

# Realisierung flexibler, unternehmensweiter Workflow-Anwendungen mit ADEPT

Manfred Reichert · Thomas Bauer · Thomas Fries · Peter Dadam

Universität Ulm

Fakultät für Informatik, Abt. Datenbanken und Informationssysteme

Email: {reichert, bauer, fries, dadam}@informatik.uni-ulm.de

## Zusammenfassung

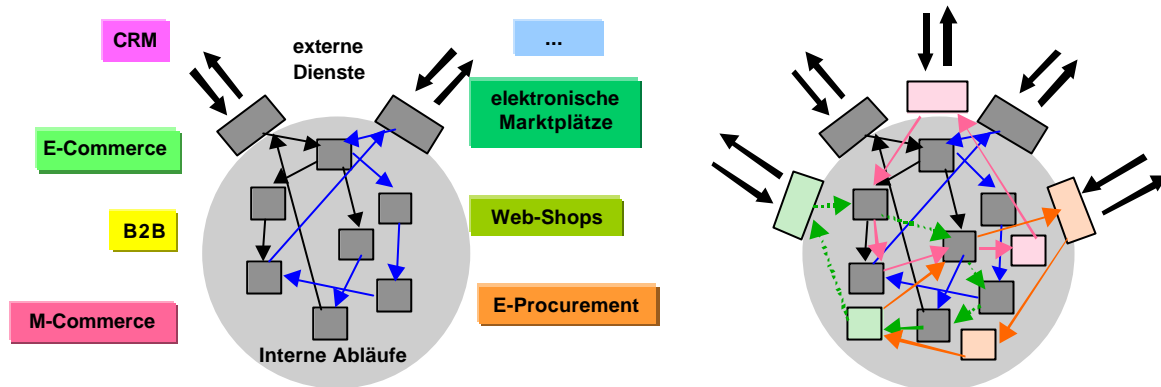
Die Unterstützung unternehmensweiter und -übergreifender Geschäftsprozesse stellt für Workflow-Management-Systeme (WfMS) eine besondere Herausforderung dar: Es sind sehr viele Organisationseinheiten (auch externe) involviert, die Prozesse können langlaufend sein (Wochen, Monate), sie müssen rasch an neue Gegebenheiten anpassbar sein, und bei Bedarf muss im Einzelfall spontan vom geplanten Ablauf abgewichen werden können (z. B. Auslassen, Einfügen oder Verschieben von Prozessschritten). Prozessorientierte Anwendungssysteme müssen – auch im Fall von Ad-hoc-Abweichungen – für EDV-Laien einfach bedienbar sein, sie müssen robust und stabil laufen und das WfMS muss auch bei einer großen Anzahl von Benutzern und Prozessinstanzen performant sein. Im Rahmen des ADEPT-Projektes arbeiten wir seit 1994 intensiv an den technologischen Grundlagen und der Entwicklung eines WfMS der nächsten Generation, das alle diese Aspekte ganzheitlich und sehr grundlegend adressiert. Der realisierte ADEPT-WfMS-Prototyp weist die Implementierbarkeit und das Zusammenspiel der entwickelten Konzepte nach und zeigt, dass Flexibilität, Robustheit und Effizienz keine Widersprüche sein müssen. Der Beitrag erläutert die zugrundeliegende Problemstellung, die technologischen Herausforderungen sowie die Einsatzperspektiven für ein solches System.

## 1 Einleitung

Mit E-Business entstehen neue Spielregeln im Markt, worauf sich die Unternehmen rasch mit geeigneten Strategien einstellen müssen. Auf ständig neue Trends und Herausforderungen müssen sie mit immer neuen Produkt- und Serviceangeboten reagieren und diese schnell in die betrieblichen Abläufe integrieren (vgl. Abbildung 1). Die optimale Gestaltung und Beherrschung der Geschäftsprozesse sowie die Fähigkeit, diese rasch und kostengünstig an neue Gegebenheiten anzupassen, wird in diesem Kontext für viele Unternehmen zur Überlebensfrage.

Die rasche Anpassbarkeit von Geschäftsprozessen wird auch deshalb immer wichtiger, weil andere Rationalisierungspotenziale für Unternehmen weitgehend ausgereizt sind und sie sich deshalb gegenüber dem Wettbewerb kaum mehr über den Preis differenzieren können. In dieser Situation werden Qualitätsaspekte zu einem entscheidenden Wettbewerbsfaktor. Vieles, was derzeit im Kontext von E-Business diskutiert wird, wie Customer Relationship Management (CRM), Supply Chain

Management (SCM) oder E-Procurement, zeigt bereits in diese Richtung und deutet an, was in Zukunft an Marktdynamik auf Unternehmen zukommen wird.



Ständig neue Trends erfordern von Unternehmen ...

... ständige neue Produkt- und Serviceangebote, die in existierende Abläufe integriert werden müssen

**Abb. 1:** Neue Herausforderungen durch E-Business

Für den schnellen und erfolgreichen Einstieg ins E-Business genügt es aber nicht, die Prozesse an bereitgestellte IT-Funktionen (z. B. SAP R/3) anzupassen. Die Unternehmen müssen vielmehr auf die sich ständig ändernde Wettbewerbssituation durch rasche Umgestaltung ihrer Geschäftsprozesse reagieren können. Sie müssen in der Lage sein, in kurzer Zeit – unter Umständen sogar innerhalb von Tagen oder Stunden – auf Aktionen der Mitbewerber zu reagieren und ihre Geschäftsprozesse sowie die sie unterstützenden Anwendungssysteme entsprechend rasch anzupassen oder sogar neu zu implementieren. Wenn sich ein Prozess ändert, wollen Nutzer allerdings nicht jedes Mal aufwendige Programmierarbeit leisten, sondern sie möchten ihre Abläufe möglichst rasch und einfach neu strukturieren bzw. definieren können. Dabei sind sowohl zugekaufte Systeme (z. B. ERP-Anwendungen) als auch historisch gewachsene Applikationen (*Legacy Systems*) auf Daten-, Funktions- und Prozessebene zu integrieren.

Dies erfordert ein radikales Umdenken hinsichtlich der IT-seitigen Unterstützung der Geschäftsprozesse. Es gilt, die rein funktions- und datenzentrierten Sichten, bei der die prozessorientierte Verknüpfung der Anwendungsfunktionen weitgehend in den Köpfen der Mitarbeiter oder „hart verdrahtet“ in den Anwendungsprogrammen vorhanden ist, konsequent zu erweitern [DaRK00]. Prozessänderungen sind in diesem Umfeld extrem fehlerträchtig und nur mit sehr hohem Aufwand an Personal, Kosten und Zeit zu bewerkstelligen [RDMK00].

Aus diesen Gründen müssen prozessorientierte Anwendungen in Zukunft rasch im „Plug & Play“-Stil entwickelt sowie an neue Gegebenheiten angepasst werden können (vgl. Abbildung 2). Um dies zu ermöglichen, wird eine Software-Technologie benötigt, die in umfassender, ganzheitlicher Weise unternehmensweite und –übergreifende Geschäftsprozesse ausführen, verwalten und überwachen kann. Es muss ferner möglich sein, notwendige Prozessänderungen rasch einzubringen und sie automatisch „per Knopfdruck“ auf schon laufende Prozesse zu übertragen. Darüber hinaus dürfen Prozesse niemals „starr“ implementiert sein. Im Bedarfsfall müssen Anwender ad hoc vom vorgeplanten Ablauf abweichen können, wenn beispielsweise eine nicht vorhergesehene Ausnahmesituation einge-

treten ist [DaRK00]. Die Unterstützung einer solchen Flexibilität ist für viele Anwendungen unerlässlich. Sie bedingt aber auch, dass systemseitig geprüft und sichergestellt werden kann, dass bei Ad-hoc-Änderungen im weiteren Verlauf des betroffenen Prozesses keine Inkonsistenzen auftreten [ReDa98, Reic00].

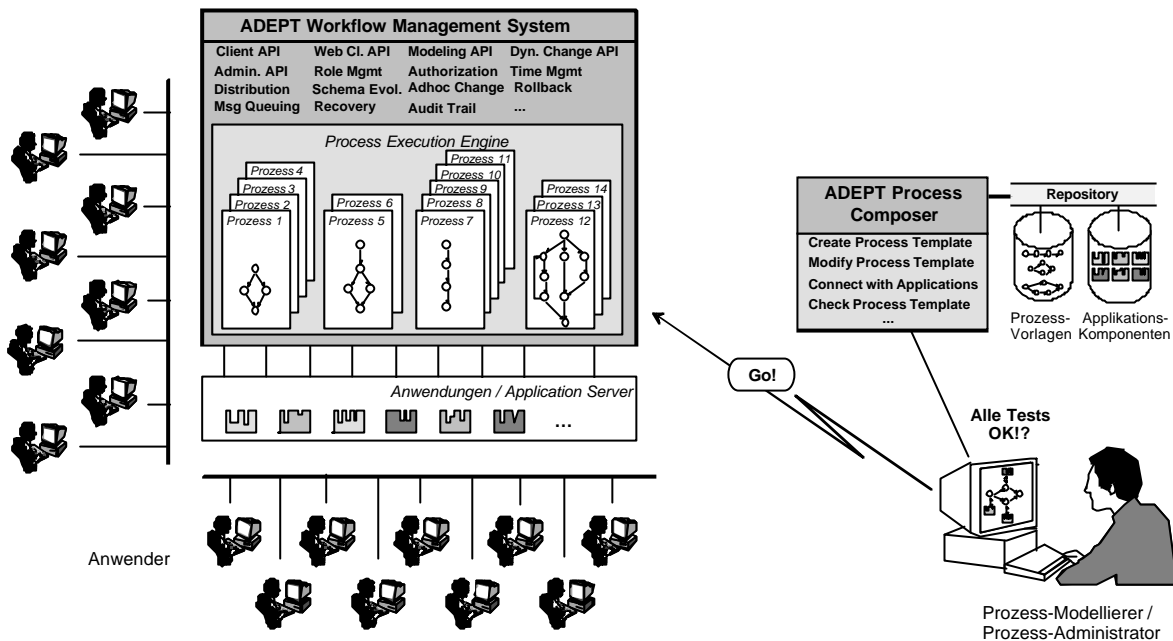


Abb. 2: Zukünftige Entwicklung prozessorientierter Anwendungen mittels Plug & Play

Für die elektronische Unterstützung von Geschäftsprozessen wichtige, von derzeitigen Workflow-Management-Systemen (WfMS) aber bisher kaum oder gar nicht betrachtete Anforderungen sind:

- Konsequente Trennung von Ablauflogik und Anwendungscode
- Rapid Prototyping durch „Plug & Play“-Techniken
- Intensive Korrektheitsprüfungen bereits zur Modellierungszeit
- Unterstützung von Ad-hoc-Änderungen einzelner Prozesse mit systemseitiger Konsistenzsicherung
- Unterstützung und Überwachung von temporalen Bedingungen
- Rasche Umsetzung von Prozessänderungen durch Prozess-Schemaevolution
- Skalierbarkeit für den unternehmensweiten Einsatz
- Unterstützung übernehmensübergreifender Anwendungen
- Scheduling und Ressourcen-Management, Bereitstellung eines „Prozess-Leitstands“
- Fundierte theoretische Basis

Production Workflow Systeme [LeRo00], wie MQSeries Workflow oder Staffware, zeigen im Prinzip bereits in die richtige Richtung. Prozesslogik und Anwendungsfunktionalität sind hier voneinander getrennt. Eine große Schwäche dieser Systeme liegt jedoch in ihrer Starrheit, d. h. in ihrer nicht vorhandenen oder sehr schwach ausgeprägten Fähigkeit, flexibel auf Ausnahmesituationen

zu reagieren [CaCP98, MüRa00, ReDa98, Wesk98]. Dies schränkt, neben anderen wichtigen fehlenden Fähigkeiten (s. [JaBS97, Reic00]), ihre breite Einsetzbarkeit ganz erheblich ein [DaRK00].

In diesem Beitrag geben wir einen Überblick zum ADEPT-Projekt. Es wurde 1994 mit dem Ziel initiiert, alle relevanten Aspekte von WfMS im Zusammenwirken zu untersuchen und eine umfassende theoretische und konzeptionelle Grundlage für die ganzheitliche Realisierung eines solchen Systems zu erarbeiten.

Der Abschnitt 2 erläutert wichtige technologische Herausforderungen für WfMS der nächsten Generation. In Abschnitt 3 zeigen wir exemplarisch, wie einige von ihnen im ADEPT-Projekt aufgegriffen werden und welche Konzepte hieraus hervorgegangen sind. Der Beitrag schließt mit einer kurzen Diskussion (Kapitel 4) und Zusammenfassung (Kapitel 5).

## 2 Technologische Herausforderungen

Workflow-Management-Systeme (WfMS) müssen in der Lage sein, ein breites Spektrum an Geschäftsprozessen flexibel und effizient zu unterstützen. Wichtige Herausforderungen in diesem Zusammenhang sind:

1. Das WfMS muss einen ausdrucksstarken Formalismus für die Prozessmodellierung (sog. Prozessmetamodell) bereitstellen, der für Entwerfer verständlich ist und der die konsistente Beschreibung aller relevanten Prozessaspekte (Kontroll- und Datenfluss, zeitliche Beschränkungen, organisatorische Aspekte, usw.) gestattet.
2. Das WfMS muss ein zuverlässiges und robustes Ausführungsverhalten besitzen. Aus diesem Grund sollte bereits zur Modellierungszeit ausgeschlossen werden, dass es während der Ausführung von Workflow-Instanzen infolge fehlerhafter Prozessmodelle zu ungewollten „Überraschungen“ kommen kann. Beispiele hierfür sind Blockierungen, nicht erfüllbare Zeitconstraints und der Aufruf von Programm-Moduln bei fehlenden oder unvollständigen Parameterdaten.
3. In vielen Anwendungsumgebungen ist jedes starre System, auch bei ansonsten idealer Prozessunterstützung, zum Scheitern verurteilt. Krankenhausprozesse etwa weisen eine hohe Variabilität und Dynamik auf, so dass üblicherweise nicht alle Prozessvarianten vormodelliert werden können [DaRK00, RDMK00]. Hinzu kommt, dass unvorhersehbare Ausnahmen im Verlauf der Ausführung von Klinikprozessen eher den Normalfall bilden. Anwender müssen deshalb im Einzelfall spontan vom geplanten Ablauf, d. h. vom modellierten Prozess, abweichen können, etwa durch Auslassen, Einfügen oder Verschieben von Prozessschritten.
4. Solche Ad-hoc-Änderungen dürfen niemals zu Konsistenz- oder Korrektheitsverletzungen führen. Das bedeutet, dass prozessorientierte Anwendungen auch im Anschluss an Änderungen robust und stabil laufen müssen. Die hierzu notwendigen On-the-fly-Modellanalysen müssen effizient durchführbar sein, was mit zunehmender Ausdrucksmächtigkeit des verwendeten Workflow-Metamodells jedoch immer schwieriger wird.
5. Prozessorientierte Anwendungen müssen – auch im Fall von Ad-hoc-Abweichungen – für EDV-Laien einfach bedienbar sein. Insbesondere sollte die mit der Festlegung einer Ad-hoc-Änderung verknüpfte Komplexität vor ihnen verborgen bleiben, etwa in Bezug auf das Re-Mapping der Ein-/Ausgabeparameter der von der Änderung betroffenen Programm-Module

oder die Behandlung fehlender Daten nach dem Löschen von Prozessschritten. Ebenso wenig sollte sich der Benutzer um die Anpassung von Prozesszuständen kümmern müssen. Stattdessen muss das WfMS geeignete Modifikationsoperatoren auf einem hohen Abstraktionsniveau anbieten (kein Pop-Up eines Modell-Editors).

6. Die Unterstützung und Überwachung komplexer Zeitbedingungen (z. B. minimale / maximale Zeitabstände zwischen Aktivitäten) ist für viele Anwendungen essentiell. Ein WfMS muss deshalb über Fähigkeiten zur Überwachung von Terminen und Terminabhängigkeiten verfügen. Die fristgerechte Durchführung von Tätigkeiten muss durch Hinweise auf drohende Terminverletzungen unterstützt werden, auch im Kontext von gewünschten Ad-hoc-Abweichungen.
7. Von einem WfMS werden häufig Daten verarbeitet, an die hinsichtlich Datenschutz und Datensicherheit strenge Maßstäbe angelegt werden müssen. Die Rechte einer Person für den Zugriff auf diese Daten sind dabei über die Rolle oder Funktion geregelt, welche sie gerade einnimmt. Bereits im statischen Fall sind die realen Rollenverteilungen und Vertretungsregelungen häufig sehr komplex und stellen hohe Anforderungen an das WfMS, insbesondere im Hinblick auf die Pflege von Organisationsmodellen.

Im dynamischen Fall kommt erschwerend hinzu, dass durch Änderungen des Ablaufgraphen keine "Datenschutzlücken" entstehen dürfen. Aus diesem Grund muss sehr fein steuerbar sein, wer in welcher Rolle und in welchen Prozesszuständen welche Änderungen durchführen bzw. welche Informationen (z. B. Ausführungshistorie, Änderunghistorie usw.) abrufen darf. Entsprechende Einschränkungen sind auch deshalb sinnvoll, weil einzelne Anwender oft nur eine eingeschränkte Sichtweise auf Abläufe besitzen. Dadurch können Prozessänderungen, die aus der Sicht des Einzelnen durchaus sinnvoll erscheinen, im Widerspruch zu übergeordneten Interessen (Einhaltung von Fristen, organisatorische Regelungen usw.) stehen. Diese Problematik ist bei unternehmensweiten und -übergreifenden Prozessen verstärkt ausgeprägt.

8. Um unternehmensweite und -übergreifende Prozesse [DaRe99] angemessen zu unterstützen, muss das WfMS auch bei einer großen Anzahl von Benutzern und Prozessinstanzen performant arbeiten.

Obwohl bereits sehr umfangreich, ist diese Liste noch nicht vollständig. Weitere wichtige Anforderungen betreffen die Evolution von Prozessschemata und – soweit sinnvoll und möglich – die Propagierung dieser Änderungen auf bereits laufende Workflow-Instanzen, die Handhabung von Interprozess-Abhängigkeiten, die komponentenbasierte Entwicklung von prozessorientierten Anwendungen oder das semantische Rollback von Workflow-Instanzen beim Auftreten logischer Fehler. Zukünftige WfMS werden, sofern sie diesen Anforderungen gerecht werden, in vielen anspruchsvollen Anwendungsdomänen einsetzbar sein. Im Rahmen des ADEPT-Projektes arbeiten wir seit 1994 intensiv an den technologischen Grundlagen und der Entwicklung eines WfMS der nächsten Generation, das alle diese Aspekte (und noch einige mehr) ganzheitlich und sehr grundlegend adressiert.

### 3 Realisierung mit dem ADEPT-WfMS

Den skizzierten Anforderungen kann man nicht dadurch gerecht werden, indem man sie isoliert voneinander betrachtet. In ADEPT versuchen wir deshalb, die verschiedenen Facetten prozessorientierter Anwendungen integriert zu behandeln: Benutzerschnittstellen, Modellierungsfragestellungen und -werkzeuge, planbare / nicht planbare Ausnahmebehandlungen, Flexibilität und dynamische Prozess-

änderungen, Schemaevolution, temporale Aspekte, Inter-Prozess-Abhängigkeiten und Skalierbarkeit. Aus Platzgründen können wir an dieser Stelle nicht auf alle diese Aspekte eingehen. Stattdessen konzentrieren wir uns auf dynamische Prozessänderungen und Skalierbarkeitsfragestellungen. Weitergehende Informationen findet man in [Baue01, BaDa00, BaRD01, HRB+00, ReDa98, Reic00].

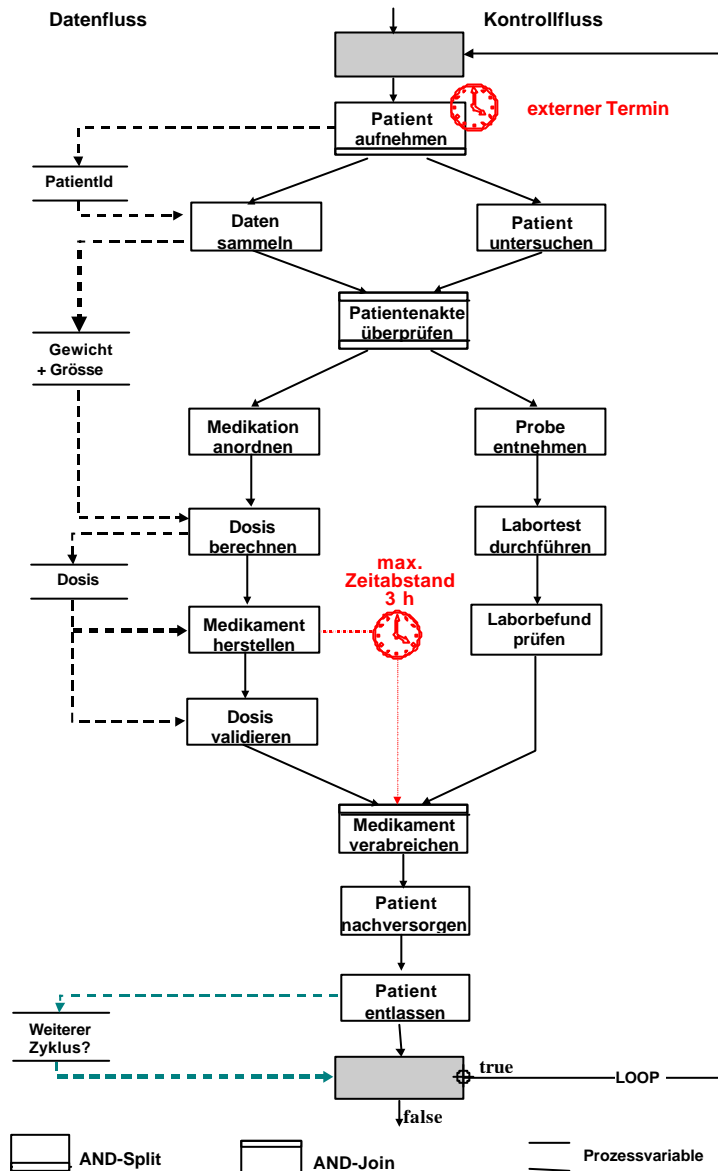


Abb. 3: Workflow-Modellierung in ADEPT

### 3.1 Prozessmodellierung

ADEPT stellt für den Prozessmodellierer ein ausdrucksstarkes Prozessmetamodell bereit, das es erlaubt, Geschäftsprozesse möglichst natürlich und in einer für die Anwender verständlichen Form abzubilden. Darüber hinaus ist die effiziente Überprüfung bzw. Sicherstellung wichtiger Modelleigenschaften möglich, etwa im Hinblick auf die Aktivierbarkeit von Prozessschritten, die

Verklemmungsfreiheit des modellierten Prozesses oder die Korrektheit der definierten Datenflüsse. ADEPT verwendet dazu das graphbasierte Prozessmetamodell  $ADEPT_{base}$ , das die integrierte Beschreibung der verschiedenen Aspekte eines Prozesses ermöglicht [RDMK00, Reic00]. Für die Kontrollflussmodellierung wird ein blockbasierter Beschreibungsansatz verfolgt, bei dem Sequenzen, Verzweigungen und Schleifen als logische Blöcke mit jeweils genau einem Ein- und Ausgangsknoten modelliert werden. Solche Kontrollblöcke können geschachtelt sein, dürfen sich aber nicht überlappen. Um die Ausdrucksmächtigkeit des Metamodells zu erhöhen, werden zusätzliche Konstrukte angeboten, etwa zur Beschreibung verschiedener Arten von „Wartet-Auf“-Beziehungen zwischen Aktivitäten paralleler Zweige oder zur Vormodellierung ausnahmebedingter Vorwärts- und Rückwärtssprünge im Kontrollfluss (z. B. partielles Rollback der Workflow-Instanz). Desweiteren stellt  $ADEPT_{base}$  auch Konstrukte zur Modellierung von Datenflüssen, zeitlichen Einschränkungen (z. B. Mindest- oder Maximalzeitabstände zwischen Aktivitäten) und organisatorischen Aspekten bereit. Ein einfaches Beispiel eines ADEPT-Prozessmodells zeigt Abbildung 3.

Auf Ausführungsebene wird jede Workflow-Instanz um Zustandsinformationen angereichert, die von der ADEPT *Process Engine* für die Workflow-Steuerung benötigt werden. Zur Unterstützung der Kontrollflusssteuerung werden den Knoten und Kanten des Ausführungsgraphen einer Workflow-Instanz entsprechende Zustandsmarkierungen zugeordnet. Für ihre Handhabung gibt es wohldefinierte Regeln, die festlegen, unter welchen Graphmarkierungen eine Prozessaktivität aktiviert werden darf und welche Folgemarkierungen sich im Anschluss an ihre Ausführung ergeben können [Reic00]. Dabei bleiben die Zustandsmarkierungen bereits abgearbeiteter bzw. nicht mehr ausführbarer Bereiche des Prozessgraphen beim Fortschreiten der Bearbeitung erhalten, zumindest solange die entsprechenden Graphregionen nicht wiederholt (z. B. nach Schleifenrücksprüngen) durchlaufen werden. Dadurch können Informationen zum bisherigen Verlauf der Prozessausführung direkt aus den aktuellen Graphmarkierungen abgeleitet werden, was für die effiziente Überprüfung der Anwendbarkeit dynamischer Änderungsoperationen vorteilhaft ist [ReDa98].

### 3.2 Unterstützung dynamischer Prozessänderungen

Berechtigte Anwender können in ADEPT dynamische Prozessänderungen auf einem hohen Abstraktionsniveau festlegen. Die Grundlage hierfür bildet das  $ADEPT_{flex}$ -Kalkül, das einen vollständigen Satz an Modifikationsoperatoren anbietet, mit denen sich die Struktur, die Attribute oder der Status von sich in Ausführung befindlichen Workflow-Instanzen abändern lassen [ReDa98].

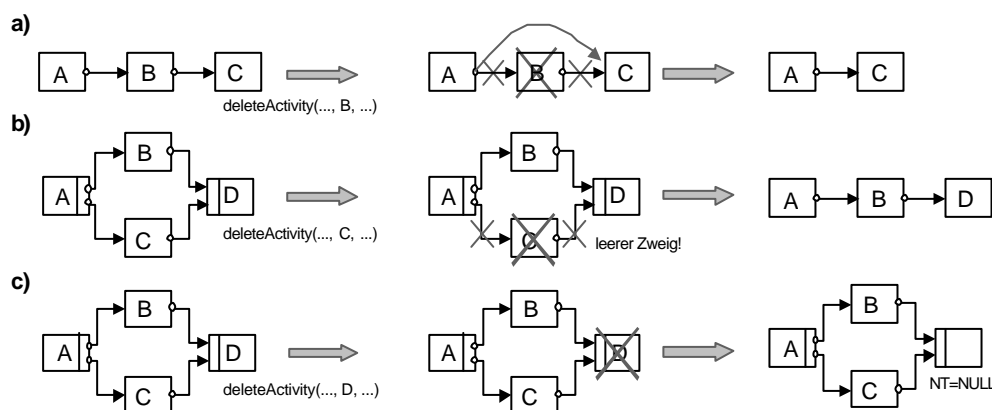


Abb. 4: Beispiele für das Löschen von Aktivitäten in ADEPT

Unterstützt werden Operationen für das Hinzufügen, Löschen (vgl. Abbildung 4) und Verschieben einzelner Aktivitäten oder ganzer Kontrollblöcke, für die Abänderung von Datenflüssen, für die Modifikation einzelner Ausführungsattribute (z. B. Ressourcen-/Bearbeiterzuordnungen, Verzweigungsprädikate) und für das kontrollierte Zurücksetzen der Prozessbearbeitung. Durch ihre kombinierte Anwendung können semantisch höherwertige Änderungen realisiert werden, etwa für das dynamische Einfügen einer Aktivität zwischen zwei Knotenmengen oder für die Umsetzung von Ad-hoc-Vorwärtssprüngen (z. B. vorzeitige Bearbeitung einer Aktivität mit Nachholen der übersprungenen Schritte).

Workflow-Instanz-Änderungen sind in ADEPT nur zulässig, wenn sie die Konsistenz des Prozesses nicht verletzen. Zur Erfüllung dieser Forderung müssen alle Aspekte des Prozesses und ihre Wechselwirkungen beachtet werden, selbst wenn sie nicht unmittelbar Gegenstand der beabsichtigten Modifikation sind. Bei Änderungen der Kontrollflussstruktur sorgt ADEPT z. B. selbständig dafür, dass die Blockstrukturierung aufrechterhalten bleibt und dass keine Verklemmungssituationen eintreten. Dies wird durch die Vorgabe geeigneter Vor- und Nachbedingungen für die Anwendung von Änderungsoperationen erreicht. Des Weiteren wird bei Anwendung von Änderungsoperationen, wie dem Einfügen, Löschen oder Verschieben von Aktivitäten (inkl. ihrer Schreizugriffe auf Prozessvariablen), sichergestellt, dass im weiteren Verlauf der Prozessausführung keine Inkonsistenzen (z. B. Lost Updates) oder Fehler (z. B. Aufruf von Aktivitätenprogrammen mit unvollständig versorgten Eingabeparametern) auftreten. Sind z. B. obligate Eingabeparameter einer Aktivität nach dem Löschen einer Vorgängeraktivität nicht mehr versorgt, muss die Löschoption entweder abgebrochen werden oder es sind begleitende Anpassungen vorzunehmen (z. B. kaskadierendes Löschen datenabhängiger Schritte, Installation von Nachforderungsdiensten, die bei der Aktivierung des datenabhängigen Schrittes gerufen werden, usw.).

ADEPT<sub>flex</sub> sorgt selbständig dafür, dass nach der Anwendung einer dynamischen Änderung wieder ein Prozess mit konsistentem Zustand resultiert. Um dies sicherzustellen, schränken wir die Anwendbarkeit von Änderungsoperationen durch den aktuellen Zustand der jeweiligen Workflow-Instanz (d. h. den Knoten- und Kantenmarkierungen ihres Ausführungsgraphen) ein. Beispielsweise dürfen bereits abgeschlossene Aktivitäten nicht mehr gelöscht oder ihre Ausführungsattribute nachträglich verändert werden. Ebenso wenig darf eine neue Prozessaktivität in einen bereits abgearbeiteten Teil eines Ausführungsgraphen eingefügt werden. Entsprechende Statusüberprüfungen werden *vor* Anwendung einer Änderungsoperation bzw. -transaktion durchgeführt. Darüber hinaus wird *nach* der strukturellen Abänderung eines Ausführungsgraphen sein Status neu bewertet und ggf. angepasst, z. B. durch die Zuordnung von Markierungen zu neu eingefügten Knoten und Kanten. Anschließend kann mit der Prozessausführung in einem konsistenten Zustand fortgefahren werden.

Das ADEPT-WfMS stellt ebenfalls sicher, dass bei der Anwendung einer Änderungstransaktion wieder ein Prozess mit konsistentem, d.h. erfüllbarem Zeitplan resultiert. Die Einhaltung dieser Forderung wird nicht nur bei der Änderung von Zeitattributen geprüft, sondern auch in Verbindung mit Modifikationen der Kontrollflussstruktur. Beispielsweise können sich durch das Einfügen einer Aktivität (mit bekannter minimaler und maximaler Zeitdauer), die frühesten Anfangszeitpunkte nachfolgender Aktivitäten nach hinten verschieben. Je nachdem, ob dadurch zuvor festgelegte Termine noch erfüllbar sind oder nicht, können die notwendigen Anpassungen automatisch durch das WfMS erfolgen, oder sie müssen in Interaktion mit den Anwendern ermittelt werden.



### 3.3 Skalierbarkeit durch verteilte Prozesssteuerung

Unternehmensweite, prozessorientierte Anwendungen sind charakterisiert durch eine große Zahl von Benutzern und gleichzeitig aktiven Workflow-Instanzen [DaRe00]. Dadurch kann die Belastung für Server sehr groß werden. Bereits die Durchführung einer Prozessaktivität macht die Übertragung mehrerer Nachrichten zwischen Server und Klienten erforderlich, etwa um Parameterdaten zu transferieren, Arbeitslisten zu aktualisieren, Programm-Module zu rufen oder Daten zwischen gerufenen Modulen und externen Datenquellen auszutauschen. Insgesamt kann das zu bewältigende Kommunikationsaufkommen sehr groß werden, was bei zunehmender Anzahl von Workflow-Instanzen zur Überlastung von Teilnetzen und zu schlechten Antwortzeiten führt. Erschwerend kommt für unternehmensübergreifende Prozesse hinzu, dass die Kommunikation teilweise über relativ langsame Weitverkehrsnetze abgewickelt werden muss. Spätestens dann stellt die Belastung des Kommunikationssystems einen kritischen Faktor für den Gesamtdurchsatz des Systems dar [Baue01].

Um die im unternehmensweiten Einsatz aufkommende Gesamtlast bewältigen zu können, muss ein WfMS skalierbar sein. In ADEPT erzielen wir Skalierbarkeit, indem wir die Systemlast auf mehrere Server verteilen. Zu diesem Zweck haben wir mit ADEPT<sub>distribution</sub> ein Ausführungsmodell entwickelt und implementiert, bei dem die Kontrolle einer Workflow-Instanz ggf. abschnittsweise durch verschiedene Server erfolgen kann [BaDa00]. Zu diesem Zweck wird der Ablaufgraph einer Prozessvorlage in *Partitionen* unterteilt (vgl. Abbildung 5). Jeder Partition wird ein Server zugeordnet, der zur Laufzeit die Koordination ihrer Aktivitäten übernimmt. Kommt es während der Ausführung einer Workflow-Instanz zu einem Übergang zwischen zwei Partitionen, so findet eine *Migration* statt, bei der ihre Kontrolle vom aktuellen Server an den Server der Zielpartition übertragen wird. Bevor dieser Server mit der Ausführung fortfahren kann, müssen instanzspezifische Daten (z.B. Informationen zum aktuellen Ausführungsstatus, aktuelle Werte bestimmter Prozessvariablen) transferiert werden. Durch diesen Ansatz werden die Aktivitäten paralleler Bearbeitungswege ggf. nebenläufig durch unterschiedliche Server koordiniert. Um dabei die Kommunikation lokal zu halten, müssen diese Server in ADEPT nicht notwendigerweise Kenntnis über den Bearbeitungsstatus der von anderen Servern kontrollierten Teilwege bzw. Aktivitäten besitzen.

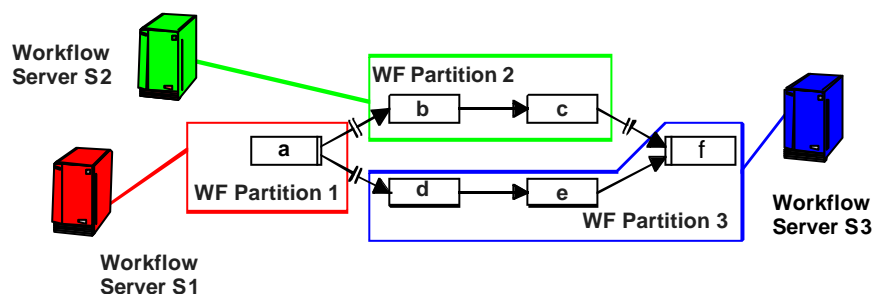


Abb. 5: Partitionierung von Workflows und Migration der Workflow-Kontrolle

Ein zentrales Anliegen des ADEPT<sub>distribution</sub>-Ansatzes ist es, die Partitionierung von Prozessgraphen und die Zuordnung von Servern so zu wählen, dass das Kommunikationsaufkommen bei der Prozessausführung minimiert wird. Zu diesem Zweck werden für den Entwickler Verfahren bereitgestellt, mit denen sich optimale Serverzuordnungen bzw. optimale Partitionierungen automatisch berechnen lassen. Dazu verwenden wir Kostenformeln, die es uns erlauben, die Güte

einer gewählten Verteilung zu bewerten. Berücksichtigt werden u.a. Kosten für den Transfer von Parameterdaten, für die Aktualisierung von Arbeitslisten sowie für die Durchführung von Migrationen. Dabei wird eine Aktivität in den meisten Fällen demjenigen Server zugeordnet, der sich am nächsten bei den zugehörigen Bearbeitern befindet. Da Migrationen ebenfalls Kommunikationskosten verursachen und die Server belasten, werden sie allerdings nur eingesetzt, wenn sie das Gesamtkommunikationsverhalten verbessern [Baue01].

## 4 Diskussion

In der Workflow-Literatur gibt es zahlreiche Veröffentlichungen, die sich mit einzelnen Aspekten von WfMS beschäftigen. Beispiele sind Arbeiten zu Modellierungssprachen (z. B. [Ober96]), zu Ad-hoc-Änderungen (z. B. [MüRa00, Wesk98]), zu Schemaevolution (z. B. [CaCP98, ElMa97, JoHe98, KrGe99]) oder zu Skalierbarkeit und verteilter Workflow-Ausführung (u.a. [AMG+95, MWW+98, ShKo97]). Jedoch gibt es kaum Projekte, welche diese Aspekte gemeinsam betrachten, insbesondere wird deren Zusammenspiel nicht hinreichend gewürdigt. Es ist nicht das Ziel dieser Arbeiten, ein bezüglich der Kommunikationskosten effizientes WfMS zu entwickeln, das funktional mächtig, skalierbar und flexibel ist. Diese Aspekte und ihr Zusammenwirken wurden in ADEPT erstmalig und in ganzheitlicher Weise untersucht (z. B. [BaRD01, DaRK00]).

WIDE erlaubt dynamische Änderungen eines Workflow-Schemas und deren Propagierung auf laufende Workflow-Instanzen [CaCP98]. Außerdem werden Workflow-Instanzen verteilt gesteuert [CGP+96]. Bei MOKASSIN [GJS+99, JoHe98] und WASA [Wesk98] wird die verteilte Workflow-Ausführung durch die zugrunde liegende CORBA-Infrastruktur realisiert. Außerdem sind Änderungen auf Schema- und Instanzebene möglich, wobei auch Konsistenzfragestellungen betrachtet werden. In INCAs [BaMR96] erfolgt die Steuerung von Workflow-Instanzen auf der Grundlage von Regeln. Die Regelmenge eines Workflows kann zur Laufzeit modifiziert werden, um dynamische Änderungen durchzuführen. – Bei all diesen Ansätzen wird aber, im Gegensatz zu ADEPT, nicht explizit auf das Zusammenspiel der erwähnten Aspekte (z. B. dynamische Änderungen und verteilte Steuerung) eingegangen.

## 5 Zusammenfassung

WfMS der nächsten Generation besitzen das Potenzial, die Entwicklung vorgangsorientierter Anwendungen nachhaltig zu verändern. Faktisch wird durch ihren Einsatz die Realisierung und der Betrieb prozessorientierter Anwendungssysteme im größeren Stil überhaupt erst möglich. Dabei sollten, wie im Fall von ADEPT, die einzelnen Programmbausteine eines Prozesses als isolierte, wiederverwendbare Komponenten implementiert werden können, deren Eingabeparameter beim Aufruf von der Laufzeitumgebung des WfMS versorgt werden und die lediglich dafür sorgen müssen, dass nach ihrer Beendigung korrekte Werte für Ausgabeparameter erzeugt werden. Alle anderen Aufgaben der Ablaufsteuerung und -überwachung (inkl. Ausnahme- und Fehlerbehandlungen) sollen durch das WfMS übernommen werden.

WfMS mit den skizzierten Eigenschaften bieten die Chance, zu einer gänzlich neuen Art der Entwicklung verteilter Informationssysteme zu gelangen, bei der Anwendungen durch die (graphische) Beschreibung von Prozessvorlagen und durch das Einstecken vorgefertigter Programmbausteine in

diese Vorlagen entwickelt werden. Spätere Prozessänderungen und daraus resultierende Anpassungen der Anwendungssysteme können bei diesem Ansatz relativ einfach durchgeführt werden. Wurde bei der Implementierung der Programmbausteine sorgfältig vorgegangen, kann z.B. die Reihenfolge der Arbeitsschritte eines Prozesses geändert oder es können neue Schritte hinzugenommen werden, ohne dass hiervon die bereits existierenden Programmbausteine betroffen sind. Schließlich können WfMS dazu beitragen, funktionsorientierte Anwendungen prozessorientiert zu integrieren und so eine gemeinsame Basis für die Verwaltung von Arbeitsabläufen zu schaffen.

Der ADEPT-Prototyp [HRB+00] ist derzeit eines der funktional mächtigsten und flexibelsten WfMS. Er weist die Implementierbarkeit und das Zusammenspiel der im ADEPT-Projekt entwickelten Konzepte auf eindrucksvolle Weise nach und zeigt, dass Flexibilität, Robustheit und Effizienz keine Widersprüche sein müssen. Der Prototyp zeigt aber auch, dass solche High-End-WfMS große Software-Systeme sind, die leicht die Code-Komplexität eines High-End-DBMS erreichen. Die derzeitige Implementierung des Prototypen umfaßt ca. 130.000 Programmzeilen Java-Code.

WfMS werden in Zukunft im Rahmen der Software-Infrastruktur von Unternehmen eine Schlüsselrolle einnehmen und sich als unverzichtbare, allgemein nutzbare Middleware-Funktion zur flexiblen Gestaltung und Steuerung von Geschäftsprozessen erweisen.

## Literatur

- [AMG+95] Alonso, G.; Mohan, C.; Günthör, R.; Agrawal, D.; El Abbadi, A.; Kamath, M.: Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management. Proc. IFIP Working Conf on Information Systems for Decentralised Organisations, Trondheim, August 1995.
- [BaDa00] Bauer, T.; Dadam, P.: Efficient Distributed Workflow Management Based on Variable Server Assignments. Proc. 12<sup>th</sup> Conf on Advanced Information Systems Engineering, Stockholm, Juni 2000, S. 94-109.
- [BaRD01] Bauer, T.; Reichert, M.; Dadam, P.: Adaptives und verteiltes Workflow-Management. Proc. Datenbanksysteme in Büro, Technik und Wissenschaft (BTW 2001), Oldenburg, März 2001, S. 47-66
- [BaMR96] Barbará, D.; Mehrotra, S.; Rusinkiewicz, M.: INCAs: Managing Dynamic Workflows in Distributed Environment, Journal of Database Mgmt., Vol. 7, No. 1, 1996, S. 5-15
- [Baue01] Bauer, T.: Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme. Dissertation, Universität Ulm, Februar 2001
- [CaCP98] Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G.: Workflow Evolution. Data & Knowledge Engineering, Vol. 24, No. 3, Januar 1998, S. 211-238
- [CGP+96] Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G.: WIDE: Workflow Model and Architecture. CTIT Technical Report 96-19, Universität Twente, 1996
- [DaRe99] Dadam, P.; Reichert, M. (eds.): Proc. Workshop on Enterprise-Wide and Cross-Enterprise Workflow-Management: Concepts, Systems, Applications, 29. Jahrestagung der GI (Informatik'99), Paderborn, Oktober 1999.
- [DaRK00] Dadam, P.; Reichert, M.; Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems? Proc. 4<sup>th</sup> Int'l Conf. on Business Information Systems (BIS'2000), Posen, April 2000, S. 36-59.

- [ElMa97] Ellis, C.; Maltzahn, C.: The Chautauqua Workflow System. Proc. 30<sup>th</sup> Hawaii Int'l Conf on System Sciences, Maui, Hawaii, 1997
- [GJS+99] Gronemann, B.; Joeris, G.; Scheil, S.; Steinfort, M.; Wache, H.: Supporting Cross-Organizational Engineering Processes by Distributed Collaborative Workflow Management – The MOKASSIN Approach. Proc. 2<sup>nd</sup> Symp on Concurrent Multidisciplinary Engineering, Bremen, September 1999
- [HRB+00] Hensinger, C.; Reichert, M.; Bauer, T.; Strzeletz, T.; Dadam, P.: ADEPT<sub>workflow</sub> – Advanced Workflow Technology for Adaptive, Enterprise-wide Processes. Demo-Proc of the 7<sup>th</sup> EDBT Conf., Konstanz, März 2000.
- [JaBS97] Jablonski, S.; Böhm, M.; Schulze, W. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen. Dpunkt, 1997
- [JoHe98] Joeris, G.; Herzog, O.: Managing Evolving Workflow Specifications. Proc. 3<sup>rd</sup> IFCIS Conf on Cooperative Information Systems, New York, August 1998
- [KrGe99] Kradolfer, M.; Geppert, A.: Dynamic Workflow Schema Evolution Based on Workflow Type Versioning and Workflow Migration. Proc. 4<sup>th</sup> IFCIS Int'l Conf on Cooperative Information Systems, Edinburgh, September 1999.
- [LeRo00] Leymann, F.; Roller, D.: Production Workflow – Concepts and Techniques. Prentice Hall, 2000.
- [MüRa00] Müller, R.; Rahm, E.: Dealing with Logical Failures for Collaborating Workflows. Proc. 5<sup>th</sup> Int'l Conf. Cooperative Inf. Sys., Eilat, September 2000, S. 210-223.
- [MWW+98] Muth, P.; Wodtke, D.; Weisfenfels, J.; Kotz-Dittrich, A.; Weikum, G.: From Centralized Workflow Specification to Distributed Workflow Execution. Journal of Intelligent Inf Systems, Vol. 10, No. 2, März 1998, S. 159-184.
- [Ober96] Oberweis, A.: Modellierung und Ausführung von Workflow mit Petri-Netzen. Teubner Verlag, 1996
- [RDMK00] Reichert, M.; Dadam, P.; Mangold, R.; Kreienberg, R.: Computerbasierte Unterstützung von Arbeitsabläufen im Krankenhaus – Konzepte, Technologien und deren Anwendung. Zentralbl Gynakol, Vol. 122, Januar 2000, S. 56–70.
- [ReDa98] Reichert, M.; Dadam, P.: ADEPT<sub>flex</sub> – Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems, Special Issue on Workflow Mgmt Sys, Vol. 10, No. 2, März 1998, S. 93–129
- [Reic00] Reichert, M.: Dynamische Ablaufänderungen in Workflow-Management-Systemen. Dissertation, Universität Ulm, Juli 2000
- [ShKo97] Sheth, A.; Kochut, K.: Workflow Applications to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration Systems. Proc. NATO Advanced Study Institute on Workflow Management Systems and Interoperability, Istanbul, August 1997, S. 12-21
- [Wesk98] Weske, M.: Flexible Modeling and Execution of Workflow Activities. Proc. 31<sup>st</sup> Hawaii Int'l Conf Sys Sciences, Software Technology Track, 1998, S. 713-722