

# Projektbericht

## MOKASSIN – Innovative Technologien zum Prozeßmanagement und zur semantischen Datenintegration

Gregor Joeris, Holger Wache, Otthein Herzog

Technologie-Zentrum Informatik, Universität Bremen

[joeris|wache|herzog@tzi.de](mailto:joeris|wache|herzog@tzi.de)

<http://www.tzi.de/grp/mokassin>

### 1 Aufgabenstellung

#### Motivation und Zielsetzung

„Änderungen sind unser Geschäft“ – Diese Aussage der Anwenderpartner des MOKASSIN-Projekts umschreibt die Rahmenbedingung, die zur Zielsetzung des Projektes geführt haben: den Paradigmenwechsel zu flexiblen und prozeßorientierten Organisationsformen, die Vernetzung unternehmensübergreifender Geschäftsprozesse und die Integration der in diesen Prozessen involvierten Anwendungssysteme durch innovative Informationstechnologien zu unterstützen.

Anwender sollen in die Lage versetzt werden, ihre spezifischen Geschäftsprozesse oder Arbeitsabläufe zu modellieren, auszuführen, zu überwachen und flexibel anzupassen. Dabei sind Prozesse unterschiedlichen Strukturierungs- und Automatisierungsgrades abzubilden: neben stark strukturierten Prozessen gilt es, auch flexible Arbeitsformen zu unterstützen; neben manuellen Aufgaben sind ebenso automatisch durch Anwendungssysteme durchzuführende Aktivitäten zusammen mit ihren Daten auf einer semantischen Ebene zu integrieren. Schließlich werden Prozesse in einer verteilten Umgebung ausgeführt, die verschiedene Arbeitsplätze, Abteilungen oder Unternehmen umfassen kann.

#### Herausforderungen

Das Projekt bewegte sich damit zum einen im Kontext von Schlüsseltechnologien wie Workflow- und Dokumentenmanagement, zum anderen wurde auf Probleme der Anwendungssystemintegration insbesondere hinsichtlich der semantischen Datentransformation fokussiert. Während die Kopplung heterogener Anwendungssysteme auf der Basis von Standards und Middleware-Technologien wie CORBA/DCOM heutzutage weitgehend unproblematisch ist, gilt dies für den Austausch der Anwendungsdaten keineswegs. Hierbei sind nicht nur syntaktische, sondern auch semantische Heterogenitäten zwischen den einzelnen Datenbeständen aufzuheben. Um eine schnelle und flexible Integration der Systeme zu ermöglichen und die Programmierung einer solchen Integration von Hand zu vermeiden, war die Entwicklung einer *semantikbasierten Integrationsmethode* für heterogener Informationsquellen vorrangiges Ziel bei der Systemintegration.

Auf dem Gebiet des *Prozeßmanagements* – so bezeichnen wir die systematische Modellierung und Ausführung von Prozessen und die integrierte Betrachtung von Workflow- und Dokumentenmanagement – stellen die oben skizzierten *Flexibilitäts-* und *Verteilungsaspekte* eine große Herausforderung an die Entwicklung eines umfassenden Prozeßmanagementsystems dar. Die Notwendigkeit, sowohl *a priori* eine flexible Ausführung von Prozessen zu ermöglichen, indem Freiheitsgra-

de für die Prozeßdurchführung offen gelassen werden, als auch *a posteriori* Prozesse jederzeit, auch während der Ausführung, flexibel an geänderte Situationen anpassen zu können, war nicht nur das Ergebnis der Analyse der Prozesse der beteiligten Anwenderpartner, sondern hat sich inzwischen zu zentralen Anforderungen auf dem Gebiet des Workflowmanagement entwickelt. Gleiches läßt sich für die Unterstützung unternehmensübergreifender Prozesse beobachten.

### Förderung und Projektpartner

Das MOKASSIN-Projekt wurde vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) vom 01. April 1996 bis zum 30. April 1999 gefördert (FKZ 01 IS 606 B 8). Es wurde unter der Leitung der VSS GmbH (Bremen) in Zusammenarbeit mit der Arbeitsgruppe Künstliche Intelligenz im Technologie-Zentrum Informatik (TZI) der Universität Bremen und dem Bremer Institut für Betriebstechnik und angewandte Arbeitswissenschaft an der Universität Bremen (BIBA) durchgeführt. Des weiteren waren während der Anforderungsanalyse zehn assoziierte Anwenderpartner beteiligt, um Prozesse und Anforderungen unterschiedlicher Branchen analysieren zu können. Aus diesen wurden drei Anwender ausgewählt, die in das Projekt bis zur abschließenden Evaluierung eingebunden waren.

## 2 Ergebnisse

Die Funktionalitäten des MOKASSIN-Systems lassen sich in die Bereiche Prozeßmanagement sowie System- und Datenintegration unterteilen. Abbildung 1 gibt eine Übersicht über diese Funktionalitäten und faßt sie in verschiedene Blöcke zusammen, anhand derer die Vorstellung der Ergebnisse strukturiert wird.

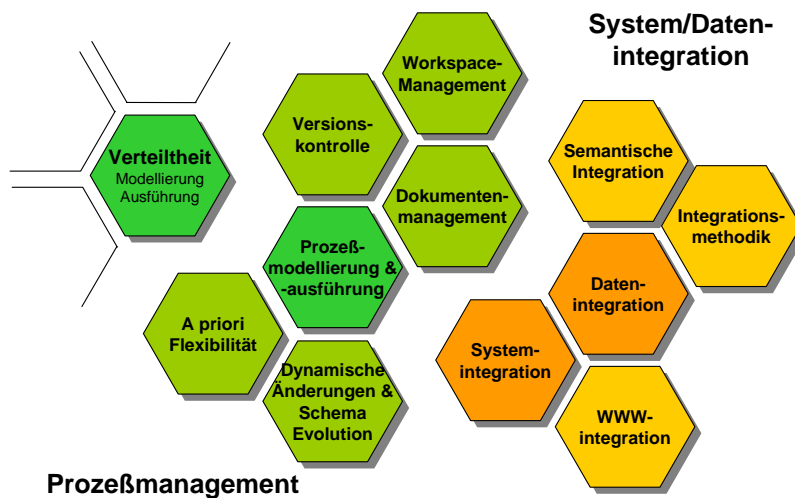


Abbildung 1: Funktionalitäten von MOKASSIN

Aus Prozeßsicht steht die Möglichkeit zur Modellierung und verteilten Ausführung von Prozessen im Mittelpunkt. Ein besonderer Schwerpunkt liegt dabei auf dem flexiblen Prozeßmanagement. Diesen beiden Aspekte werden in den folgenden beiden Kapiteln im Detail beschrieben, während wir die weiteren Komponenten des Prozeßmanagements nur kurz skizzieren:

Das integriertes *Dokumentenmanagement* stellt ein Repository zur Ablage der Dokumente und ihrer Versionen zur Verfügung und kontrolliert die Replikationen der Dokumente, die jedem Prozeßbeteiligten in seinem individuellen *Workspace* bereitgestellt werden. Auf die Integration von Techniken des Versions- und Workspace-Managements auf konzeptioneller Ebene wurde besonderer Wert gelegt, um eine Kooperationsunterstützung im Rahmen eines Workflows zu ermöglichen.

Um auch unternehmensübergreifende Prozesse modellieren und ausführen zu können, wurden alle bisher erwähnten Funktionalitäten des Prozeßmanagements über verschiedene dezentrale und unabhängige Server hinweg über Proxy-Dienste integriert. Die Kopplung von Prozeßmanagementservern ist in Abbildung 1 als Verteiltheitsaspekt angedeutet.

Ein anderer Bereich von Funktionalität bietet MOKASSIN bei der *Integration von Systemen und Daten*. Über eine standardisierte CORBA-Schnittstelle können Funktionen heterogener Anwendungssysteme in die Prozesse eingebunden werden. Beispielhaft wurde darüber hinaus gezeigt, wie sich *web-basierte Anwendungen* in das System integrieren lassen. Der Schwerpunkt der Arbeiten lag jedoch auf dem Gebiet der semantische *Datenintegration*. Auf diese werden wir in Kapitel 2.3 detaillierter eingehen.

## 2.1 Modellierung und verteilte Ausführung durch Aufgabeagenten

In MOKASSIN wurde ein Prozeßmanagementsystem der nächsten Generation für dynamische und verteilte Prozesse entwickelt. Während wir die Ergebnisse hinsichtlich der Flexibilitätsproblematik im nächsten Kapitel betrachten, konzentrieren wir uns an dieser Stelle auf die verteilte Ausführung von Workflows durch sogenannte Aufgabenagenten. Neben diesen Flexibilitäts- und Verteilungsaspekten waren die Bereitstellung einer Prozeßmodellierungssprache auf hohem Abstraktionsniveau und die Unterstützung der Wiederverwendbarkeit der Prozeßmodelle wichtige Entwurfsziele.

**Verteilte Workflowausführung durch interagierende Agenten:** Das Herzstück jedes Workflowmanagement-Systems ist die Workflow-Engine, die die modellierten Prozesse ausführt. In MOKASSIN wurde auf der Basis der Agententechnologie eine vollständig verteilte Workflow-Engine entwickelt [JoHe99]. Jede Aufgabe wird von einem sogenannten Aufgabenagenten koordiniert, dessen Verhalten aus dem Prozeßschema abgeleitet wird. Es handelt sich somit nicht um autonome Agenten, vielmehr steht das Interaktionsparadigma bei den Aufgabenagenten im Vordergrund. Insgesamt wird es durch diesen Ansatz möglich, Workflowunterstützung in verschiedenste Anwendungsszenarien und -systeme einzubetten, da auf eine monolithische Workflow-Engine verzichtet werden kann.

Ein Aufgabenagent hat sowohl das Wissen über den Kontext der Aufgabe (die Vorgänger- und Nachfolgaufgaben, die Oberaufgabe und Unteraufgaben, die Produzenten und Konsumenten von Ein- und Ausgabedaten) als auch über das Zustandsübergangsverhalten der Aufgabe. Dieses Verhalten wird durch Zustände, Zustandsübergänge sowie Event-Condition-Action (ECA) Regeln beschrieben. Letztere legen fest, auf welches Ereignis unter welcher Bedingung ein Zustandsübergang durchgeführt werden darf. Bei einem Zustandsübergang wird ein Ereignis ausgelöst, das vermöge der Kontextinformationen an in Beziehung stehende Aufgabenagenten übermittelt wird. Wird beispielsweise eine Aufgabe beendet, so überprüfen die Agenten der Nachfolgaufgaben in Reaktion auf dieses Ereignis, ob ihre Aufgabe nun bereit zur Ausführung ist und führen ggf. einen Zustandsüber-

**Kommentar:** Dieses wird aus dem kontextfreien Verhalten der Aufgabendefinition und dem kontextsensitiven Verhalten abgeleitet, das sich durch die Verwendung einer Aufgabe innerhalb eines Workflows ergibt.

gang in den Zustand 'ready' aus. Neben diesem Standardablauf lassen sich auch sehr komplexe Verhaltensmuster realisieren (siehe nächstes Kapitel).

## 2.2 Flexibles Prozeßmanagement

Die Anforderungsanalyse hat hinsichtlich des Flexibilitätsaspekts zwei zentrale Anforderungen erbracht: Zum einen die Modellierung von Workflows, die a priori eine flexiblere Prozeßdurchführung zulassen, indem sie Freiheitsgrade für die Ausführung offenlassen, zum anderen die Steigerung der Flexibilität durch eine nachträglich mögliche Änderung eines Prozesses und die Evolution von Prozeßmodellen. Sowohl zur Unterstützung der a priori als auch der a posteriori Flexibilität wurden in MOKASSIN neue Technologien entwickelt und umgesetzt:

- **Modellierung von a priori flexiblen Prozeßmodellen:** Durch eine wesentlich erweiterte Möglichkeiten zur Verhaltensmodellierung können in MOKASSIN Prozesse mit unterschiedlichem Grad an Determiniertheit modelliert und a priori *Freiheitsgrade für die Ausführung* offengelassen werden. Der Kontrollfluß wird deklarativ in Form von Abhängigkeiten zwischen Aufgaben in einem Aufgabengraphen beschrieben, so daß im Gegensatz zu einer prozeduralen Sicht partiell definierte Workflows unterstützt werden können. Des Weiteren ist es möglich, neben der herkömmlichen End-Start-Abhängigkeit neue Arten von Kontrollflußabhängigkeiten zu definieren und in verschiedenen Workflows wiederzuverwenden. Zahlreiche flexible Verhaltensmuster können so realisiert werden (z.B. wechselseitiger Ausschluß, Synchronisation dynamisch aktivierter Zweige, verteilte Transaktionsprotokolle). Die Definition eines Kontrollflußabhängigkeitstyps erfolgt durch einen Experten auf der Basis von ECA-Regeln und berücksichtigt die verteilte Workflowausführung [JoHe99]. Der Ansatz kombiniert somit die Flexibilität regelbasierter Ansätze mit dem hohem Abstraktionsniveau graphbasierter Prozeßmodellierungssprachen.
- **Kooperationsunterstützung:** Neben einem flexiblen Kontrollfluß wird in MOKASSIN auch ein *flexibler Datenaustausch* unterstützt. Dies wird durch die Integration von Versions- und Workspacemanagementkonzepten erreicht. Das System verwaltet somit nicht nur die Arbeitslisten der Akteure, sondern ebenfalls für jede Aufgabe einen individuellen Arbeitsbereich. Durch einen geregelten Datenaustausch zwischen diesen Arbeitsbereichen, beispielsweise durch die frühzeitige Freigabe von Dokumenten, ist es möglich, kooperative Tätigkeiten im Rahmen eines wohldefinierten Workflows zu unterstützen. Im Gegensatz zu einer freien Kooperation, wie sie von Groupwaresystemen unterstützt wird, sprechen wir hier von *kontrollierter Kooperation*.
- **Dynamische Prozeßänderungen:** Dynamische Änderungen von Prozessen können in MOKASSIN auf der Basis des agentenbasierten Ausführungsmodell ohne zusätzliche Sprachkonzepte behandelt werden. Unter Verwendung der ECA-Regeln und in Analogie zu den Zustandsübergängen haben die Aufgabenagenten das Wissen darüber, wann eine Änderung durchgeführt werden kann bzw. wie auf eine Änderung reagiert wird. Eine weitere Besonderheit des Ansatzes ist es, daß sich das Ausführungsverhalten einzelner Änderungsoperationen an verschiedene Aufgabenarten und Situationen anpassen läßt. Die Möglichkeit dieser situativen Anpassung hat sich dabei als wichtige Grundlage für die Durchführung dynamischer Änderungen erwiesen.
- **Workflow-Evolutionsmanagement:** Neben der Änderung einzelner Instanzen wird in MOKASSIN auch die Modifikation des Prozeßschemas und die Verwaltung verschiedener Prozeßversionen unterstützt. Der Ansatz beruht auf der Adaptierung von Konzepten der Schema-Versionierung für das Workflow-Management und dem Modell zur integrierten Repräsentation von Prozeßsche-

ma und –instanzen. Damit wurde erstmals ein umfassendes Konzept zur Prozeßschemaversionierung entwickelt [JoHe98]. Darüber hinaus wird die direkte Migration bereits laufender Instanzen unterstützt, wobei insbesondere eine vom Ausführungszustand abhängige Migration erlaubt wird. Damit können in MOKASSIN komplexe Schemaänderungen und ihre Auswirkungen auf laufende Instanzen gehandhabt werden.

## 2.3 Semantische Datenintegration

Im Bereich der semantischen Datenintegration wurde der Informationsaustausch zwischen existierenden Informations- und Applikationssystemen untersucht. In MOKASSIN wurde zur Realisierung der semantischen Datenintegration auf die Mediator/Wrapper-Architektur [Gar+95] zurückgegriffen. Die in MOKASSIN im Bereich der semantischen Datenintegration erzielten Ergebnisse führen die Entwicklung der Mediatoren [Wie92] fort. Konkret sind diese

- eine semantikbasierte Integrationsmethode heterogener Informationsquellen für Mediatoren und
- eine Integration von Kontexttransformationen in Mediatoren [Wac99].

**Semantikbasierte Integrationsmethode:** Bei der Integration existierender Informationsquellen und Applikationssysteme muß sich implizit oder explizit mit deren Semantik auseinandergesetzt werden. Nur auf der semantischen Ebene können korrespondierende Informationseinheiten gefunden werden; der Abgleich der syntaktischen Diskrepanzen ist dann vergleichsweise einfach. In MOKASSIN wurde eine Integrationsmethode entwickelt, welche auf einer expliziten Repräsentation der Semantik von Informations- und Applikationssystemen aufsetzt. Im Gegensatz zu anderen Integrationsmethoden erhebt und pflegt diese Integrationsmethode gleichzeitig die semantische Beschreibung. Insbesondere fokussiert die Integrationsmethode auf den Aufbau und die Wartung einer globalen, einheitlichen Terminologie, auf die die semantischen Beschreibungen zurückgreifen. Hierbei ist die Methode auf eine minimale und pragmatische Erhebung der Terminologie und der semantischen Beschreibungen ausgerichtet. Aus dem Vergleich der semantischen Beschreibungen abduziert ein Unterstützungssystem semantisch korrespondierende Informationseinheiten, welche dann automatisch in entsprechende Regeln für eine Spezifikation eines Mediators überführt werden können.

**Kontexttransformation:** Eine Information muß immer im Kontext ihrer Quelle betrachtet werden. Insbesondere muß eine Information explizit in den Kontext einer Anfrage übersetzt werden. Die Mediator-Technologie wurde dahingehend erweitert, daß Kontexttransformationen nun ebenfalls als separate Regeln beschrieben werden können. Die Kontextregeln erforderten eine Erweiterung der Ausführungsmaschine im Mediator. Die explizite Kontexttransformation vereinfacht die Modellierung der Transformationen entscheidend.

## 3 Agenten-basiertes Rahmenwerk

Die Grundlage der Architektur des MOKASSIN-Systems bildet ein Agenten-basiertes Rahmenwerk, in dem die verschiedenen Aspekte des Systems eingebunden sind (diese logische Sicht auf das System ist in Abbildung 2 dargestellt). Sowohl das Prozeßmanagementsystem als auch das System zur semantischen Datentransformationen enthalten Komponenten, mit denen sich Wissen über die Pro-

zesse bzw. Informationsquellen beschreiben läßt, sowie Komponenten, die dieses Wissen interpretieren:

Im Falle des Prozeßmanagements wird über die Komponente zur Prozeßmodellierung Wissen über die verschiedenen Aspekte eines Prozesses (Aufgaben und Workflow, organisatorische Aspekte, involvierte Systeme, Daten und Dokumente) erfaßt. Das Prozeßwissen wird von den Aufgabenagenten zur Steuerung des Workflows verwendet, die somit eine verteilte Workflow-Engine als Kern des Prozeßmanagements darstellen. Ergänzt wird diese Funktionalität durch die integrierten Repository-Dienste. Aufgrund der erweiterten Funktionalitäten bietet der sog. ActorDesktop einem Akteur neben der Verwaltung der Arbeitsliste eine erweiterte Funktionalität zum Zugriff auf den aufgabenbezogenen Workspace, zum Monitoring des Prozeßkontextes und zum Durchführen von ad hoc Änderungen.

Das Prozeßmanagementsystem wurde als drei-schichtiges Client-/Server-System in C++ entwickelt. Die Datenbankschicht des Servers ist auf der Basis eines RDBMS und die Applikationsschicht als verteiltes Objekt bzw. Agentensystem unter Einsatz von CORBA plattformunabhängig realisiert worden. Die Clients wurden unter MS Windows auf der Basis der MFC entwickelt.

Im Falle der semantischen Datenintegration ermöglicht die Datenmodellierungskomponente MECA (MEdiator Configuration Assistant) die Beschreibung von Datenquellen auf einer semantischen Ebene. Hier kann insbesondere auch die Internstruktur von Dokumenten beschrieben werden, die im Prozeßmanagement nur auf einer grobgranularen Ebene betrachtet und verwaltet werden. Mit Hilfe dieser Beschreibungen kann der MediatorAgent einen Anfrageplan erstellen, um Daten, die zwischen zwei Anwendungen auszutauschen sind, in das entsprechende Anwendungsformat transformieren. MECA und MediatorAgent sind in Prolog und XPCE implementiert worden.

Anwendungssysteme, die von der Workflow-Engine aufgerufen und deren Daten über den Mediator transformiert werden können, werden mit Hilfe der Systemwrapper in das MOKASSIN-System integriert, die in C++ und mit CORBA realisiert wurden. Hierbei wird zwischen dem SystemCallAgent, der den Aufruf des Anwendungssystems übernimmt und mit dem Aufgabenagent interagiert, und dem DatenAgenten unterschieden, der speziell den Zugriff auf Datenquellen sicherstellt.

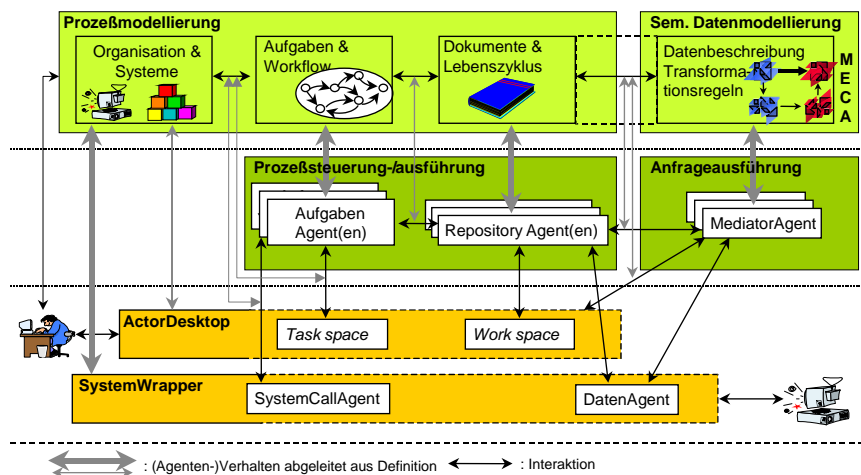


Abbildung 2: Die Komponenten des Systems

## 4 Ausblick

Nicht zuletzt aufgrund des großen Interesses der Anwenderpartner wird zur Zeit die Fortsetzung des Projektes mit dem Ziel der industriellen "Härtung" der Ergebnisse angestrebt. Darüber hinaus wird untersucht, wie Ansätze zum Workflow- und Wissensmanagement integriert werden können. Schließlich befinden sich zu den Themen flexibles Prozeßmanagement und semantische Datenintegration zwei Dissertationen in der Abschlußphase.

Weitere Informationen über das Projekt einschließlich online verfügbarer Veröffentlichungen und Präsentationen finden sich im WWW unter <http://www.tzi.de/grp/mokassin>.

## 5 Referenzen

- [Gar+95] Garcia-Molina, H.; Hammer, J.; Ireland, K.; Papakonstantinou, Y., Ullman, J.; Widom, J.: Integrating and accessing heterogeneous information sources in Tsimmis. In: *Proceedings of the AAAI Symposium on Information Gathering*, pp. 61-64, Stanford, California, March 1995.
- [JoHe98] Joeris G., Herzog O.: Managing Evolving Workflow Specifications. In *Halper M. (ed.) Proceedings of the 3<sup>rd</sup> IFCIS Conf. On Cooperative Information Systems (CoopIS'98)*, New York, Aug. pp. 310-319 (1998).
- [JoHe99] Joeris G., Herzog O.: Towards Flexible and High-Level Modeling and Enacting of Processes. In *Proc. of the 11<sup>th</sup> Int. Conf. on Advanced Information Systems Engineering (CAiSE'99)*, LNCS 1626, Springer Verlag, Heidelberg, Germany, pp. 88-102 (1999).
- [Wac99] Wache, H.: Towards Rule-based Context Transformation in Mediators. *Proc of 2nd International Workshop on Engineering Federated Information Systems (EFIS'99)*, Kùhlungsborn, Germany, May 1999.
- [Wie92] Wiederhold, G.: Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3) March 1992, pp. 38-49.