzept wird vom Workflow Designer ebenfalls umfassend unterstützt. Der Datenfluss wird zwar nicht explizit modelliert, aber durch die eines Workflows zu Grunde liegende Tabelle wird dieser ausreichend gewährleistet.

Die Workflow-Engine des SQL Servers ist eine zentralisierte Lösung. Die Ereignis- und Validierungsskripte werden alle auf dem Server ausgeführt und dem Benutzer standardmäßig über eine auf HTML basierende Benutzeroberfläche angezeigt. Das heißt, es ist nicht vorgesehen eine Steuerung von (Sub-) Workflows auf Client-Seite zu ermöglichen. Eine verteilte Steuerung von Workflows müsste also durch einen Workaround implementiert werden, dessen Realisierung als nicht unproblematisch angesehen werden muss.

Die Ansteuerung externer Komponenten kann innerhalb der Ereignisskripte über Funktionsaufrufe einer DLL oder durch Automatisierung von COM-Komponenten realisiert werden. D.h. die SPIW-Komponenten müssten über eine der beiden Zugriffsformen zur Verfügung gestellt werden. Das würde weiteren Aufwand verursachen, da die SPIW-Komponenten nur als .NET-Klassen zur Verfügung stehen.

Der MS Workflow Designer ermöglicht keinen Import bzw. Export von Workflow-Modellen in ein anderes Format. So existiert beispielsweise keine Möglichkeit einen Workflow in ein XML-Format zu exportieren, genauso wenig wie ein Import einer XML-Datei ermöglicht wird. D.h. mit Erstellung eines Modells durch den MS Workflow Designer findet gleichzeitig auch eine Festlegung desselben als Workflow-Engine statt. Eine Entkopplung der Komponenten Process Definition Tools und Workflow Enactment Service, wie im Referenzmodell der WfMC gefordert ([WMC95]), findet nicht statt.
Es besteht aber die Möglichkeit einen modellierten Workflow auf andere SQL Server zu transferieren.

4.2. Eigenentwicklung einer Workflow-Steuerung

Die Komponenten des Projekts SPIW werden auf Basis des Microsoft .NET-Frameworks entwickelt. Auf den mobilen Clients wird das .NET Compact Framework eingesetzt.
Aus diesem Grunde wird die Workflow-Steuerung ebenfalls mit der .NET-Technologie realisiert. Dafür wird die Programmiersprache C# in der Entwicklungsumgebung Microsoft Visual Studio .NET verwendet.

4.2.1. Auswahl eines Modellierungstools

Der Focus der Diplomarbeit ist die Entwicklung einer Workflow-Steuerung. Deshalb wird keine Entwicklung eines Modellierungstools angestrebt, sondern ein externes Modellierungstool verwendet.

Ein Modellierungstool muss folgende grundlegende Anforderungen unterstützen: Es muss die Modellierung der Workflow-Sprachelemente Aktivität, Reihenfolgen, alternative und parallele Verzweigungen, Zusammenführungen, Subprozesse und Datenflüsse erlauben. Außerdem ist die Definition von Eigenschaften zu den Elementen erforderlich. So sollten beispielsweise Hilfetexte zur Anwenderführung angegeben werden können. Ebenso ist eine Exportmöglichkeit des Modells in ein offenes Format, wie z.B. XML, erforderlich.

Als erste Alternative wurde die Standard-Visualisierungsanwendung Microsoft Visio betrachtet. Durch unterschiedliche Schablonen ermöglicht Visio dem Anwender Modelle verschiedenster Art zu erstellen. Diese Schablonen können durch den Anwender verändert werden. Dadurch ist es kein Problem mit Visio die erforderlichen Workflow-Sprachelemente zu modellieren.
Die Definition von Eigenschaften von Elementen wird von Visio ermöglicht durch benutzerdefinierte Eigenschaften.
Ein Export in ein XML-Format wird ebenfalls zur Verfügung gestellt. Allerdings ist dieses Format auf den visuellen Aufbau des Modells ausgerichtet. D.h. die XML-Datei enthält sehr viele Informationen über Positionen und Aussehen der modellierten Elemente. Sie ist damit sehr unübersichtlich und es ist sehr mühsam die Workflow-relevanten Daten herauszufiltern. Viel problematischer ist aber, dass Verbindungen zwischen Elementen nur über die Positionen der Elemente identifiziert werden können. Denn es werden lediglich die Positionen der Elemente und der Pfeile angegeben und nicht etwa die Information, dass die Elemente X und Y über einen Pfeil miteinander verbunden sind. Dadurch ist der Aufwand zur Identifizierung von Verbindungen zwischen Elementen sehr hoch.

Die zweite betrachtete Alternative ist das Prozessmodellierungs- und Prozesssimulationswerkzeug Leu Smart, welches auf einer speziellen Art von Petri-Netzen, sog. FunSoft-Netze basiert.
Auch Leu Smart ermöglicht alle erforderlichen Workflow-Sprachelemente zu modellieren. Aktivitäten können in Leu Smart durch das gleichnamige Modellierungselement modelliert werden.
Die Darstellung von Datenflüssen wird durch Informationsspeicher ermöglicht. Reihenfolgen werden durch Pfeile zwischen den Informationsspeichern und den Aktivitäten modelliert. Hierbei ist zu beachten, dass einem Informationsspeicher immer eine Aktivität und einer Aktivität immer ein Informationsspeicher folgen muss.
Verzweigungen können dadurch modelliert werden, dass einer Aktivität mehrere Informationsspeicher folgen. Durch Angabe des Ausgangsschaltverhaltens, das von Leu Smart für Simulationszwecke benötigt wird, kann angegeben werden, ob der Kontrollfluss alternativ oder parallel verzweigen soll.

Subworkflows können durch das Element der Verfeinerung von Leu Smart oder durch Anbindung von Ablaufmodellen zu einer Aktivität realisiert werden.
Weitere Eigenschaften der Workflow-Elemente können über das Beschreibungsfeld oder das Dokumentationsfeld angegeben werden.
Leu Smart speichert die Modelle in XML-Dateien ab, die schließlich von der Workflow-Engine verwendet werden können. Auch diese XML-Dateien enthalten viele Informationen, die für die Workflow-Ausführung nicht relevant sind, wie beispielsweise die Positionen der Elemente. Allerdings sind die Dateien wesentlich übersichtlicher und kleiner als bei Visio. Der große Vorteil gegenüber Visio ist aber, dass Verbindungen zwischen Workflow-Elementen einfach ermittelt werden können. So sind innerhalb des XML-Tags, das Aktivitäten definiert (<agency> ... </agency>), Listen von Vorgänger- und Nachfolger-Elementen enthalten. Außerdem finden sich innerhalb des Tags Informationen zu Verzweigungen (parallel oder alternativ).

Da durch Leu Smart alle Anforderungen relativ einfach (im Vergleich zu Visio) erfüllt werden können, wird Leu Smart als Workflow Definition Tool verwendet.

4.2.2. Import der XML-Dateien

Die von Leu Smart generierten XML-Dateien müssen von der Workflow-Engine importiert werden. Dazu muss ein XML-Parser entwickelt werden, der die Workflow-relevanten Informationen aus der XML-Datei liest und sie in entsprechende Objekte umwandelt.
Es gibt verschiedene Techniken mit denen XML-Dateien geparst werden. Die populärsten sind dabei SAX und DOM.
Bei DOM wird die komplette Datei eingelesen und ein entsprechender Baum aufgebaut. Das ermöglicht den willkürlichen Zugriff auf alle Tags der XML-Datei und ist damit recht einfach anzuwenden. Der Nachteil ist aber, dass die komplette Datei eingelesen werden muss und damit die Vorgehensweise sehr ressourcenaufwändig ist.
Bei SAX wird die XML-Datei sequentiell abgearbeitet. Dadurch sind keine willkürlichen Zugriffe auf die Elemente der Struktur möglich. Auf ein bereits gelesenen Element kann im nachhinein nicht mehr zugegriffen werden. Dieser Ansatz hat den Vorteil, dass nicht die komplette Datei eingelesen werden muss und ist damit recht ressourcensparend.
Das .NET-Framework bietet Parser-Klassen über den Namespace „System.Xml“ für DOM und für eine abgewandelte Form von SAX (siehe [Sk01]). Gleiches gilt für das auf den Clients eingesetztem .NET Compact Framework (siehe [Ro02]). Die Realisierung eines Imports, der aus den von Leu Smart generierten Dateien die Workflow-relevanten Informationen heraus parst sollte also kein großes Problem darstellen.

4.2.3. Diskussion: Realisierung allgemeiner Anforderungen

Die durch Leu Smart vorgegebene Workflow-Sprache ist, wie oben ausgeführt, mächtig genug alle wesentlichen Elemente zu unterstützen.

Aufgrund der großen Flexibilität einer Eigenentwicklung können alle Varianten der Verteilung der Workflow-Ausführung (server-, client-seitige Steuerung oder Kombination aus beiden) realisiert werden. Zur Kommunikation den zwischen verteilten Systemen wird ein Action-Event-Mechanismus verwendet, der in der zeitgleich erstellten Diplomarbeit [Gö03] entwickelt wird.

Mit der Eigenentwicklung kann die Ansteuerung beliebiger externer Schnittstellen (der SPIW-Komponenten) realisiert werden. Im Gegensatz zum MS Workflow Designer kann unmittelbar auf .NET-Klassen zugegriffen werden, da die Eigenentwicklung ebenfalls auf .NET basiert.

Das zu entwickelnde System ist zwar durch Verwendung von XML-Dateien offener als der MS Workflow Designer. Aber es werden lediglich von Leu Smart erstellte XML-Dateien unterstützt, so dass das System nicht als offen zu bezeichnen ist. Die Offenheit könnte aber erreicht werden, wenn die von Leu Smart erstellten Dateien in das von der WfMC entwickelte XPDL-Format konvertiert würden und die Workflow-Engine mit den XPDL-Dateien arbeitet.

Der Nachteil der Eigenentwicklung gegenüber des MS Workflow Designers ist der entsprechend höhere Programmieraufwand.

4.3. Fazit

Aufgrund der Restriktionen des MS Workflow Designers scheint dieser zur Realisierung der geforderten Lösung eher weniger geeignet.
Deshalb wird die Eigenentwicklung unter Verwendung von Leu Smart als Modellierungstool präferiert.

5. Geplantes Vorgehen

Der folgenden Zeitplanung liegen folgende Annahmen zu Grunde: Für die Fertigstellung der Diplomarbeit stehen 6 Monate, also ca. 26 Wochen zur Verfügung. Davon werden 3 Wochen als Zeitpolster eingeplant, die ggf. bei Zeitverschiebungen im Plan flexibel eingesetzt werden können.

Die Planung der Diplomarbeit erfolgt analog zum beabsichtigten Aufbau der schriftlichen Ausarbeitung. Dieser Aufbau soll wie folgt aussehen:

1. Einleitung
2. Einordnung in die Literatur
3. Workflow Management
    a. Begriffe
    b. Workflow-Management-Systeme
4. Anforderungen
    a. funktionale Anforderungen
    b. nicht-funktionale Anforderungen
5. Eigenschaften / Restriktionen der (eingesetzten) mobilen Clients
6. Verteilung der Workflow-Ausführung
7. Lösungsalternativen
8. Definition Workflow-Sprache
9. Konzeption und Entwurf
10. (prototypische) Implementierung und Test
11. Analyse der Implementierung

Die einleitenden Kapitel „Einleitung“ und „Einordnung in die Literatur“ werden am Ende der Erstellungsphase der Diplomarbeit verfasst. Für beide Kapitel wird eine Bearbeitungszeit von insgesamt einer Woche eingeplant.

Das Kapitel Workflow Management soll innerhalb von drei Wochen erstellt werden. Diese drei Wochen verteilen sich wie folgt: Die Definitionen grundlegender Begriffe sind innerhalb einer Woche zu erstellen. In den restlichen zwei Wochen sind allgemeine Konzepte von WfMS (Ziele und Aufgaben), die verschiedenen Arten von WfMS, das Referenzmodell der WfMC und die Architektur eines WfMS nach [JBS99] vorzustellen.

Die Anforderungen an die Workflow-Steuerung sind innerhalb von vier Wochen zu identifizieren und darzustellen. Die Fertigstellung dieses Kapitels ist damit bis zur 7. Woche zu realisieren. Für die Identifikation der funktionalen Anforderungen ist ein Zeitfenster von drei und für die nicht-funktionalen Anforderungen ein Zeitfenster von einer Woche vorgesehen.

Die Eigenschaften und insbesondere Restriktionen eingesetzter mobiler Clients sind innerhalb einer Woche zu erarbeiten und damit bis zur 8. Woche abzuschließen.

Zur Entwicklung des Konzepts zur Verteilung der Workflow-Ausführung ist ein Zeitrahmen von zwei Wochen veranschlagt. Diese Aufgabe ist bis zur 10. Woche fertig zu stellen.

Eine Darstellung der Lösungsalternativen und schließlich die Auswahl der anzuwendenden Alternative, die zum Teil bereits in diesem Proposal vorgenommen wurde, ist innerhalb von zwei Wochen durchzuführen. Die Fertigstellung erfolgt also bis zur 12. Woche. Diese zwei Wochen sollten gleichmäßig auf die beiden Alternativen, MS Workflow Designer und Eigenentwicklung, aufgeteilt werden.

Für die Definition der Workflow-Sprache, die ja entscheidend von der gewählten Lösungsalternative abhängt, sind ebenfalls zwei Wochen eingeplant, so dass diese Tätigkeit bis zur 14. Woche beendet wird. Bei Verwendung von XML-Dateien als Speichermedium der Workflow-Modelle, wie es laut Kapitel 4 präferiert wird, sollte in diesem Kapitel ein kurzer Überblick über XML gegeben werden.

Für die Konzeptions- und Entwurfsphase ist ein Zeitraum von vier Wochen eingeplant. In dieser Phase ist innerhalb einer Woche ein Klassenmodell für den Import des Workflow-Modells zu entwickeln. In den zwei darauf folgenden Wochen sollte ein Klassenmodell der Workflow-Engine, das insbesondere auch den Aspekt der Verteilung der Workflow-Ausführung berücksichtigt, erstellt werden. Schließlich ist in der letzten Woche ein Datenbankschema zur Speicherung der workflow-relevanten Daten zu entwerfen. Das Ergebnis der Konzeptions- und Entwurfsphase ist die Erstellung einer System-architektur. Diese sollte also bis zur 18. Woche fertiggestellt werden.

Für die Implementierung und die Analyse der Implementierung sind vier Wochen Zeitaufwand veranschlagt worden, so dass die Kapitel „Implementierung und Test“ und „Analyse der Implementierung bis zur 22. Woche abgeschlossen sein sollten.

Eine Übersicht der Zeit- und Aufwandsplanung ist der Tabelle 1 zu entnehmen.

	Aufgabe	Aufwand in Wochen	Fertigstellung bis Woche
1.	Einleitung	0,5	23.
2.	Einordnung in die Literatur	0,5	23.
3.	Workflow Management a. Begriffe b. Workflow-Management-Systeme	3	3.
4.	Anforderungen a. funktionale Anforderungen b. nicht-funktionale Anforderungen	4	7.
5.	Eigenschaften / Restriktionen der (eingesetzten) mobilen Clients	1	8.
6.	Verteilung der Workflow-Ausführung	2	10.
7.	Lösungsalternativen	2	12.
8.	Definition Workflow-Sprache	2	14.
9.	Konzeption und Entwurf	4	18.
10. 11.	(prototypische) Implementierung und Test Analyse der Implementierung	4	22.

Tabelle 1: Übersicht Terminplanung der Diplomarbeit

6. Literaturverzeichnis

[AG95] Alonso, G.; Günthör, R.; Kamath, M.; Agrawal, D.; El Abbadi, A.; Mohan, C.: *Exotica/FMDC: Handling Disconnected Clients in a Workflow Management System.* In: Proc. 3rd Int'l Conf. on Cooperative Information Systems. Wien, Mai 1995.

[AG96] Alonso, G.; Günthör, R.; Kamath, M.; Agrawal, D.; El Abbadi, A.; Mohan, C.: *Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients*. In: Distributed and Parallel Databases, 4(3), S. 229ff. Juli 1996.

[BD98] Bauer, Thomas; Dadam, Peter: *Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse*. In: Ulmer Informatik-Berichte, Nr. 98-02. Januar 1998.

[BD99] Bauer, Thomas; Dadam, Peter: *Verteilungsmodelle für Workflow-Management-Systeme – Klassifikation und Simulation*. Informatik Forschung und Entwicklung, Volume 14 Issue 4: 203-217. Springer Verlag, Dezember 1999.

[BRD01] Bauer, Thomas; Reichert, Manfred; Dadam, Peter: *Effiziente Übertragung von Prozessinstanzdaten in verteilten Workflow-Management-Systemen*. Informatik Forschung und Entwicklung, Volume 16 Issue 2: S. 76-92. Springer Verlag, 2001.

[Bu95] Bussler, Christoph: *User Mobility in Workflow-Management-Systems*. In: Proceedings of the Telecommunications Information Networking Conference (TINA '95), Melbourne, Australia. Februar 1995.

[Ch99] Chapmen, Elizabeth F.: *Developing a Team Solution Using Microsoft Access Workflow Designer. Microsoft Corporation*, September 1999. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc2k/html/mod_walkthrough.asp

[EG96] Eder, Johann; Groiss, Herbert: *Ein Workflow-Management-System auf der Basis aktiver Datenbanken*. In: Vossen, Gottfried; Becker, Jörg: Geschäftsprozessmodellierung und Workflow-Management, S. 389ff. 1. Auflage, unveränderter Nachdruck. International Thomson Publishing GmbH, 1995.

[GS01] Gruhn, Volker; Schöpe, Lothar: *Unterstützung von verteilten Softwareentwicklungsprozessen durch integrierte Planungs-, Workflow- und Groupware-Ansätze*. Memo Nr. 115. Universität Dortmund, Fachbereich Informatik, Lehrstuhl Software-

Technologie. September 2001.

[Gö03] Göbel, Stefan: *Konzeption und Entwicklung eines Action/Event-Mechanismus für mobile Endgeräte mit XML*. Universität Dortmund, Fachbereich Informatik, Lehrstuhl Software-Technologie. 2003.

[HSS96] Heinl, P., Schuster, H., Stein, K.: *Behandlung von Ad-hoc-Workflows im MOBILE Workflow-Modell*. In: Proc. Softwaretechnik in Automation und Kommunikation - Rechnergestuetzte Teamarbeit (STAK'96, Muenchen).1996.

[JBS99] Jablonski, Stefan; Böhm, Markus; Schulze, Wolfgang: *Workflow-Management – Entwicklung von Anwendungen und Systemen*. 1. Auflage, korrigierter Nachdruck. dpunkt-Verlag, 1999.

[JH99] Jing, J.; Huff, K.; Sinha, H.; Hurwitz, B.; Robins, B.: *Workflow and Application Adaptations in Mobile Environments*. In: Second IEEE Workshop on Mobile Computing Systems and applications. Februar 1999.

[KSS02] Kotthoff, Christian; Süselbeck, Richard; Schöpe, Lothar: *„Architektur des Spiw-Kommunikationssystems*“, Draft 0.0. Universität Dortmund, Fachbereich Informatik, Lehrstuhl 10. Dezember 2002.

[Le99] Lehmann, Frank R.: *Fachlicher Entwurf von Workflow-Management-Anwendungen*. 1. Auflage. B.G. Teubner, 1999.

[Li99] Lindert, Frank: *Fraktales Prozessmanagement*. Technische Universität Berlin, Fachbereich 13 Informatik, Dissertation, 1999.

[MW98] Muth, P., Wodtke, D., Weissenfels, J., Kotz Dittrich, A., Weikum, G.: *From Centralized Workflow Specification to Distributed Workflow Execution*. In: JIIS - Special Issue on Workflow Managament, Volume 10, Number 2, Kluwer Academic Publishers, März 1998.

[MWY00] Marra, Marci; Whitcomb, Valerie; Yoder, Doug: *Workflow Designer for Exchange: Automating Workflow on Exchange Folders*. Micosoft Corporation, August 2000. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmes2k/html/pwd_buildworkflow.asp

[OMG00] Object Management Group: *Workflow Management Facility – Specification.* Version 1.2. April 2000.

[Ro96] Rose, Thomas: *Vorgangsmanagementsysteme: Modellierungs- und Implementierungskonzepte*. In: Vossen, Gottfried; Becker, Jörg: Geschäftsprozessmodellierung und Workflow-Management, S. 319ff. 1. Auflage, unveränderter Nachdruck. International Thomson Publishing GmbH, 1995.

[Ro02] Roof, Larry: *Getting Started with Visual Studio .NET and the Microsoft .NET Compact Framework*. Microsoft Corporation, März 2002. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetcomp/html/anch_dotNetCompfx.asp.

[Sch97] Schuster, Hans: *Architektur verteilter Workflow-Management-Systeme*. Universität Erlangen-Nürnberg, Technische Fakultät, Dissertation, 1997.

[Sk01] Skonnard, Aaron: *XML in .NET: .NET Framework XML Classes and C# Offer Simple, Scalable Data Manipulation*. Microsoft Corporation, MSDN Magazine, Januar 2001. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmsxml/html/whatsnew40rtm.asp

[WMC95] Workflow Management Coalition: *The Workflow Reference Model*. Issue 1.1. Januar 1995.

[WMC99] Workflow Management Coalition: Terminology & Glossary. Issue 3.0. Februar 1999.

[WMC01] Workflow Management Coalition: *Workflow Management Coalition Workflow Standard – Interoperability Wf-XML Binding*. Version 1.1. November 2001.

[WMC02] Workflow Management Coalition: *Workflow Process Definition Interface – XML Process Definition Language*. Version 1.0. Oktober 2002.