

# Von ADEPT zur AristaFlow<sup>®</sup> BPM Suite – Eine Vision wird Realität

## "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen

**Peter Dadam<sup>1</sup>, Manfred Reichert<sup>1</sup>, Stefanie Rinderle-Ma<sup>1</sup>,  
Kevin Göser<sup>2</sup>, Ulrich Kreher<sup>2</sup>, Martin Jurisch<sup>2</sup>**

<sup>1</sup> Universität Ulm, Institut für Datenbanken und Informationssysteme (DBIS)  
{peter.dadam, manfred.reichert, stefanie.rinderle}@uni-ulm.de  
www.uni-ulm.de/dbis

<sup>2</sup> AristaFlow GmbH, Ulm  
{kevin.goeser, ulrich.kreher, martin.jurisch}@aristaflow.com  
www.AristaFlow.com

### Zusammenfassung

Angeregt durch ein Forschungsprojekt im Bereich klinischer Informationssysteme, wurde von uns Mitte der 90er Jahre das Forschungsprojekt ADEPT gestartet, welches im Bereich Prozess-Management das nahezu Unmögliche anstrebte und mittlerweile auch erreicht hat: Hochgradig flexible Ausführung von Unternehmensprozessen, Realisierung robuster prozessorientierter Anwendungen "per Konstruktion" sowie ein für alle Anwendergruppen (Prozess-Implementierer, Systemadministratoren, Endbenutzer) einfach zu benutzendes System. Dieser Beitrag beschreibt die Hintergründe des ADEPT-Projekts sowie unsere Motivation für die gesteckten Ziele, die von uns verfolgte Vision und deren vollständige Umsetzung in der nunmehr verfügbaren AristaFlow<sup>®</sup> BPM Suite.

## 1 Motivation

Infolge der Globalisierung der Märkte stehen heute alle Unternehmen unter einem enormen Druck, sich rasch auf neue Marktsituationen einzustellen [MRB08]. Erforderliche Reaktionen können z.B. sein, dass sie ihre Produkte und Dienstleistungen zu niedrigeren Kosten realisieren müssen, um diese zu wettbewerbsfähigen Preisen anbieten zu können, dass sie ihre Produkte und Dienstleistungen in kürzerer Zeit entwickeln und herstellen müssen, dass sie individualisierte Interaktionsformen für verschiedene Kundengruppen realisieren müssen oder dass sie die Entwicklung und Herstellung ihrer Produkte anders organisieren, weil Kunden eine entsprechende Partizipation verlangen.

Herausforderungen dieser Art – wenn auch unter anderen Vorzeichen – kennt man in Krankenhäusern und Universitätskliniken schon lange. Einerseits sehen sich diese einem hohen wirtschaftlichen Druck ausgesetzt, der sie zunehmend zwingt, ihre Abläufe zu optimieren und zu standardisieren. Andererseits lassen sich ihre Produkte (z.B. diagnostische Verfahren, chirurgische Eingriffe, Therapien, etc.) nicht in ein starres Konzept pressen, da sich die Voraussetzungen und Rahmenbedingungen von Patienten mehr oder weniger stark unterscheiden können. Dies ist auch der Grund, warum bis heute Workflow- und Prozess-Management-Systeme [ReDa00] in dieser Domäne nur in einem sehr geringem Umfang (wenn überhaupt) Fuß fassen konnten [LeRe07]. Und auch existierende Branchensoftware für Krankenhäuser wird den genannten Herausforderungen in keinster Weise gerecht, und bietet keine adäquate Unterstützung für Krankenhausprozesse.

Die Anforderungen an eine geeignete Technologie zur Unterstützung von Krankenhausprozessen sind allerdings auch enorm [DaRe00]. Einerseits müssen sich neue Prozesse einfach und schnell realisieren lassen, andererseits müssen sie anschließend robust und stabil laufen. Weiter will man eine gewisse Steuerung des Prozesses durch ein Informationssystem haben, bei Bedarf im Einzelfall aber vom vorgeplanten Ablauf abweichen können; und dies darf die robuste Ausführung natürlich nicht gefährden. Auch soll das System diesbezüglich sehr mächtig sein, und dennoch ist eine einfache Bedienbarkeit ein absolutes Muss. Erschwerend kommt zu alledem hinzu, dass man sich auf Systemebene nicht auf die einfache Weitergabe elektronischer Dokumente beschränken kann,

sondern dass sich beliebige Anwendungssysteme, z. B. ERP-Systeme, OP-Planungssysteme, Befundungssysteme und Laborautomaten, integrieren lassen müssen.

Als Mitte der 90er Jahre das ADEPT-Projekt gestartet wurde [Dada95], bildeten diese Herausforderungen aus dem klinischen Bereich die "Messlatte" für unsere Lösungsvorschläge. Zum einen flossen hier unsere Erfahrungen aus einem Forschungsprojekt im Bereich "Klinische Informationssysteme" [KRD93, Kuhn94, Kuhn94a] ein, das wir gemeinsam mit mehreren Partnern aus dem Universitätsklinikum Ulm durchgeführt haben. Zum anderen haben wir Mitte der 90er nochmals ein dediziertes Workflow-Forschungsprojekt durchgeführt, in dem wir alle relevanten Typen von Prozessen der Universitäts-Frauenklinik in Ulm systematisch erfasst und analysiert haben [RDMK00].

Diese systematische Untersuchung der fachlichen Anforderungen aus dem klinischen Bereich hatte zwei Hintergründe: Zum einen wollten wir uns bewusst mit der klinischen Realität auseinandersetzen (Credo: "Wir definieren kein Problem weg!"), andererseits wollten wir aber auch keine künstlich überzogenen Ansprüche an die Technologie stellen. Was den letzten Punkt angeht, konnten wir jedoch "beruhigt" sein. Je tiefer wir in die klinische Materie eintauchten, auf desto mehr technologische Herausforderungen sind wir gestoßen. Neben wirklich sehr hohen Anforderungen an die Flexibilität und Ausnahmebehandlungsmöglichkeiten prozessorientierter Krankenhausinformationssysteme, haben wir es in dieser Domäne z.B. mit sehr komplexen Organisationsstrukturen, anspruchsvollen Kompetenz- und Vertreterregelungen sowie ressourcenintensiven Prozessen zu tun. Die Sorge, dass wir in Bezug auf die von uns definierten fachlichen und technologischen Anforderungen evtl. "zu weit springen" könnten, erwies sich daher als völlig unbegründet [DRK00, DaRe00]. Es ging uns allerdings nie darum, eine spezielle Technologie für den klinischen Anwendungsbereich zu entwickeln, sondern es war uns von vornherein klar, dass eine Technologie für Prozess-Management-Systeme, welche diesen Bereich adäquat "bedienen" würde, auf breiter Front in verschiedensten Anwendungsbereichen einsetzbar sein wird (was sich später in diversen Projekten auch bewahrheitet hat [BBKK02, BKK04, Brei06, RüWa07, RüWa08]).

Dieser Beitrag gliedert sich wie folgt: Kapitel 2 diskutiert ausgewählte technologische Herausforderungen für prozessorientierte Informationssysteme. In Kapitel 3 beschreiben wir die von uns im ADEPT-Projekt verfolgte Forschungsvision, die Antworten auf diese technologischen Herausforderungen liefern soll. Dem schließt sich in Kapitel 4 eine Zusammenfassung ausgewählter Ergebnisse des ADEPT-Projekts an. Ferner beschreiben wir in diesem Kapitel den aktuellen Übergang vom Forschungsprototypen zur AristaFlow® BPM Suite. Kapitel 5 fasst den aktuellen Entwicklungsstand der AristaFlow® BPM Suite zusammen und Kapitel 6 erörtert deren Nutzungsmodalitäten. Kapitel 7 schließt mit einer kurzen Zusammenfassung und einem Ausblick.

## 2 Die Herausforderungen

Aus Platzgründen verzichten wir hier auf eine detaillierte Begründung der identifizierten technologischen Herausforderungen (siehe [DRK00, DaRe09] für die entsprechenden Hintergründe), sondern beschränken uns im Wesentlichen darauf, diese nachstehend zu beschreiben.

### Robustheit

Prozessorientierte Informationssysteme (POIS) sind inhärent komplexer als herkömmliche, funktionsorientierte Informationssysteme, da die prozessorientierte "Verschaltung" der Anwendungsfunktionen bzw. -komponenten eine zusätzliche Fehlerquelle darstellt. Fehler dieser Art können z.B. sein, dass der Prozess in einen Verklemmungszustand (Deadlock) gerät und nicht mehr weiterschalten kann, dass ein Zyklus fehlerhaft modelliert wurde, infolgedessen der Prozess in eine Endlosschleife gerät, oder dass zwingend erforderliche Aufrufparameter von Anwendungsfunktionen zum Aufrufzeitpunkt nicht versorgt sind. Weitere Fehler können durch ein zu "liberales" Prozess-Meta-Modell entstehen, das Modellierungen zulässt, die zur Ausführungszeit zu seltsamen Effekten führen (man kennt diesen Effekt von Programmiersprachen wie Assembler oder C).

Hierzu nur zwei kleine Beispiele: Abb. 1 zeigt einen Prozessgraphen, in dem vergessen wurde, den Kontrollkonnektor zwischen den Aktivitäten B und C zu zeichnen. Ein Prozess-Management-System, welches mehrere Startknoten pro Prozess erlaubt, würde zur Ausführungszeit auf beiden Teilgraphen (A,B) sowie (C,...,L) gleichzeitig mit der Ausführung beginnen. Dies entspricht im vorliegenden Fall nicht dem gewünschten Verhalten und kann z.B. zu inkonsistenten Daten oder zum Fehlschlag von Aktivitäten führen. In diesem einfachen Beispiel sieht man das Problem natürlich sofort auf den ersten Blick, bei komplexen Prozessgraphen mit dutzenden bis hunderten von

Aktivitäten (siehe z.B. [BBR06]), ist das Entdecken eines solchen Fehlers in der Regel eine mühsame und zeitraubende Angelegenheit.

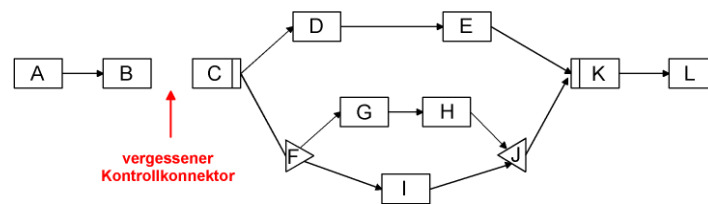


Abb. 1: Vergessener Kontrollkonnektor

Abb. 2 zeigt ein Prozessmodell, bei dem versehentlich der XOR-Split (Knoten F) mit einem AND-Join-Knoten (Knoten J) abgeschlossen wurde. Wird so etwas von einem Prozess-Management-System als Prozessmodell akzeptiert, hängt das Verhalten zur Ausführungszeit von der konkret implementierten Schaltlogik des Systems ab. Je nach System treten in einer solchen Situation ein Laufzeitfehler (Exception) oder ein Deadlock auf, oder aber die Prozessinstanz wird nach Ausführung von H bzw. I vom System als beendet betrachtet. Alle diese Verhaltensformen sind möglich und in vielen existierenden Systemen auch vorzufinden. Weitere Beispiele für Fehler dieser Art finden sich in [KoVa07, KoVa07a]. Es ist unschwer vorauszusehen, dass der Markterfolg von Prozess-Management-Systemen bzw. POIS ganz entscheidend davon abhängen wird, wie aufwendig es für die Prozessmodellierer jeweils ist, Fehler dieser Art zu entdecken bzw. zu vermeiden. Aufwendige Tests, wie bei konventioneller Software-Entwicklung oder Verwendung existierender Workflow-Management-Systeme üblich, sind aufgrund der zunehmenden Forderung nach rascher Implementier- bzw. Anpassbarkeit von POIS bei weitem nicht ausreichend und viel zu teuer!

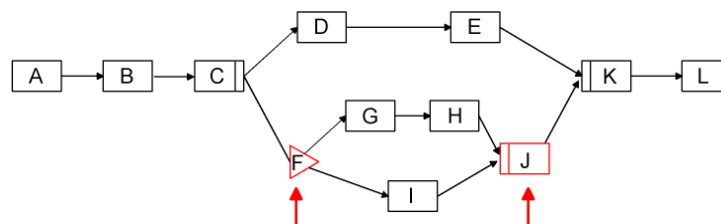


Abb. 2: Mismatch von XOR-Split- und -Join-Knoten

### Flexibilität und Adaptivität

Wie in Kapitel 1 erörtert, besteht die große Herausforderung für Unternehmen darin, sich rasch und flexibel auf sich verändernde Rahmenbedingungen einstellen zu können [RDB03, WRR08]. Ein Teilaspekt davon haben wir bereits unter "Robustheit" angesprochen, und zwar die Anforderung, rasch neue Prozessmodelle erstellen zu können, die anschließend robust und stabil und mit der vom Anwender intendierten Semantik ausführbar sind. Flexibilität bedeutet aber auch, im Einzelfall vom vorgeplanten Ablauf abweichen zu können, falls dies aus fachlicher Sicht nötig wird. Man stelle sich z.B. einen Patienten vor, bei dem aufgrund von Medikamentenunverträglichkeiten oder sonstigen Kontraindikationen gewisse Untersuchungen anders als ansonsten üblich durchgeführt werden müssen. Oder es wurde im Prozessmodell hinterlegt, dass Auftragsbestätigungen über 50.000 EUR stets von zwei Prokuristen digital unterschrieben werden müssen. Dies sei aber im konkret vorliegenden Fall infolge der Abwesenheit der in Frage kommenden Personen wegen Urlaub und Erkrankung technisch nicht realisierbar. Wenn z.B. von diesen Personen das telefonische OK kommt, dann sollte (analog zur "manuellen Praxis" ohne POIS) die Prozessausführung in diesem Einzelfall entsprechend abgeändert und die beiden digitalen Unterschriften entweder durch eine andere Maßnahme (z.B. einen entsprechenden Aktenvermerk) ersetzt oder später nachgeholt werden können.

Starre POIS, die es Anwendern trotz Autorisierung nicht gestatten, bei Bedarf ad hoc vom geplanten Ablauf (d.h. dem modellierten Prozess) abzuweichen, zwingen diese dann dazu, entweder am System vorbei zu arbeiten oder – noch schlimmer – die internen Systemzustände "händisch" zu manipulieren, um die gewünschte Änderung doch noch "irgendwie" realisieren zu können. Beides stellt aus fachlicher Sicht keine zufriedenstellende Lösung dar, ist infolge der Unzulänglichkeiten existierender Prozess-Management-Technologie und Branchensoftware aber oftmals der einzige Weg, um mit Ausnahmesituationen bzw. nicht vormodellierten Fällen umgehen zu können. Unterstützt das POIS dagegen Ad-hoc-Abweichungen auf Prozessinstanzebene, muss wiederum die Robustheit in

Bezug auf die weitere Ausführung dieser Prozessinstanz gewährleistet sein. Ad-hoc-Abweichungen ohne solche Zusicherungen anzubieten, wie in einigen Systemen erfolgt, grenzt an Fahrlässigkeit, da Benutzer in der Regel nur sehr begrenzten Einblick haben, welche systemseitigen Konsequenzen derartige Prozesseingriffe haben.

Ein weiterer Aspekt betrifft die Anpassung bestehender Prozessmodelle (Prozessschemas) an neue Gegebenheiten, etwa an Änderungen der gesetzlichen Rahmenbedingungen, an geänderte Interaktionen mit Kunden bzw. Geschäftspartner oder zur Behebung von Mängeln im bisherigen Prozessablauf [WRR08, WeRe08]. Bei kurz laufenden Prozessen wird man in der Regel die bereits in Ausführung befindlichen Prozessinstanzen nach dem alten Schema zu Ende kommen lassen und neue Prozessinstanzen auf Basis des neuen Schemas starten. Falls die Prozesse jedoch lang laufend sind (Wochen, Monate oder gar Jahre) oder das zu behebbende Problem gravierend ist, dann muss es zwingend eine Möglichkeit geben, die bereits in Ausführung befindlichen Prozessinstanzen auf das neue Schema zu migrieren (sog. *Prozess-Schema-Evolution* [RiDa03]). Natürlich muss auch hier systemseitig wieder garantiert werden können, dass migrierte Instanzen in der Folge keine Laufzeitfehler verursachen. Einige Systeme erlauben es zwar, ein Prozessschema mit sich in Ausführung befindlichen Prozessinstanzen strukturell zu modifizieren und die Ausführung aller Instanzen auf dem neuen Schema fortzusetzen; allerdings kann dies nachweislich zu schwerwiegenden Inkonsistenzen, Folgefehlern oder gar Systemabstürzen führen. Man stelle sich einen Krankenhausbetrieb vor, der infolge solcher technischen Unzulänglichkeiten behindert wird.

Schließlich darf es kein Entweder-Oder geben: Also entweder Ad-hoc-Änderungen auf Prozessinstanzebene oder Prozess-Schema-Evolution. Beide Änderungsformen sind für die Praxis zwingend erforderlich und müssen daher auch gemeinsam unterstützt werden.

### **Einfache Benutzbarkeit**

Die oben beschriebenen Anforderungen müssen erfüllt sein, wenn Prozess-Management-Systeme sich für Einsätze in nicht-trivialen, anspruchsvollen Anwendungsumgebungen empfehlen möchten. Ein breiter Einsatz dieser Technologie wird jedoch nur dann stattfinden können, wenn sowohl die Basis-Technologie (d.h. das Prozess-Management-System) als auch die mit ihr realisierten POIS einfach zu bedienen sind. Die Entwicklung der Datenbanktechnologie kann hier als Beispiel und Vorbild dienen. Die auf dem hierarchischen oder dem Netzwerk-Datenmodell basierenden Datenbanksysteme der ersten Generation (verfügbar etwa ab Ende der 60er Jahre) waren funktional gesehen bereits sehr mächtig. Sie konnten komplexe Datenstrukturen direkt abbilden und boten auch bereits diverse Möglichkeiten, um systemseitig die Integrität der Daten zu sichern, etwa durch Wertebereichsüberprüfungen oder durch Gewährleistung der referenziellen Integrität. Was diese Fähigkeiten angeht (und in Bezug auf Performanz sowieso), waren sie den relationalen Datenbanksystemen der ersten Stunde, die Anfang der 80er Jahre auf den Markt kamen, haushoch überlegen. Trotz dieser Fähigkeiten gestaltete sich die Marktdurchdringung der "alten" Datenbanksysteme relativ zäh, weil ein hoher Schulungsaufwand für die Anwendungsentwickler erforderlich war, um datenbankbasierte Anwendungen realisieren und pflegen zu können. Diesbezüglich waren die relationalen Datenbanken um Größenordnungen besser, der Gewinn an Produktivität auf Entwicklerseite war enorm. Außerdem waren die relationalen Datenbanksysteme in der Lage, nicht nur die anfänglichen Schwächen zu beseitigen, sondern auch noch eine erheblich größere Funktionalität als die Systeme der ersten Generation zu bieten. Als Folge wird diese Technologie heute für alle möglichen Arten von Anwendungen eingesetzt, von einfachen Einzelplatzanwendungen bis hin zu hochkomplexen, integrierten (ggf. sogar verteilten) Informationssystemen.

Fest steht: Nur wenn es gelingt, für Prozess-Management-Systeme einen ähnlichen Weg zu gehen, d.h. eine hohe Funktionalität und eine einfache Benutzbarkeit für die verschiedenen Benutzergruppen (Prozess-Implementierer, Service-Implementierer, System-Administratoren und Endbenutzer) zu realisieren, besteht das Potenzial, dass ihr einmal ein ähnlicher Erfolg beschieden sein wird.

## **3 Die ADEPT-Forschungsvision**

Wir konzentrieren uns im Folgenden auf Forschungs- bzw. Lösungsansätze, die wir im Rahmen des ADEPT-Projekts [Reic05] verfolgt haben und die technologische Antworten auf die vorangehend beschriebenen Herausforderungen bieten. Der Vollständigkeit halber sei erwähnt, dass das ADEPT-Projekt erheblich breiter angelegt war und zahlreiche weitere Aspekte von (flexiblen) Prozess-Management-Systemen einschloss: Unterstützung von Zeitaspekten [Grim97], Inter-Prozess-Koor-

dination [Hein00, Hein01], Zugriffskontrolle [PDA08, Webe05, RiRe05, RiRe07], Performanz- und Verteilungsaspekte [BaDa00, Baue01, BaRe02, BRD03, ReBa07], Architektur- und Systemimplementierungsfragen [RRD04b, Rind06, RRJK06, Reic08] sowie Implementierungsaspekte von Anwendungsfunktionen bzw. -diensten [Blas96, Brus99, AADR04, Atki06, AtDa07].

Die ADEPT-Forschungsvision Mitte der 90er Jahre kann man wie folgt charakterisieren:

1. Selbst anspruchsvolle Prozesse müssen rasch realisiert bzw. implementiert werden können. Weiter müssen Ad-hoc-Abweichungen zur Laufzeit (auf Prozessinstanzebene) möglich sein, ohne dadurch die robuste und effiziente Ausführung der Prozessinstanzen zu beeinträchtigen.
2. Das Prozess-Management-System sowie das darauf ggf. basierende POIS müssen für jedermann einfach zu bedienen sein: Prozess-Implementierer, Entwickler von Anwendungsfunktionen bzw. -services, Systemadministratoren und Endbenutzer.

Das klingt nach "eierlegender Wollmilchsau" und war in der Tat eine Funktionalität, die weit über das hinausging, was existierende Systeme und Vorschläge aus dem Forschungsbereich seinerzeit zu leisten im Stande waren.<sup>1</sup> Aber eines war uns von vornherein klar: Wenn diese Vision überhaupt realisierbar sein sollte, dann nur wenn es gelingt, die inhärent mit diesen Anforderungen verbundene Komplexität im Kern des Prozess-Management-Systems zu "verbergen" und "nach oben" semantisch hohe und einfach bedienbare Schnittstellen für die verschiedenen Benutzergruppen anzubieten.

Für den Prozessentwickler sollte die prozessorientierte Komposition von Anwendungsfunktionen im "Plug & Play"-Stil möglich sein (siehe Abb. 3). Bei dieser Komposition soll weitgehend davon abstrahiert werden, wie die den Prozess-Schritten zugeordneten Anwendungsfunktionen und Services implementiert sind. Darüber hinaus sollen durch systemseitige Prüfungen gewährleistet werden, dass das resultierende Prozessmodell strukturell keine unerwünschten Eigenschaften aufweist (z.B. isolierte Knoten oder Verklemmungen). Außerdem soll sichergestellt werden, dass die den verschiedenen Prozessschritten zugeordneten Anwendungsfunktionen bzw. -services in der gewünschten Anordnung auch korrekt "verschaltbar" sind, d.h. unter allen möglichen Prozessausführungen, die das Prozessmodell zulässt, stets alle nicht-optionalen Aufrufparameter mit Eingabewerten versorgt sein werden.

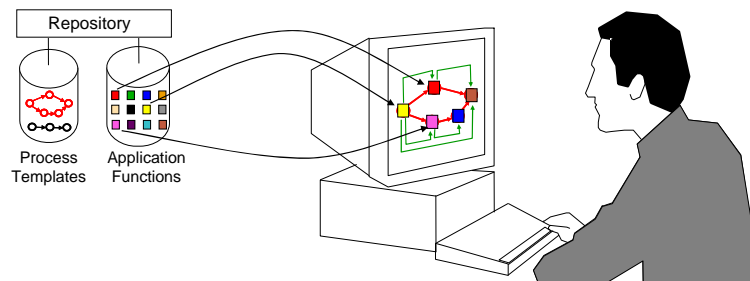


Abb. 3: Komposition von Prozessen mittels "Plug & Play" [Dada97]

Fertige Prozesse sollen "gekapselt" alles mitbringen, was zu ihrer Ausführung benötigt wird. D.h. es sollen beliebige Anwendungsfunktionen und -dienste integrierbar sein, einschließlich automatisch ausführbarer Dienste, interaktiver Anwendungsprogramme und existierender Anwendungen. Im Falle von interaktiven Benutzerschnittstellen sollen sich diese nahtlos in die bestehende Bedienoberfläche einfügen (kein "Fenster-über-Fenster-Terror!"). D.h. nach Übergabe an die Laufzeitumgebung des Prozess-Management-Systems sollen Prozesse ohne Installations- und Implementierungsaufwand ausführbar sein. Diese geforderte Eigenschaft hat dem Projekt auch seinen Namen gegeben: ADEPT = **A**pplication **D**evelopment based on **E**ncapsulated pre-modeled **P**rocess **T**emplates.

Das Prozess-Management-System soll es weiter erlauben, Prozessinstanzen während ihrer Ausführung individuell zu adaptieren (siehe Abb. 4). Entsprechende Änderungsoperationen sollen eine normale API-Funktionalität des Prozess-Management-Systems bilden, wobei systemseitig dieselben Korrektheitsprüfungen wie zur Entwicklungszeit durchgeführt werden sollten. Die Manipulation von Prozessinstanzen soll nach Möglichkeit das volle Spektrum an Änderungsoperationen umfassen und auf semantisch hoher Ebene erfolgen, so dass auch einfach zu benutzende, individuelle gestaltete Endbenutzerschnittstellen mit vertretbarem Implementierungsaufwand realisiert werden können.

<sup>1</sup> Eine ausführliche Würdigung des Stands der Technik zu Beginn des ADEPT-Projektes findet sich in [DaRe09]; einen umfassenden Vergleich derzeit existierender Ansätze bieten [WSR09] und [WRR08].

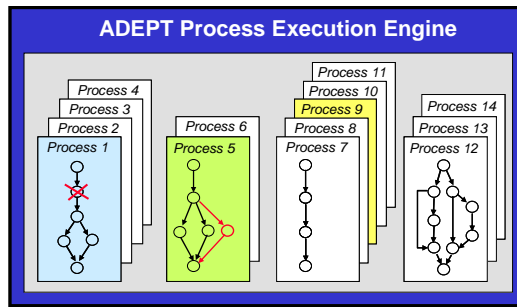


Abb. 4: Prozess-Engine mit individuell geänderten Prozessinstanzen

Abb. 5 illustriert, wie wir uns eine Ad-hoc-Abweichung durch den Endbenutzer vorgestellt haben:



a) Eine Ausnahmesituation tritt auf



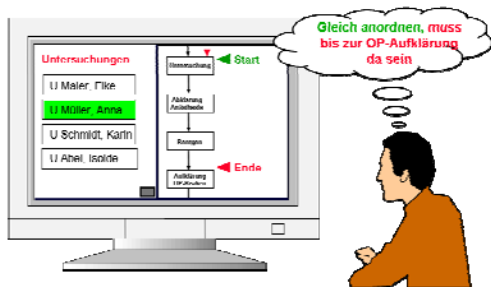
b) Benutzer betätigt den „Ausnahmeknopf“



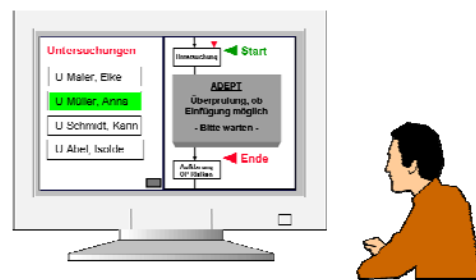
c) Benutzer wählt Art der Änderung



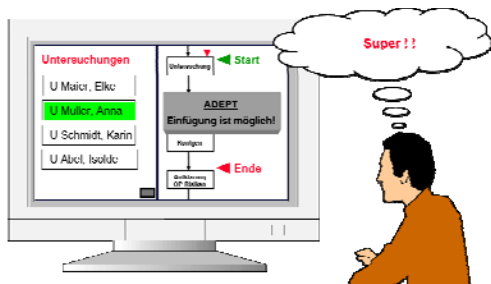
d) Benutzer wählt einzufügenden Schritt



e) Die Einbindung des Schrittes wird festgelegt



f) System prüft, ob Änderung zulässig



g) Die Änderung kann durchgeführt werden



h) Benutzer setzt Tätigkeit fort

Abb. 5: Durchführung einer Ad-hoc-Abweichung aus Benutzersicht [Dada06]

In diesem Beispiel (vgl. [Dada06]) wird angenommen, dass während der Ausführung einer einzelnen Prozessinstanz, etwa dem Behandlungsprozess eines bestimmten Risikopatienten, eine zusätzliche Laboruntersuchung benötigt wird, die aber in der zugehörigen Prozessvorlage an dieser Stelle im Prozess nicht vorgesehen ist (Abb. 5a). Es wird daher – wenn der Änderungswunsch systemseitig akzeptiert werden kann – die laufende Prozessinstanz (und nur diese!) individuell verändert. Nachdem der Benutzer den „Ausnahmeknopf“ betätigt hat (Abb. 5b), kann er die Art der gewünschten Ad-hoc-Änderung angeben (Abb. 5c). Soll eine Einfügung erfolgen, werden kontextabhängig die zur Verfügung stehenden Anwendungsfunktionen angezeigt (Abb. 5d). Dies können einfache Funktionen, wie „Brief schreiben“ oder „E-Mail versenden“ sein, aber auch komplexe Anwendungsdienste oder sogar ganze Prozesse. Der Benutzer muss jetzt noch angeben, nach welchem Schritt (bzw. welchen Schritten) die neue Aktivität dem zuständigen Bearbeiter angeboten werden soll und vor welchem Schritt (bzw. welchen Schritten) ihre Bearbeitung abgeschlossen sein muss (Abb. 5e). Nach Festlegung der Änderungen prüft das System die Zulässigkeit der resultierenden Abweichungen (Abb. 5f und Abb. 5g).

Auch eine *Prozess-Schema-Evolution* soll für die damit betrauten Anwender einfach durchführbar sein. Idealerweise sollte das Prozess-Management-System auf Basis des bisherigen Schemas, des neuen Schemas, dem aktuellen Zustand einer Prozessinstanz sowie ggf. vorgenommenen instanzspezifischen Ad-hoc-Änderungen selbstständig ableiten können, ob die betrachtete Instanz auf das neue Schema migriert werden kann oder nicht [RRD04, RRD04b, RRW08]. Keinesfalls sollen die Anwender direkt irgendwelche internen Instanzzustände manipulieren und natürlich sollen bei Migration von Instanzen auf das neue Schema dieselben Korrektheitskriterien gelten bzw. systemseitigen Prüfungen angewandt werden, wie bei der Prozesskomposition bzw. bei Ad-hoc-Abweichungen.

Abb. 6a - Abb. 6c illustrieren, wie wir uns den Ablauf einer Prozessschemaevolution aus Benutzersicht vorgestellt haben: Der Prozessmodellierer lädt das Prozessschema aus dem Prozess-Repository, nimmt dann die erforderlichen Änderungen an diesem Schema vor und erzeugt eine neue Schemaversion (vgl. Abb. 6a). Das System bestimmt die erforderlichen Vorbedingungen und Anpassungsmaßnahmen für die laufenden Prozessinstanzen dieses Schemas, soweit diese auf das neue Schema migriert werden können bzw. sollen (vgl. Abb. 6b + Abb. 6c). Auf Knopfdruck führt das System anschließend die Migration der ausgewählten Prozessinstanzen auf das neue Schema durch (vgl. Abb. 6d).

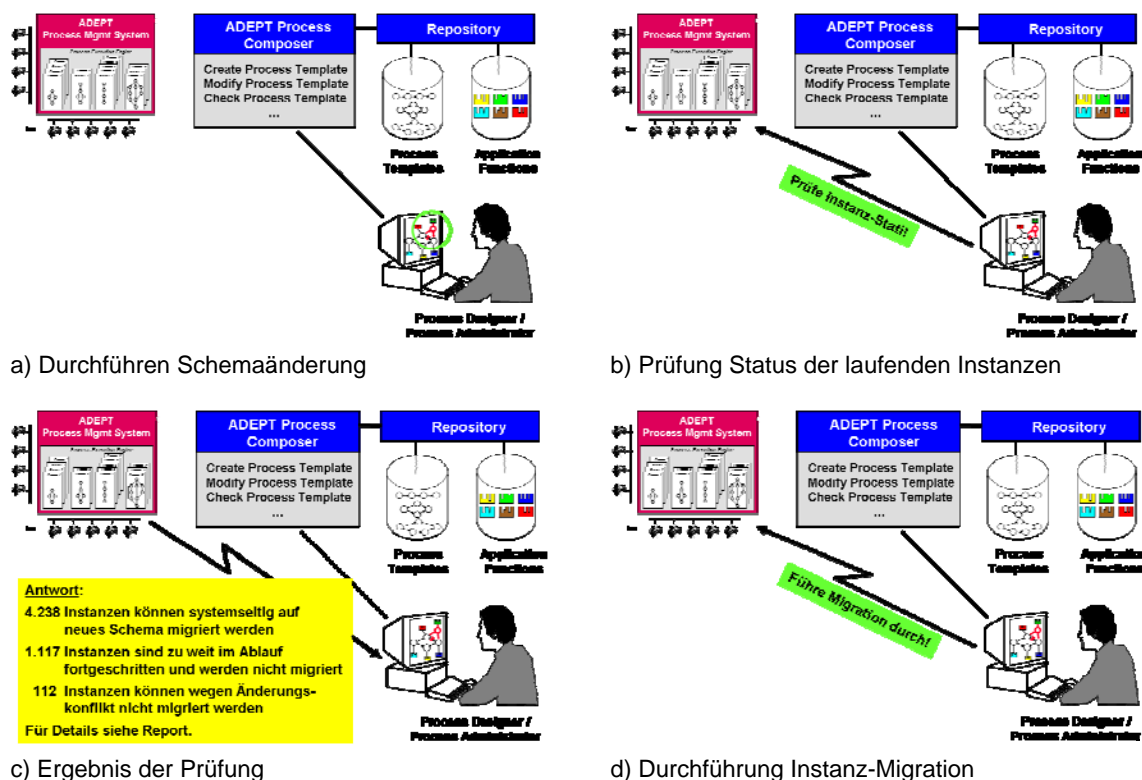


Abb. 6: Prozessschemaevolution [Dada06]

## 4 Die Umsetzung: Von ADEPT1 über ADEPT2 zur AristaFlow® BPM Suite

Wie bereits in Kapitel 3 erläutert, verfolgten wir das Ziel, die resultierende Komplexität von den Anwendern dadurch fernzuhalten, dass wir diese im Systemkern "verstecken" und "nach oben" funktional mächtige, aber dennoch einfach zu bedienende Schnittstellen (API) anbieten. Das Dilemma war, ein Prozess-Meta-Modell zu finden, das einerseits ausdrucksstark genug war, um die Realweltprozesse adäquat darstellen zu können, andererseits aber auch über eine solide Theoriebasis verfügt, um formal fundierte Aussagen über die "Korrektheit" eines Prozessmodells treffen zu können. Darüber hinaus musste es auch noch geeignet für Ad-hoc-Abweichungen sein und effiziente Korrektheitsanalysen (auch während der Prozessausführung) zulassen. Nachdem zu diesem Zeitpunkt kein Prozess-Meta-Modell existierte, das diesen Ansprüchen genüge (siehe [DaRe09] für eine ausführlichere Diskussion hierzu), entwickelten wir letztlich das ADEPT-Prozess-Meta-Modell (siehe Abb. 7), das alle gewünschten Eigenschaften in sich vereinigt: Es ist trotz der gewählten Blockstruktur aufgrund seiner verschiedenen Erweiterungen sehr ausdrucksstark, hat eine solide formale Basis und erlaubt effiziente Überprüfungen der erstellten Prozessmodelle auf Korrektheit. Im Kontext von Ad-hoc-Abweichungen ermöglicht es darüber hinaus, sehr rasch zu entscheiden, ob und wie eine gewünschte Änderungsoperation realisiert werden kann [ReDa98, Reic00].

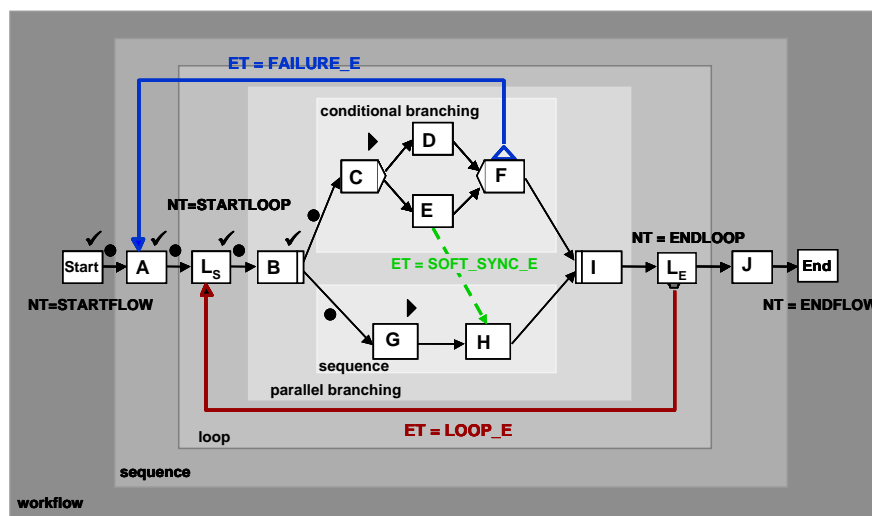


Abb. 7: ADEPT Prozess-Meta-Modell

Der entscheidende Durchbruch war die Entwicklung von semantisch hohen Änderungsoperationen zur Manipulation von Prozessgraphen sowohl bei der Prozessmodellierung als auch bei Ad-hoc-Abweichungen. Dies ermöglichte nicht nur die sehr viel einfachere und sicherere Manipulation von Prozessgraphen, sondern führte auch zu einer geänderten Vorgehensweise bei der Modellierung von Prozessen. Anstelle, wie sonst meist üblich, den Modellierer erst einmal den Prozess (mehr oder weniger) frei "malen" zu lassen, um anschließend systemseitig zu überprüfen, ob dieser syntaktisch korrekt ist, werden in ADEPT kontextabhängig nur solche Operationen zugelassen, die einen konsistenten Prozessgraphen wieder in einen neuen konsistenten Prozessgraphen überführen. Wir bezeichnen dieses Konstruktionsprinzip als "Correctness by Construction". Es findet auch auf Prozessinstanzebene im Kontext von Ad-hoc-Abweichungen seine Anwendung (und ist auch, wie sich noch zeigen sollte, im Zusammenhang mit der Prozess-Schema-Evolution sehr hilfreich).

Abb. 8 und Abb. 9 zeigen am Beispiel von ADEPT2 bzw. dem AristaFlow® Process Template Editor, wie – abhängig von der gewählten Markierung im Prozessgraph – der Prozessvorlagen-Editor die verschiedenen Änderungsoperationen kontextsensitiv zur Auswahl anbietet. In Abb. 8a ist nichts markiert, demzufolge sind in Abb. 9a keine Änderungsoperationen wählbar, außer der hier nicht zu sehenden Operation "Insert Data Element" (d.h. die Operation zur Einfügung globaler Prozessvariablen), die stets wählbar ist. In Abb. 8b wurde ein Knoten im Prozessgraph markiert, demzufolge schaltet der Prozessvorlagen-Editor alle Änderungsoperationen frei, deren Wirkung durch die Markierung des einzelnen Knotens eindeutig festgelegt ist, und zwar "Insert Surrounding {AND, XOR, LOOP} Block" und Delete Node (siehe Abb. 9b). In Abb. 8c wurden zwei aufeinanderfolgende



Knoten markiert, wobei die grüne Färbung<sup>2</sup> den Anfang und die blaue<sup>3</sup> das Ende des markierten Bereichs anzeigt. Nun kann zusätzlich zu vorher noch ein AND-, XOR- oder LOOP-Block zwischen die beiden Knoten eingefügt werden. Allerdings ist "Delete Node" dann nicht mehr wählbar (siehe Abb. 9c), weil unklar wäre, welcher der beiden markierten Knoten gemeint ist.

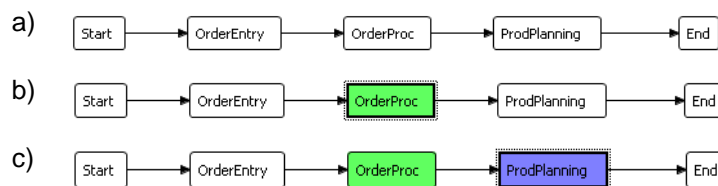


Abb. 8: Markierungen im Prozessgraph

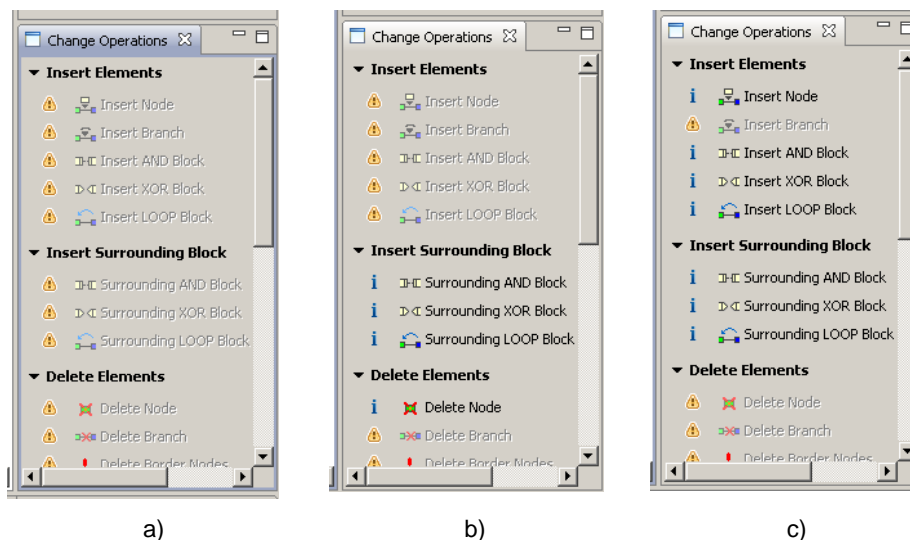


Abb. 9: Freigeschaltete Änderungsoperationen

1998 wurde von uns das erste ADEPT-System fertig gestellt (im Folgenden als "ADEPT1" bezeichnet). Es realisierte bereits das "Correctness by Construction"-Prinzip [Reic00], eine Palette von Operationen für Ad-hoc-Abweichungen [ReDa98] sowie die Handhabung zeitlicher Constraints [Grim97]. Von ADEPT1 wurden in den Folgejahren noch verschiedene Systemvarianten abgeleitet [RRD03], etwa zur Unterstützung verteilter Prozess-Ausführungen (unter voller Beibehaltung der Ad-hoc-Flexibilität) [BaDa00, Baue01, BaRe02, ReBa07] und zur Interprozess-Koordination [Hein00, Hein01]. Darüber hinaus wurde ADEPT1 in vielen anspruchsvollen Forschungsprojekten mit großem Erfolg eingesetzt [BBKK02, BKK04, GoGa05, Gola05, Grei05, Grei06, MoAn07, RWRW05, WRWR05]. Hieraus konnten wir wertvolle Erkenntnisse sowie zahlreiche fachliche Anforderungen für die Weiterentwicklung der ADEPT-Technologie gewinnen.

Im Jahr 2001 begannen wir dann, uns intensiv mit dem Thema "Prozess-Schema-Evolution" zu befassen. Unser Ziel war es, weitgehend systemseitig bestimmen zu können, welche Prozessinstanzen auf das neue Schema migriert werden können und welche nicht (vgl. Abb. 6). Auch hier hatten wir wieder den Anspruch, eine umfassende und anwendungsneutrale Lösung zu entwickeln. Die formalen Grundlagen und methodischen Konzepte wurden wieder systemunabhängig entwickelt [Rind04, RRD04a]. Allerdings zeigte sich, dass die spezifischen Eigenschaften des ADEPT-Prozess-Meta-Modells in Verbindung mit den semantisch hohen Änderungsoperationen sehr hilfreich für eine effiziente Realisierung und die Implementierung einfach benutzbarer Schnittstellen ist. Wie anhand eines "Proof-of-Concept"-Prototyps [RRD04] gezeigt wurde, kann man basierend auf den Ergebnissen dieser Arbeiten eine Lösung implementieren, die im Wesentlichen der in Abb. 6 dargestellten Vorgehensweise entspricht.

<sup>2</sup> hellgrau in einen Graustufen-Ausdruck  
<sup>3</sup> dunkelgrau in einem Graustufen-Ausdruck

Im Jahr 2004 starteten wir dann – zusammen mit der Universität Mannheim und verschiedenen Industriepartnern – das AristaFlow-Projekt.<sup>4</sup> Ein wesentliches Ziel dieses Projektes war die Untersuchung, wie der Entwurf und die Entwicklung von Anwendungsfunktionen gestaltet werden sollte, damit die für das Plug & Play erforderlichen Informationen möglichst ohne Zusatzaufwand abgeleitet werden können. Ferner sollte untersucht werden, was in diesem Zusammenhang ggf. noch geleistet werden muss, um ein noch höheres Maß an Robustheit für die hierauf aufbauenden Prozessimplementierungen zu erzielen [AADR04, Atki06, AtDa07, ABFJ08, AtSt08, AtSt08a]. Ein anderes wesentliches Ziel, und hier lag der eigentliche Schwerpunkt des Projektes, war der Entwurf eines "Next-Generation"-Prozess-Management-Systems ("ADEPT2"), das in umfassender und integrierter Weise die Prozesskomposition im "Plug & Play"-Stil, "Correctness by Construction", Ad-hoc-Flexibilität sowie Prozess-Schema-Evolution zu unterstützen vermag. Dieser Entwurf sollte mit den Praxispartnern im industriellen Kontext unter verschiedenen Aspekten validiert werden.

Wir hatten beim Entwurf von ADEPT2 den Ehrgeiz, die gewünschte Funktionalität nicht nur "irgendwie" zu realisieren, um dann anschließend (wieder) ein schlecht wartbares Experimentalsystem zu haben, sondern das System so zu konzipieren, dass es trotz der enormen funktionalen Mächtigkeit immer noch schlank, wartbar und einfach erweiterbar bleibt. Dies resultierte letztlich in einer serviceorientierten Architektur (siehe Abb. 10), bei deren Entwurf sehr darauf geachtet wurde, die bereitgestellte Funktionalität – wo immer möglich – generisch zu implementieren, um sie an verschiedener Stelle im System sowie auf API-Ebene nutzen zu können. So verwenden z.B. der Prozessvorlagen-Editor, der Workflow-Client für das Monitoring von Prozess-Instanzen sowie der Client für die Visualisierung und Ad-hoc-Änderung einzelner Prozess-Instanzen dieselbe Codebasis, die durch entsprechende Parametrisierung jeweils die gewünschte Funktionalität "freischaltet". Auch die in Abb. 9 dargestellte, kontextabhängige Auswahl an erlaubten Operationen ist nicht etwa im Editor "hart verdrahtet", sondern dieser erhält diese Information vom *ChangeOperations-Service*, der auch die Änderung am Prozessgraphen durchführt. Derselbe *ChangeOperations-Service* wird auch vom ADEPT2-Laufzeitsystem im Kontext von Ad-hoc-Änderungen auf Prozess-Instanzebene gerufen und steht auch auf API-Ebene zur Verfügung, etwa für die Entwicklung anwendungsspezifischer Workflow-Clients. Weitere Details zur ADEPT2-Architektur und -Implementierung finden sich in [Rind06, RRJK06, RJR07, Reic08].

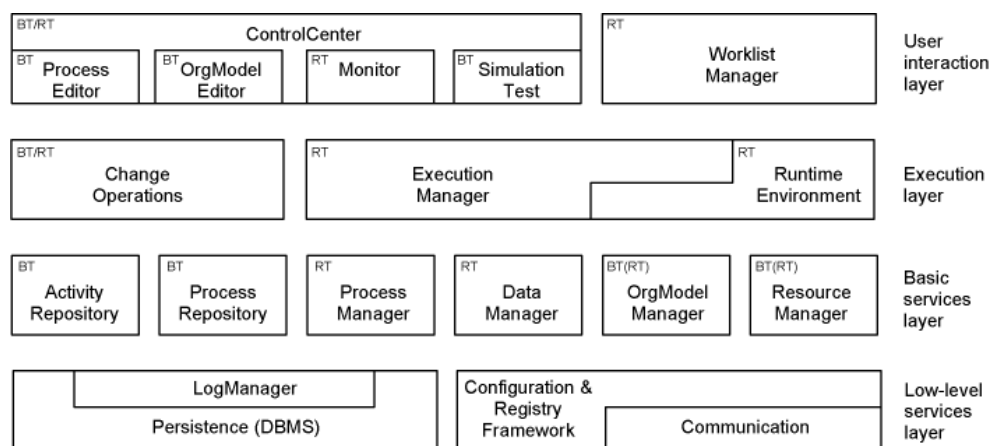


Abb. 10: Grobarchitektur von ADEPT2 (BT: Buildtime, RT: Runtime)

Um die Weiterentwicklung von ADEPT2 über das Ende des AristaFlow-Projekts hinaus zu gewährleisten und einen industriellen Einsatz für eine breite Nutzung der Software zu ermöglichen, wurde Anfang 2008 mit den federführenden Architekten und Entwicklern von ADEPT2 eine Ausgründung aus dem DBIS-Institut vorgenommen und gemeinsam mit Industriepartnern die AristaFlow GmbH in Ulm gegründet. Diese wird die ADEPT2-Codebasis zur Produktreife führen und deren Pflege und Weiterentwicklung als industriell einsetzbares Produkt übernehmen. Das Produkt trägt den Namen *AristaFlow® BPM Suite* [Juri08]. In seine Entwicklung sind neben den vorangehend beschriebenen Forschungsergebnissen auch die langjährigen Erfahrungen unserer Industriepartner aus ihren zahlreichen Workflow- und Software-Projekten eingeflossen.

<sup>4</sup> Das AristaFlow-Projekt wurde von 2004 bis 2007 vom Land Baden-Württemberg im Rahmen des PRIMIUM-Forschungsverbundes (siehe [www.primium.org](http://www.primium.org)) gefördert.

## 5 Stand der Entwicklung

Nachdem sich Vorversionen der AristaFlow<sup>®</sup> BPM Suite bereits in diversen internen und externen Einsatztests (z. B. [RüWa07, RüWa08]) bestens bewährt hatten, wurde das Produkt von der AristaFlow GmbH im Januar 2009 offiziell freigegeben und kann ab sofort bezogen werden (für nähere Informationen hierzu siehe Abschnitt 6). Die AristaFlow<sup>®</sup> BPM Suite besteht derzeit aus den folgenden Software-Komponenten: *Process Template Editor*, *Activity Repository Editor*, *OrgModel Editor*, *TestClient*, *Client*, *AutomaticClient*, *Monitor* und *Server*. Hinzu kommt eine offene API.

### Komponenten der AristaFlow<sup>®</sup> BPM Suite

Der **AristaFlow Process Template Editor** basiert auf den im vorherigen Abschnitt beschriebenen Prinzipien des "Correctness by Construction". In Ergänzung dazu werden, nebenläufig zum Modellierungsvorgang, weitere Fehleranalysen durchgeführt, etwa um Inkonsistenzen im aktuell modellierten Datenfluss anzuzeigen. In dem in Abb. 11 dargestellten Beispiel etwa wird von Aktivität B lesend auf das Datenelement X zugegriffen, ohne dass dieses im Prozessablauf zuvor mit einem Wert versehen wurde. Wird nun z.B. eine Schreibkante von Aktivität A (oder vom Startknoten) zur Aktivität X gezogen, verschwindet die angezeigte Fehlermeldung wieder. Es können nur solche Prozessvorlagen an den AristaFlow-Server (zum Einstellen in dessen Prozess-Repository) übertragen werden, die keine Fehler mehr aufweisen und die auch ansonsten vollständig sind, d.h. allen Prozessknoten (mit Ausnahme sog. "Nullknoten") sind ausführbare Komponenten und im Falle interaktiver Prozess-Schritte auch Bearbeiterzuordnungsauadrücke zugewiesen worden.

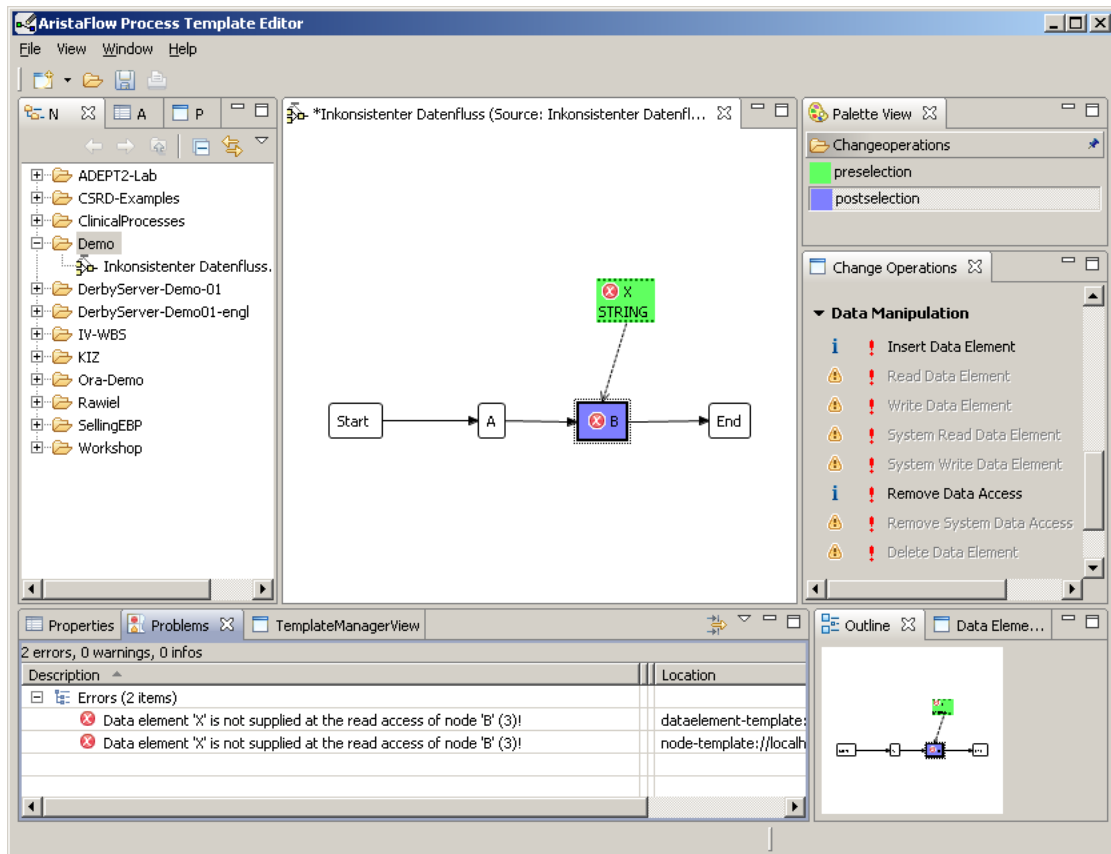


Abb. 11: AristaFlow Process Template Editor mit Anzeige eines inkonsistenten Datenflusses

Der **AristaFlow Process Template Editor** unterstützt das erwähnte "Plug & Play"-Prinzip in vollem Umfang. Alle im *Activity Repository* registrierten Aktivitätenvorlagen sind im *Process Template Editor* wählbar und können mittels Drag & Drop auf eine Kante im Prozessgraphen (dann wird dort ein neuer Knoten eingefügt) oder einen bestehenden Knoten gezogen werden. In Abb. 12 sieht man im linken Teilfenster (*Activity Repository Browser*) Beispiele für speziell konfigurierte Aktivitäten, die sich auf die Relation "Kunde" beziehen, sowie allgemeine Vorlagen, wie die unter "Generics" aufgeführten. Neben den im *Activity Repository* hinterlegten Aktivitätenvorlagen können auch ganze Prozesse

in Aktivitäten eingesetzt werden, was dann zu einem hierarchischen Prozessgeflecht führt. Diese Sub-Prozesse werden ebenfalls per Drag & Drop in den Prozessgraphen eingefügt. Dabei kann man zwischen verschiedenen Formen der Einbettung dieses Prozesses in den Hauptprozess sowie verschiedene Ausführungsmodi (Subprozess oder eigenständiger Prozess) wählen.

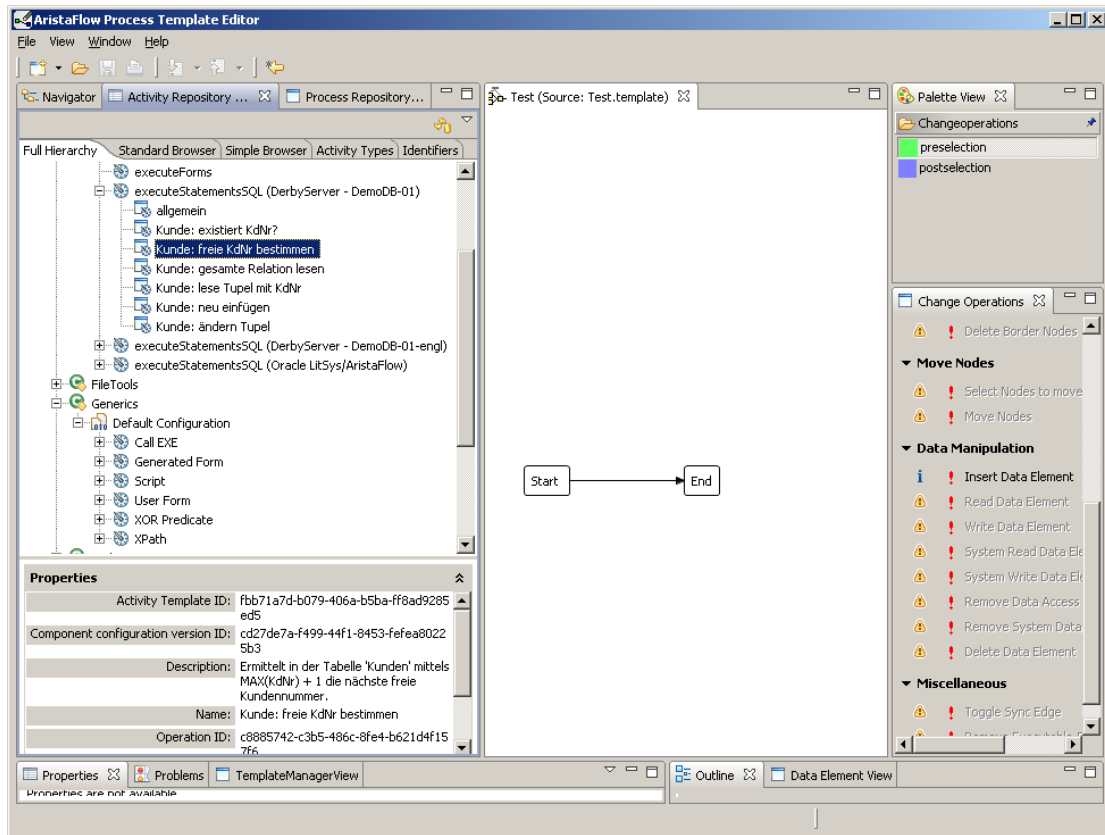
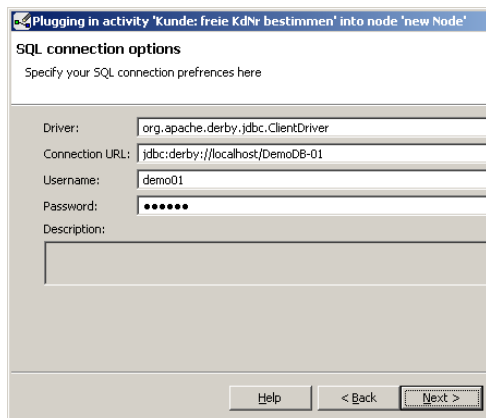
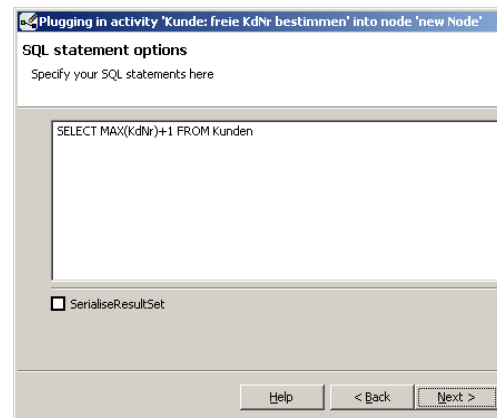


Abb. 12: Plug & Play Unterstützung im AristaFlow Process Template Editor

Der **AristaFlow Activity Repository Editor** dient dazu, Aktivitäten jeder Art dem AristaFlow-System "bekannt" zu machen. Zu diesem Zweck füllt man typspezifische Formulare aus. Für die in Abb. 12 dargestellte Aktivität "Kunde: freie KdNr bestimmen" würde man z.B. unter anderem Angaben für den Verbindungsaufbau zur Datenbank (Abb. 13a) sowie das zu verwendende SQL-Statement (Abb. 13b) hinterlegen. Bei einer allgemeineren Aktivitätenvorlage für einen Datenbankzugriff, würde man diese Angaben nur teilweise im *Activity Repository* spezifizieren oder auch ganz weglassen, und sie dann erst während des "Plug & Play"-Dialogs machen.



a)



b)

Abb. 13: Konfigurationsangaben für Datenbankaktivität

Der **AristaFlow OrgModel Editor** dient zur Erstellung und Pflege des Organisationsmodells. Das Organisationsmodell von AristaFlow ist sehr ausdrucksstark und erlaubt die Beschreibung auch sehr komplexer Unternehmensstrukturen. Entsprechend mächtig sind auch die Bearbeiterzuordnungsdrücke, die man – unter Bezugnahme auf das hinterlegte Organisationsmodell – den Aktivitäten bei Bedarf zuordnen kann. Abb. 14 zeigt in vergrößerter Form an, welche Elementtypen das Organisationsmodell aufweist und welche miteinander in Beziehung gesetzt werden können.

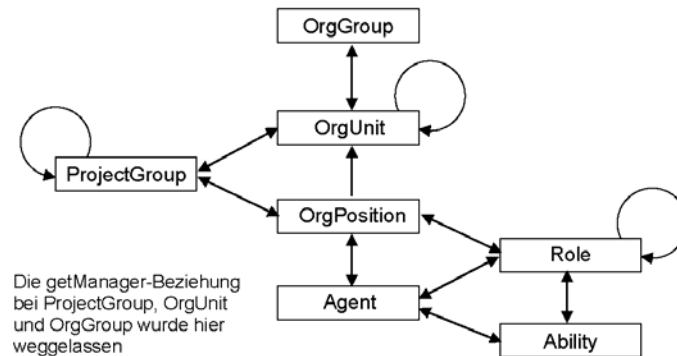


Abb. 14: Grobübersicht – Elemente und Beziehungen des AristaFlow-Organisationsmodells

Der **AristaFlow Testclient** dient dazu, strukturell korrekte, aber hinsichtlich Zuordnung von ausführbaren Diensten und Bearbeiterzuordnungsdrücken noch nicht vollständige Prozessvorlagen, testweise auszuführen. AristaFlow ergänzt in diesem Fall evtl. fehlende "Executables" (d.h. Aktivitätenprogramme) durch automatisch erzeugte Formulare, um Eingabeparameter anzuzeigen sowie Ausgabeparameter (die normalerweise von der Aktivität gesetzt werden) von Hand eingeben zu können.

Der **AristaFlow Client** ist ein "Standard Client", der die anstehenden Aufgaben seines Benutzers in einer Arbeitsliste anzeigt. Außerdem bietet er ihm die auf dem Server hinterlegten Prozessvorlagen zum Starten neuer Prozessinstanzen an (sofern der Benutzer für diese die "Startberechtigung" besitzt). Der Client verfügt zudem über Funktionen wie "Wiedervorlage", "Delegation" und "Nachfrage". Darüber hinaus kann man sich für die in der Arbeitsliste befindlichen Tätigkeiten (d.h. Aktivitäten) die zugehörige Prozessinstanz graphisch anzeigen lassen.

Der **Automatic Client** dient zur Ausführung automatischer Schritte, die keine Benutzerinteraktion erfordern. Aus Sicht des AristaFlow-Servers verhält sich der Automatic Client im Wesentlichen wie ein normaler (d.h. interaktiver) Client. Auch ihm werden die für ihn anstehende Aufträge in seine "Arbeitsliste" gestellt. Im Unterschied zum AristaFlow Client wird das zugeordnete Aktivitätenprogramm dann aber automatisch (d.h. ohne Benutzerinteraktionen) gestartet und ausgeführt. Ein Prozessschritt wird zu einem "automatischen Schritt", in dem "automaticclient" als Bearbeiterzuordnungsdruck gewählt wird.

Der **AristaFlow Monitor** ist für den Systemspezialisten gedacht und ermöglicht unter anderem, die laufenden Prozessinstanzen zu überwachen. Beispielsweise können sich autorisierte Benutzer den aktuellen Zustand oder die Ausführungshistorie einer Prozessinstanz anschauen sowie einen Blick in die Logdatei des zugeordneten Ausführungsdienstes werfen. Darüber hinaus kann der Monitor auch eine aktuell fehlgeschlagene Aktivität zurücksetzen sowie Ad-hoc-Änderungen an einer Prozessinstanz vornehmen. Dabei steht ihm das volle Spektrum an Änderungsoperationen wie bei der Prozessmodellierung zur Verfügung. So wurde die in Abb. 15 als fehlgeschlagen markierte Aktivität "Get Amazon Offer" zunächst zurückgesetzt und dann zusammen mit der nachfolgenden Aktivität "Get Amazon Price" im Prozessgraph nach hinten verschoben und, wie in Abb. 16 gezeigt, nach der Aktivität "CheckSpecialOffers" platziert.<sup>5</sup> Natürlich wurden auch diese Änderungen systemseitig einer Korrektheitsprüfung unterzogen.

Der **AristaFlow Server** ist die Kernkomponente der AristaFlow BPM Suite. Er verwaltet die laufenden Prozessinstanzen, die instanziierten Prozessvorlagen, das *Activity Repository* und das Organisationsmodell. Er setzt intern auf einer relationalen Datenbank auf, wobei das konkret verwendete

<sup>5</sup> Der konkrete Ablauf dieser Verschiebeoperation ist wie folgt: (1) Markierung des zu verschiebenden Blocks durch Markierung der entsprechenden Knoten. (2) Auswahl (der jetzt freigeschalteten) Operation 'Select Nodes to Move'. (3) Markierung der Einfügestelle: 'CheckSpecialOffers' (grün) und 'Choose offer' (blau). (4) Auswahl der jetzt freigeschalteten Operation 'Move Nodes'. – Fertig!

System austauschbar ist. Es muss im Wesentlichen lediglich über eine JDBC-Schnittstelle verfügen. Der Server basiert auf einer erweiterbaren Architektur, in deren Entwurf von Beginn an Aspekte wie Skalierbarkeit, Performanz und Wartbarkeit einbezogen wurden. Insbesondere sind auch die vorangehend diskutierten Systemfunktionen nahtlos in die Architektur integriert. Eine ausführliche Beschreibung der Architektur sowie ihrer Komponenten und Entwurfsprinzipien findet sich in [Reic08].

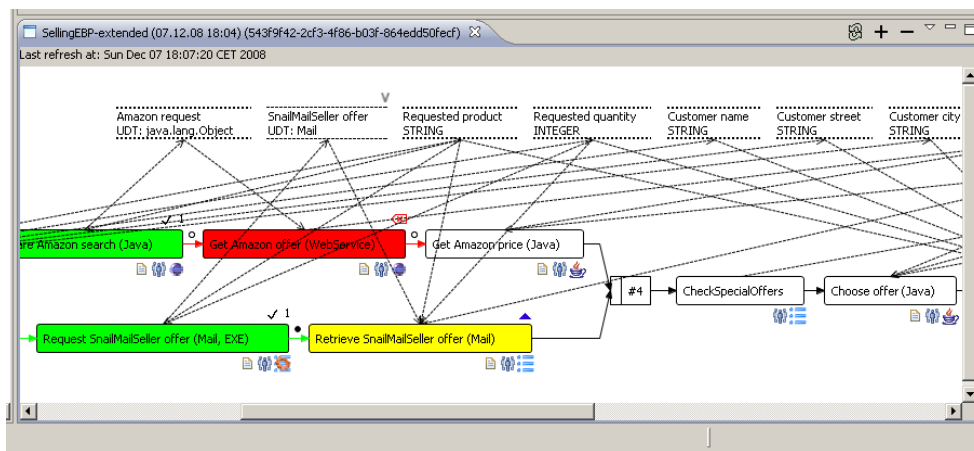


Abb. 15: AristaFlow Process Monitor (1)

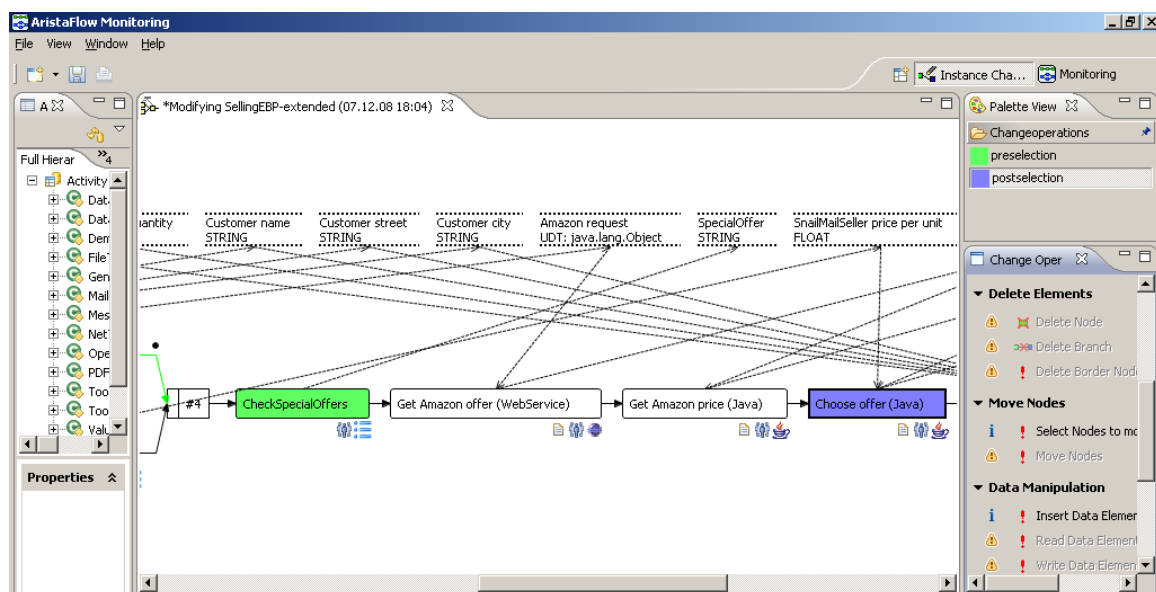


Abb. 16: AristaFlow Process Monitor (2)

Abschließend sei angemerkt, dass die AristaFlow BPM Suite eine **offene API** für alle Systemfunktionen bietet. Einerseits kann man das System eigenständig und ohne zusätzliche Software als Entwicklungs- und Ausführungsplattform für POIS nutzen, unter Verwendung der oben angegebenen Entwicklungs- und Laufzeitkomponenten. Andererseits erlaubt es das offene API aber auch, Teile des Systems als Workflow-Service in bestehende Software (z.B. Branchensoftware) einzubauen. Da das API sowohl Entwicklungs- als auch Laufzeitfunktionen abdeckt, können sich Softwarehäuser bei Bedarf „maßgeschneiderte“ Client-Anwendungen (inkl. Editoren) entwickeln. Intern bauen alle Lösungen auf denselben, umfassend dokumentierten Schnittstellen auf.

### Umsetzung der ADEPT-Forschungsvision

Die AristaFlow BPM Suite setzt die in Abschnitt 3 beschriebene ADEPT-Forschungsvision vollständig um und geht sogar noch darüber hinaus. Die vorgenommenen Erweiterungen betreffen z.B. die folgenden Komponenten bzw. Aspekte: Bereitstellung generischer Aktivitätensvorlagen verschiedener Art, Definition von Verzweigungsprädikaten mit mächtigem XOR-Prädikat-Composer (der per

Konstruktion garantiert, dass die gewählten Prädikate sowohl überlappungsfrei sind als auch der gesamte Wertebereich abgedeckt wird), Definition von Mitarbeiterzuordnungsausdrücken auf Grundlage eines sehr mächtigen Organisationsmodells, mehrstufige Eskalationsmechanismen (für den Fall, dass die in Arbeitslisten stehende Aktivitäten nicht rechtzeitig bearbeitet werden) und vieles Andere mehr.

Erste Erfahrungen mit Teilnehmern an Schulungsveranstaltungen haben gezeigt, dass das Ziel, rasch neue, robust ausführbare Prozesse erstellen zu können, voll erreicht wurde. Die Teilnehmer waren in der Regel bereits nach einer halbstündigen Einführung in der Lage, selbstständig erste Prozessmodelle zu erstellen und diese zur Ausführung zu bringen. Ähnliches gilt in Bezug auf die von der AristaFlow BPM Suite gebotenen Möglichkeiten für Prozessinstanzänderungen.

## 6 Nutzungsmodalitäten der AristaFlow<sup>®</sup> BPM Suite

Für die *Lehre* sowie für *institutsinterne Forschungsprojekte an Hochschulen* wird die AristaFlow-Software über das vom Institut für Datenbanken und Informationssysteme (DBIS) der Universität Ulm eingerichtete und betreute Portal *AristaFlow-Forum* ([www.AristaFlow-Forum.de](http://www.AristaFlow-Forum.de)) zur Verfügung gestellt. Dieses AristaFlow-Forum ist als Selbsthilfeeinrichtung der AristaFlow-Anwender gedacht und soll der wechselseitigen Hilfe (unter Beteiligung des Instituts DBIS) sowie dem Austausch von Prozessbeispielen, implementierten Aktivitäten und ähnlich nützlichen Artefakten dienen. Details zu den Nutzungsmodalitäten der AristaFlow-Software für Forschung und Lehre finden sich auf diesem Portal bzw. können von uns direkt erfragt werden (siehe Kontaktdaten am Ende des Artikels).

Bei weiterführenden Implementierungsaufgaben, welche die Bereitstellung entsprechender Entwicklungsumgebungen sowie eine entsprechende Schulung durch die AristaFlow GmbH erfordern, muss mit der AristaFlow GmbH eine spezielle Vereinbarung getroffen werden (auch hierzu finden sich Kontaktdaten am Ende des Artikels). Dasselbe gilt auch, wenn die AristaFlow BPM Suite in einem extern geförderten Forschungs- und Entwicklungsprojekt einer Hochschule (z.B. BMBF- oder EU-Projekten) eingesetzt werden soll oder wenn ein direkter Produktsupport durch die AristaFlow GmbH gewünscht wird. Für einen *kommerziellen Einsatz* der AristaFlow-Software – auch für einen testweisen Einsatz – ist ausschließlich die AristaFlow GmbH zuständig.

## 7 Zusammenfassung und Ausblick

Obwohl mittlerweile eigentlich niemand mehr bestreitet, dass Prozess-Management-Systeme sowie die auf ihrer Basis entwickelten POIS "flexibel" sein müssen, um wirklich auf breiter Front einsetzbar zu sein, ist es erstaunlich und zugleich erschreckend, wie wenig die derzeit verfügbaren Produkte und Forschungsprototypen diesbezüglich zu leisten vermögen [WSR09, WRR08]. Einer der Gründe hierfür mag sein, dass niemand den Schritt gewagt hat, einmal "gegen den Strom" zu schwimmen und dass man seitens der Industrie zu viele Schnellschüsse in Form von halbgaren "Standards" produziert, anstelle nach Lösungen der wirklich fundamentalen Probleme zu suchen.

Vor etwas mehr als 10 Jahren entstand am Institut für Datenbanken und Informationssysteme der Universität Ulm die Vision eines Prozess-Management-Systems, das einerseits funktional mächtiger sein sollte als die bis dato verfügbaren Technologien, andererseits aber einfacher und sicherer benutzbar sein sollte. Diese Idee war irgendwie verrückt und faszinierend zugleich. Die resultierenden Herausforderungen in Bezug auf das Verstehen der zu beachtenden Aspekte und deren wechselseitige Abhängigkeiten waren enorm. Und es war keine Vision, die im luftleeren Raum entstanden ist, sondern durch ganz reale Anwendungen (zunächst) aus dem klinischen Umfeld motiviert war.

Es dauerte auch ca. drei Jahre, bis eine detaillierte Anforderungserhebung durchgeführt, ein mächtiges Prozess-Meta-Modell konzipiert und eine umfassende Theorie entwickelt waren [ReDa98], welche in Kombination miteinander eine geeignete Basis für unser Vorhaben darstellten. Ein erster Prototyp ("ADEPT1") konnte bereits eindrucksvoll demonstrieren, dass die Theorie auch praktisch funktioniert, d.h. dass Ad-hoc-Abweichungen, Robustheit und einfache Benutzbarkeit keine Widersprüche sein müssen [Reic00]. Mit den Forschungs- und Entwicklungsarbeiten zu ADEPT2 wurde dann eine technologische Stufe erreicht, die weit über das hinausgeht, was andere Systeme derzeit zu leisten vermögen. Es gibt unseres Wissens weltweit kein System, das einen dermaßen hohen Grad an Flexibilität (Ad-hoc-Abweichungen, Prozess-Schema-Evolution), mit systemseitigen Korrektheitszusicherungen ("Correctness by Construction") und einfacher Bedienbarkeit realisiert. Durch die in Kapitel 4 beschriebene service-orientierte Architektur kann man das System nicht nur "als Ganzes"

einsetzen, sondern auch Teile davon in anderen Systemen als (interne) Komponente "einbauen". Durch das im *Process Template Editor* realisierte "Correctness by Construction"-Prinzip in Verbindung mit dem Plug & Play von Aktivitätenvorlagen (sowie den zur Verfügung stehenden generischen Aktivitäten) gestaltet sich die Realisierung prozessorientierter Anwendungen sehr einfach und macht diese Technologie daher auch für Ausbildungszwecke im BPM-Bereich sehr interessant.

Wir freuen uns sehr darüber, dass es gelungen ist, die Weiterentwicklung dieses Prozess-Management-Systems durch die Gründung der AristaFlow GmbH auf eine stabile Basis zu stellen und hierdurch auch einen industriellen Einsatz dieser innovativen Technologie zu ermöglichen. Darüber hinaus sind wir fest davon überzeugt, dass durch die gebotenen Möglichkeiten zur Flexibilisierung von Prozessen die Realisierung von POIS für anspruchsvollere Anwendungsdomänen, etwa der Medizin oder dem Produktentwicklungsbereich, erst möglich wird.

Wir betrachten ADEPT2- bzw. die AristaFlow-Software aber nicht als Abschluss einer Entwicklung, sondern als den Einstieg in eine nächste Generation von Prozess-Management-Systemen, welche völlig neue Möglichkeiten und Einsatzgebiete für die Unterstützung von Prozessen eröffnen. Natürlich werfen anspruchsvolle Anwendungen immer wieder auch neue, wissenschaftlich interessante Fragestellungen auf. Aus diesem Grund befassen wir uns mit einer Reihe weiterer hochgradig relevanter Forschungsfragestellungen, die in diesem Beitrag keine Erwähnung gefunden haben, die aber in der ein oder anderen Form auch in die Weiterentwicklung der AristaFlow BPM Suite einfließen wird. Beispielhaft seien an dieser Stelle einige unserer aktuellen Arbeiten genannt: Sicherstellung von Compliance-Vorschriften [LRD08], Evolution von Organisationsmodellen und Zugriffsregelungen [RiRe05, RiRe07, RiRe08], Zugriffskontrolle in flexiblen Prozess-Management-Systemen [PDA08, Webe05], Visualisierung und Monitoring komplexer Prozesse [BBR06, BRB07], Integration von Daten und Prozessen [MRH07,MRH08, RiRe06], Konfiguration und Verwaltung von Prozessvarianten [HBR08], Assistenz von Benutzern bei der Durchführung von Prozessänderungen [Rind05, WRWR05] sowie Ableitung von Prozessverbesserungen aus Prozessänderungen [Guen08, LRW08] genannt. Nähere Informationen zu diesen und anderen aktuellen Forschungsarbeiten finden sich unter [www.uni-ulm.dbis](http://www.uni-ulm.dbis) → Forschung.

## Danksagung

Der Weg von ADEPT1 über ADEPT2 bis hin zur AristaFlow<sup>®</sup> BPM Suite wäre nicht möglich gewesen, ohne die Unterstützung und Anregungen, die wir durch Kolleginnen und Kollegen am Institut DBIS sowie in zahlreichen Forschungsprojekten erfahren haben. Insbesondere gebührt unser Dank den vielen hoch motivierten Studenten, die im Rahmen ihrer Diplom-, Master- und Bachelorarbeiten, in Praktika sowie als wissenschaftliche Hilfskräfte im Laufe der Jahre Stück für Stück dazu beigetragen haben offene Fragen zu klären sowie Konzepte zu implementieren und zu validieren. Besonders erwähnen möchten wir an dieser Stelle Dipl.-Inf. Markus Lauer, der zum ADEPT2-Team der ersten Stunde gehörte, und Ende 2008 leider überraschend und viel zu jung verstorben ist.

## Kontakt

Universität Ulm  
Institut für Datenbanken und Informationssysteme  
89069 Ulm  
{peter.dadam, manfred.reichert, stefanie.rinderle}@uni-ulm.de  
[www.uni-ulm.de/dbis](http://www.uni-ulm.de/dbis) bzw. [www.AristaFlow-Forum.de](http://www.AristaFlow-Forum.de)

AristaFlow GmbH  
Sedanstr. 18  
89077 Ulm  
[info@aristaflow.com](mailto:info@aristaflow.com)  
[www.AristaFlow.com](http://www.AristaFlow.com)

## Referenzen

- [AADR04] Acker, H.; Atkinson, C.; Dadam, P.; Rinderle, S.; Reichert, M.: Aspekte der komponentenorientierten Entwicklung adaptiver prozessorientierter Unternehmenssoftware. Proc. 1. Verbundtagung AKA 2004, Augsburg, Dezember 2004. LNI P-57, 2004, S. 7-24
- [ABFJ08] Atkinson, C.; Brenner, D.; Falcone, G.; Juhasz, M.: Specifying High-Assurance Services. IEEE Computer, 41(8):64-70, 2008



- [AtDa07] Atkinson, C.; Dadam, P.: AristaFlow: Komponentenbasierte Anwendungsentwicklung, Prozesskomposition mittels Plug & Play und adaptive Prozessausführung. In: Haasis, K.; Heinzl, A.; Klumpp, D. (Hrsg.): Aktuelle Trends in der Softwareforschung, Tagungsband zum doIT Forschungstag, Mannheim, Juli 2007, S. 186-197
- [Atki06] Atkinson, C.; Stoll, D.; Acker, H.; Dadam, P.; Lauer, M.; Reichert, M.: Separating Per-client and Pan-client Views in Service Specification. Proc. IW-SOSE '06, Int'l Workshop on Service Oriented Software Engineering, Shanghai, China, May 2006, S. 47-52
- [AtSt08] Atkinson, C.; Stoll, D.: An Environment for the Orthographic Modeling of Workflow Components. Proc. PRIMMIUM Subconference at the Multikonferenz Wirtschaftsinformatik (MKWI), 2008, Garching, February 2008, CEUR Workshop Proceedings, Vol. 328 (<http://ceur-ws.org/Vol-328>)
- [AtSt08a] Atkinson, C.; Stoll, D.: Orthographic Modeling Environment. Proc. FASE 2008, Proc. 11th Int' Conf., on Fundamental Approaches to Software Engineering, Budapest, Hungary, March/April 2008, LNCS 4961, S. 93-96
- [BaDa00] Bauer, T.; Dadam, P.: Efficient Distributed Workflow Management Based on Variable Server Assignments. Proc. Int'l Conf. on Advanced Information Systems Engineering, CAiSE 2000, Stockholm, Sept. 2000, LNCS 1789, S. 94-109
- [BaRe02] Bauer, T.; Reichert, M.: Dynamische Änderung von Serverzuordnungen in verteilten Workflow-Management-Systemen. Datenbank-Spektrum, Heft 4/2002, S. 59-67
- [Baue01] Bauer, T.: Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme. Dissertation, Universität Ulm, Fakultät für Informatik, Februar 2001 (auch; TENEA Verlag für Medien, Berlin, 2001)
- [BBKK02] Bassil, S.; Benyoucef, M.; Keller, R.K.; Kropf, P.: Addressing Dynamism in E-negotiations by Workflow Management Systems. Proc. Int'l Workshop on Database and Expert Systems Applications (3<sup>rd</sup> Int'l Workshop on Negotiations in e-Markets - Beyond Price Discovery), Aix-en-Provence, France, September 2002
- [BBR06] Bobrik, R.; Bauer, T.; Reichert, M.: Proviado - A Personalized and Configurable Visualizations of Business Processes. In: Proc. 7th Int'l Conf. on Electronic Commerce and Web Technologies (EC-WEB'06), 2006, Krakow, Poland. LNCS 4082, S. 61-71.
- [BKK04] Bassil, S.; Keller, R.K.; Kropf, P.: A Workflow-Oriented System Architecture for the Management of Container Transportation. Proc. Int'l Conf. on Business Process Management, BPM 2004, Potsdam, Germany, June 2004, LNCS 3080, S. 116-131
- [Blas96] Blaser, R.: Konfiguration verteilter Anwendungen aus vorgefertigten Programmbausteinen. Diplomarbeit, Universität Ulm, Fakultät für Informatik, Abt. DBIS, 1996
- [BRB07] Bobrik, R.; Reichert, M.; Bauer, T.: View-based Process Visualization. Proc. of the 5th Int'l Conf. on Business Process Management (BPM'07), Brisbane, Australia, 2007, LNCS 4714, S. 88-95.
- [BRD03] Bauer, T.; Reichert, M.; Dadam, P.: Intra-subnet Load Balancing in Distributed Workflow Management Systems. Int'l Journal of Cooperative Information Systems, 12(3):205-323, 2003.
- [Brei06] Breier, F.: Workflowbasierte Unterstützung von sich ändernden Entwicklungsprozessen - Anforderungen und Lösungsansätze. Diplomarbeit, Universität Ulm, 2006
- [Brus99] Brust, J.: Komponentenbasierte Entwicklung von flexiblen und robusten Workflow-Anwendungen. Diplomarbeit, Universität Ulm, Fakultät für Informatik, Abt. DBIS, 1999
- [Dada06] Dadam, P.; Acker, H.; Göser, K.; Jurisch, M.; Kreher, U.; Lauer, M.; Rinderle, S.; Reichert, M.: ADEPT2 - Ein adaptives Prozess-Management-System der nächsten Generation. Tagungsband Stuttgarter Softwaretechnik Forum 2006 - Science meets Business - Aktuelle Trends aus der Softwaretechnik-Forschung, November 2006.
- [Dada95] Dadam, P.; Kuhn, K.; Reichert, M.; Beuter, Th.; Nathe, M.: ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen. Proc. 25. GI-Jahrestagung (GISI 95), Zürich, Sept. 1995. Springer-Verlag, 1995, S. 677-686
- [Dada97] Dadam, P.: Business Information Systems: Trends and Technological Challenges. Proc. Int'l Conf. on Business Information Systems, BIS '97, Poznan, April 1997, S. 509-524.

- [DaRe00] Dadam, P.; Reichert, M.: Towards a New Dimension in Clinical Information Processing. Medical Infobahn for Europe, Proc. MIE2000, Hannover, 2000, IOS Press, S. 295-301
- [DaRe09] Dadam, P.; Reichert, M.: The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support. Computer Science - Research & Development, Special Issue on Flexible Process-aware Information Systems, 2009
- [DRK00] Dadam, P.; Reichert, M.; Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems?. Proc. Int'l Conf. on Business Information Systems, BIS 2000, 4th Int'l Conf., Poznan, Poland, 2000, 2000, S. 36-59
- [GoGa05] Golani, M.; Gal, A.: Flexible Business Process Management Using Forward Stepping and Alternative Paths. Proc. Int'l Conf. on Business Process Management, BPM 2005, Nancy, France, Sept. 2005, LNCS 3649, S. 48-63
- [Gola05] Golani, M.: Dynamic Modification Mechanism of Business Processes Modeling in Workflows. PhD Thesis, Technion, Israel Institute of Technology, Haifa, 2005
- [Grei05] Greiner, U.; Müller, R.; Rahm, E.; Ramsch, J.; Heller, B.; Löffler, M.: AdaptFlow: Protocol-based Medical Treatment Using Adaptive Workflows. Methods of Information in Medicine, 44:80-88, 2005.
- [Grei06] Greiner, U.: Quality-Oriented Execution and Optimization of Cooperative Processes: Model and Algorithms. Dissertation, Fakultät für Mathematik und Informatik, Universität Leipzig, 2006
- [Grim97] Grimm, M.: ADEPT-TIME: Temporale Aspekte in flexiblen Workflow-Management-Systemen. Diplomarbeit, Universität Ulm, Abt. DBIS, 1997
- [Guen08] Guenther, C.W.; Rinderle-Ma, S.; Reichert, M.; van der Aalst, W.M.P.; Recker, J.: Using Process Mining to Learn from Process Changes in Evolutionary Systems. Int'l Journal of Business Process Integration and Management, 3(1):61-78, 2008
- [HBR08] Hallerbach, A.; Bauer, T.; Reichert, M.: Managing Process Variants in the Process Lifecycle. Proc. of the 10<sup>th</sup> Int'l Conf. on Enterprise Information Systems (ICEIS'08), Barcelona, Spain, 2008, S. 154-161.
- [Hein00] Heinlein, C.: Workflow- und Prozeßsynchronisation mit Interaktionsausdrücken und -graphen. Dissertation, Universität Ulm, Fakultät für Informatik, Mai 2000
- [Hein01] Heinlein, C.: Workflow and Process Synchronization with Interaction Expressions and Graphs. Proc. ICDE 2001, Heidelberg, Germany, April 2001, S. 243-252
- [Juri08] M. Jurisch: AristaFlow Next-Generation BPM: Robustheit, Flexibilität, Integration – Wissenschaftliche Konzepte in der Praxis. In: Stuttgarter Software-Technik-Forum, 2008.
- [KoVa07] Koehler, J.; Vanhatalo, J.: Process Anti-patterns: How to Avoid the Common Traps of Business Process Modeling, Part 1: Modeling Control Flow. IBM WebSphere Developer Technical Journal, February 2007.
- [KoVa07a] Koehler, J.; Vanhatalo, J.: Process Anti-patterns: How to Avoid the Common Traps of Business Process Modeling, Part 2: Modeling Data Flow. IBM WebSphere Developer Technical Journal, April 2007.
- [KRD93] Kuhn, K.; Reichert, M.; Dadam, P.: Ein offenes klinisches Datenbank- und Informationssystem zur Integration autonomer Teilsysteme. Prod. 38. Jahrestagung der GMDS, Lübeck, 1993, S. 54-57, MMV Medizin Verlag, München
- [Kuhn94] Kuhn, K.; Reichert, M.; Nathe, M.; Beuter, T.; Dadam, P.: An Infrastructure for Cooperation and Communication in an Advanced Clinical Information System. Proc. 18th Ann. Sym. on Computer Applications in Medical Care 1994, SCAMC '94, Washington, 1994, S. 519-523
- [Kuhn94a] Kuhn, K.; Reichert, M.; Nathe, M.; Beuter, T.; Heinlein, C.; Dadam, P.: A Conceptual Approach to an Open Hospital Information System. Proc. 12th Int'l Congress on Medical Informatics, MIE '94, Lisbon, May 1994, S. 374-378
- [LeRe07] Lenz, R.; Reichert, M.: IT Support for Healthcare Processes - Premises, Challenges, Perspectives. Data and Knowledge Engineering, 61(1):39-58, 2007
- [LRD08] Ly, L.T.; Rinderle-Ma, S.; Dadam, P.: Integration and Verification of Semantic Constraints in Adaptive Process Management Systems. Data and Knowledge Engineering, 64(1):3-23, 2008

- [LRW08] Li, C.; Reichert, M.; Wombacher, A.: Discovering Reference Process Models by Mining Process Variants. In: Proc. 6th Int'l Conference on Web Services (ICWS'08), September 2008, Beijing, China. IEEE Computer Society Press, S. 45-53
- [MoAn07] Mourão, H.; Antunes, P.: Supporting Effective Unexpected Exceptions Handling in Workflow Management Systems. Proc. ACM SAC'07, 2007, pp. 1242-1249.
- [MRB08] Mutschler, B.; Reichert, M.; Bumiller, J.: Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors and Implications. IEEE Transactions on Systems, Man, and Cybernetics, 38(3):280-291, 2008
- [MRH07] Müller, D.; Reichert, M.; Herbst, J.: Data-driven Modeling and Coordination of Large Process Structures. Proc. of the 15<sup>th</sup> Int'l Conf. on Cooperative Information Systems (CoopIS'07), Vilamoura, Algarve, Portugal, 2007, LNCS 4803, S. 131-149.
- [MRH08] Müller, D.; Reichert, M.; Herbst, J.: A New Paradigm for the Enactment and Dynamic Adaptation of Data-driven Process Structures. Proc. of the 20<sup>th</sup> Int'l Conf. on Advanced Information Systems Engineering (CAiSE'08), Montpellier, 2008, LNCS 5074, S. 48-63.
- [PDA08] Predeschly, M.; Dadam, P.; Acker, H.: Security Challenges in Adaptive E-Health Processes. Proc. SAFECOMP 2008, Newcastle, U.K., 2008, LNCS 5219, pp. 181-192
- [RDB03] Reichert, M.; Dadam, P.; Bauer, T.: Dealing with forward and backward jumps in workflow management systems. Int'l Journal Software and Systems Modeling (SOSYM), 2(1):37-58, 2003
- [RDMK00] Reichert, M.; Dadam, P.; Mangold, R.; Kreienberg, R.: Computerbasierte Unterstützung von Arbeitsabläufen im Krankenhaus - Konzepte, Technologien und deren Anwendung. Zentralblatt für Gynäkologie, 122(1):53-67, 2000
- [ReBa07] Reichert, M.; Bauer, T.: Supporting Ad-hoc Changes in Distributed Workflow Management Systems. In: Proceedings 15th Int'l Conf. on Coop. Information Systems (CoopIS'07), November 2007, Vilamoura, Portugal. LNCS 4803, S. 150-168
- [ReDa98] Reichert, M.; Dadam, P.: ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems, Kluwer Academic Publ., 10(2):93-129, 1998s
- [ReDa00] Reichert, Manfred and Dadam, Peter: Geschäftsprozessmodellierung und Workflow-Management - Konzepte, Systeme und deren Anwendung. Industrie Management, 16(3):23-27, 2000
- [Reic00] Reichert, M.: Dynamische Ablaufänderungen in Workflow-Management-Systemen. Dissertation, Universität Ulm, Fakultät für Informatik, Mai 2000
- [Reic05] Reichert, M.; Rinderle, S.; Kreher, U.; Dadam, P.: Adaptive Process Management with ADEPT2. In: Proc. Int'l Conf. on Data Engineering (ICDE'05), Tokyo, Japan. IEEE Computer Society Press, 2005, S. 1113-1114
- [Reic08] Reichert, M.; Dadam, P.; Jurisch, M.; Kreher, U.; Göser, K.: Architectural Design of Flexible Process Management Technology. Proc. PRIMM Subconference at the Multikonferenz Wirtschaftsinformatik (MKWI), 2008, Garching, February 2008, CEUR Workshop Proceedings, Vol. 328 (<http://ceur-ws.org/Vol-328>)
- [RiDa03] Rinderle, S.; Dadam, P.: Schemaevolution in Workflow-Management-Systemen ("Aktuelles Schlagwort"). Informatik-Spektrum, Band 26, Heft 1, Februar 2003, S. 17-19
- [Rind04] Rinderle, S.: Schema Evolution in Process Management Systems. Dissertation, Universität Ulm, Fakultät für Informatik, Dezember 2004
- [Rind05] Rinderle, S.; Weber, B.; Reichert, M.; Wild, W.: Integrating Process Learning and Process Evolution - A Semantics Based Approach. In: Proc. Int'l Conf. on Business Process Management (BPM'05), Nancy, France, 2005, LNCS 3649, S. 252-267
- [Rind06] Rinderle, S.; Kreher, U.; Lauer, M.; Dadam, P.; Reichert, M.: On Representing Instance Changes in Adaptive Process Management Systems. Proc. ProFlex 2006, First IEEE Workshop on Flexibility in Process-aware Information Systems (in conjunction with WETICE 2006), Manchester, June 2006, S. 297-302
- [RiRe05] Rinderle, S.; Reichert, M.: On the Controlled Evolution of Access Rules in Cooperative Information Systems. In: Proc. 13th Int'l Conf. on Cooperative Information Systems (CoopIS'05), Agia Napa, Cyprus, 2005, LNCS 3760, S. 238-255

- [RiRe06] Rinderle, S.; Reichert, M.: Data-driven Process Control and Exception Handling in Process Management Systems. Proc. of the 18<sup>th</sup> Int'l Conf. on Advanced Information Systems Engineering (CAiSE'06), Luxembourg, 2006, LNCS 4001, S. 273–287.
- [RiRe07] Rinderle-Ma, S.; Reichert, M.: A Formal Framework for Adaptive Access Control Models. Springer, Journal on Data Semantics IX ,Vol. LNCS 4601, 2007, S. 82-112
- [RiRe08] Rinderle-Ma, S.; Reichert, M.: Managing the Life Cycle of Access Rules in CEOSIS. Proc. of the 12th IEEE International Enterprise Computing Conference (EDOC'08), September 2008, Munich, Germany. IEEE Computer Society Press, S. 257-266.
- [RJR07] Rinderle, S.; Jurisch, M.; Reichert, M.: On Deriving Net Change Information From Change Logs - The DeltaLayer Algorithm. Proc. Datenbanksysteme in Business, Technologie und Web, BTW 2007, Aachen, September 2007, LNI 103, S. 364-381
- [RRD03] Reichert, M.; Rinderle, S.; Dadam, P.: ADEPT Workflow Management System - Flexible Support for Enterprise-Wide Business Processes. Proc. 1st Int'l Conf. on Business Process Management (BPM '03), Eindhoven, 2003, LNCS 2678, S. 371-379.
- [RRD04] Rinderle, S.; Reichert, M.; Dadam, P.: Flexible Support of Team Processes by Adaptive Workflow Systems. Distributed and Parallel Databases, 16(1):91-116, 2004
- [RRD04a] Rinderle, S.; Reichert, M.; Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems - A Survey. Data & Knowledge Engineering, 50(1):9-34, 2004
- [RRD04b] Rinderle, S.; Reichert, M.; Dadam, P.: Disjoint and Overlapping Process Changes: Challenges, Solutions, Applications. In: Proc. 11th Int'l Conf. on Cooperative Information Systems (CoopIS'04), Agia Napa, Cyprus, 2004, LNCS 3290, S. 101-121
- [RRJK06] Rinderle, S.; Reichert, M.; Jurisch, M.; Kreher, U.: On Representing, Purging, and Utilizing Change Logs in Process Management Systems. Proc. Int'l Conf. on Business Process Management, BPM 2006, Vienna, Austria, 2006, LNCS 4102, S. 241-256
- [RRW08] Rinderle-Ma, S.; Reichert, M.; Weber, B.: Relaxed Compliance Notions in Adaptive Process Management Systems. Proc. 27th Int'l Conference on Conceptual Modeling (ER'08), October 2008, Barcelona, Spain. Springer, LNCS 5231, S. 232-247
- [RüWa07] Rüppel, U.; Wagenknecht, A.: Improving Emergency Management by Formal Dynamic Process-modelling. Proc. 24th Conf. on Information Technology in Construction (W78), Maribor, Slovenia, Juli 2007, S. 559-564
- [RüWa08] Rüppel, U.; Wagenknecht, A.: Towards a Process-driven Emergency Management System for Municipalities. Proc. 12th Int'l Conf. on Comp. in Civil and Building Engineering & 2008 Int'l Conf. on Inf. Technology in Construction, Beijing, China, 2008
- [RWRW05] Rinderle, S.; Weber, B.; Reichert, M.; Wild, W.: Integrating Process Learning and Process Evolution - A Semantics Based Approach. Proc. Int'l Conf. on Business Process Management, BPM 2005, Nancy, France, 2005, LNCS 3649, S. 252-267
- [Webe05] Weber, B.; Reichert, M.; Wild, W.; Rinderle, S.: Balancing Flexibility and Security in Adaptive Process Management Systems. In: Proc. 13th Int'l Conf. on Cooperative Information Systems (CoopIS '05), Agia Napa, Cyprus, 2005, LNCS 3760, S. 59-76
- [WeRe08] Weber, B.; Reichert, M.: Refactoring Process Models in Large Process Repositories. In: Proc. 20th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'08), June 2008, Montpellier, France. Springer, LNCS 5074, S. 124-139
- [WRR08] Weber, B.; Reichert, M.; Rinderle-Ma, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering , 66(3):438-466, 2008
- [WRWR05] Weber, B.; Rinderle, S.; Wild, W.; Reichert, M.: CCB-Driven Business Process Evolution. Proc. ICCBR'05, 6th Int'l Conf. on Case-Based Reasoning), Chicago, IL, USA, August 2005, LNCS 3620, S. 610-624
- [WSR09] Weber, B.; Sadiq, S.; Reichert, M.: Beyond Rigidity - Dynamic Process Lifecycle Support - A Survey on Dynamic Changes in Process-aware Information Systems. Computer Science - Research & Development, 2009.