

Von funktionsorientierten zu prozessorientierten Informationssystemen – Herausforderungen und Lösungsansätze –

Peter Dadam, Manfred Reichert, Stefanie Rinderle
Universität Ulm, Fakultät für Informatik
Abt. Datenbanken und Informationssysteme
89069 Ulm

Schlüsselworte:

Prozessorientierung, prozessorientierte Informationssysteme, Adaptivität, Prozess-Management-Systeme, Workflow-Management, Anwendungsintegration

Zusammenfassung

Prozessorientierung und, damit einhergehend, die Implementierung prozessorientierter Informationssysteme, wird für die Unternehmens-IT in den nächsten Jahren ein "Mega-Thema" werden. Die Einführung von Prozesstechnologie darf jedoch nicht dazu führen, dass Unternehmen inflexibel werden und dass spätere Änderungen am (Geschäfts-)Prozess mit hohen Kosten oder hohen Fehlerrisiken verbunden sind. Dieser Beitrag beschreibt Anforderungen an Prozess-Management-Systeme aus Benutzersicht, erläutert die dafür notwendigen technologischen Grundlagen und kommentiert den aktuellen Stand sowie das Potenzial dieser Systeme.

1. Einleitung und Überblick

Die zunehmende Globalisierung der Märkte übt einen massiven Kosten- und Wettbewerbsdruck auf die Unternehmen in fast allen Branchen aus. In immer kürzeren Entwicklungszyklen müssen neue Produkte und Dienstleistungen zur Marktreife gebracht sowie neue Formen der Zusammenarbeit zwischen dem Unternehmen und seinen Kunden, Partnern und Zulieferern realisiert werden. Diese Entwicklungen haben naturgemäß einen massiven Einfluss auf die betrieblichen Abläufe (Prozesse), die deshalb ebenfalls in immer kürzerer Folge an die neuen Erfordernisse angepasst werden müssen. Die Fragen, die sich hierbei stellen, sind:

- Wie rasch geht die Implementierung eines neuen Prozesses? – Sprechen wir hier über Tage, Wochen, Monate oder gar Jahre?
- Zu welchen Kosten ist diese Prozessimplementierung realisierbar?
- Welches Fehlerrisiko ist damit verbunden? – D.h. wie groß ist die Gefahr, dass infolge der Einführung eines neuen Prozesses existierende Prozesse nicht mehr korrekt funktionieren, etwa wegen Überforderung der Mitarbeiter oder wegen Softwarefehlern?

- Wie teuer sind spätere Änderungen am Prozess?
- Wie entgeht man der "Wartungsfalle"? – D.h. wie vermeidet man, dass die eingesetzte Software durch diese ständigen Änderungen einen Komplexitätsgrad erreicht, der nicht mehr vernünftig handhabbar ist?

Diese Fragen bzw. Herausforderungen werden dazu führen, dass in den nächsten Jahren das Thema "Prozessorientierung" sowie damit verbundene IT-Lösungen – die *prozessorientierten Informationssysteme* – in den Unternehmen sehr stark in den Mittelpunkt des Interesses rücken werden. Diese Systeme müssen dann allerdings hinsichtlich Flexibilität und Adaptivität weit über das hinausgehen, was derzeitig Workflow-Management- und EAI-Systeme zu leisten im Stande sind, wenn sie den Erfordernissen wirklich gerecht werden wollen. Die Beschreibung der Anforderungen an prozessorientierte Informationssysteme bzw. die zugrunde liegende Basis-Technologie – im Folgenden als *Prozess-Management-Systeme* (PMS) bezeichnet – aus Anwendersicht sowie die Vorstellung technologischer Lösungsansätze bilden die Schwerpunkte dieses Beitrages.

Der Beitrag ist wie folgt aufgebaut: Im nächsten Kapitel erläutern wir, was wir unter Prozess-Management-Systemen und prozessorientierten Informationssystemen verstehen und wie diese beiden Dinge zusammenhängen. In Kapitel 3 gehen wir dann zunächst auf die Anforderungen an prozessorientierte Informationssysteme aus Anwendersicht ein, um dann in Kapitel 4 näher zu betrachten, welche technologischen Herausforderungen die Realisierung dieser Anforderungen für PMS nach sich zieht. Hierzu werden wir dann einige Lösungsansätze aufzeigen. Der Beitrag endet mit abschließenden Bemerkungen und Empfehlungen in Kapitel 5.

2. Was sind Prozess-Management-Systeme und prozessorientierte Informationssysteme?

Die Wirkungsweise von *Prozess-Management-Systemen* – wir vermeiden hier bewusst den so vielseitig deutbaren Begriff "Workflow-Management-System" – lässt sich am besten in Analogie zu Datenbank-Management-System (DBMS) erläutern. Wir orientieren uns dabei an den relationalen DBMS.

- Ein DBMS verwaltet Meta-Daten über Daten (Relationen-Schemata) – ein PMS verwaltet Meta-Daten über Prozesse (Prozess-Schemata).
- Ein DBMS manipuliert (Instanz-)Daten von Relationen-Schemata (Update, Insert, Delete von Tupeln) – ein PMS manipuliert Prozess-Instanzen (Änderung des Prozess-Status, Änderung von Prozess-Instanz-Variablen).
- Ein DBMS ändert Relationen-Schemata (ALTER TABLE ...) und propagiert diese Änderungen auf die (Daten-)Instanzen – ein PMS (Ziel!) ändert Prozess-Schemata und propagiert diese Änderungen auf laufende Prozess-Instanzen.

Der Zusammenhang zwischen PMS und prozessorientierten Informationssystemen ist nun analog wie bei DBMS und DBMS-gestützten Informationssystemen. DBMS und PMS sind jeweils die (anwendungs-)neutralen Basis-Systeme. Kommen anwendungsbezogene Daten und Funktionen hinzu, dann sprechen wir von einem Informationssystem.

Ein wichtiges Anliegen prozessorientierter Informationssysteme ist die klare Trennung von (Geschäfts-)Prozesslogik und Anwendungsfunktionen. D.h. der Prozess als solcher wird im zugrundeliegenden PMS hinterlegt und in gewisser Weise dort "ausgeführt". Aus Benutzersicht gestaltet sich diese "Ausführung" so, dass eine ausführbare Tätigkeit allen oder ausgewählten Benutzern, die über die entsprechende Berechtigung ("Rolle") verfügen, in ihre Arbeitsliste gestellt wird. Wählt der Benutzer aus dieser Arbeitsliste eine Tätigkeit aus, so startet das PMS die dieser Tätigkeit zugeordnete Anwendung. Beendet der Benutzer diesen Arbeitsschritt, "schaltet" das PMS weiter und stellt wiederum die nachfolgende(n) Tätigkeit(en) den jeweiligen Benutzern in deren Arbeitsliste. – Unter der Oberfläche kann der Ablauf hierbei erheblich komplexer aussehen. Im allgemeinen Fall sind nicht nur menschliche Benutzer involviert, sondern es werden PMS-seitig auch automatische Schritte ausgeführt und ggf. unterschiedliche Anwendungssysteme auf diese Weise integriert (siehe Abb. 1).

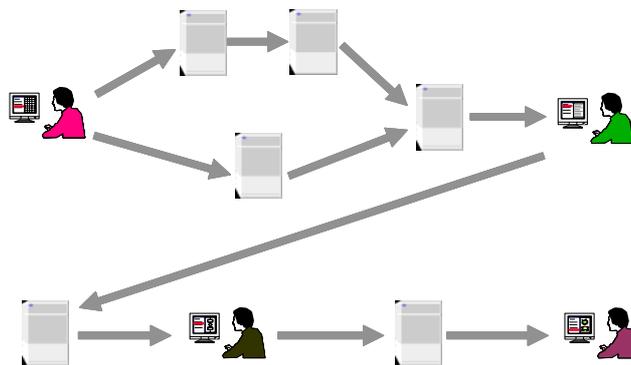


Abb. 1: Prozessorientiertes Informationssystem

Bei den auszuführenden Programmen kann es sich um einfache Formulare, um die Ausführung von einfachen Desktop-Programmen, aber auch um den Aufruf von komplexen Business-Functions in einem ERP-System handeln. Im allgemeinen Fall muss man davon ausgehen, dass das PMS in der Lage sein muss, beliebige Programme (z.B. auch Webservices) aufzurufen, diese mit Aufrufparametern bzw. Eingabedaten zu versorgen und eventuelle Rückgabewerte wieder entgegenzunehmen. Diese Rückgabewerte wiederum müssen bei Bedarf nachfolgenden Prozessschritten wieder als Aufrufparameter oder Eingabedaten zur Verfügung gestellt werden.

3. Was müssen Prozess-Management-Systeme leisten? – Die Anwendersicht

Eine ganz wesentliche Forderung ist sicherlich, dass neue prozessorientierte Anwendungen bzw. prozessorientierte Informationssysteme erheblich rascher als heute realisiert werden können. Das Ziel ist, prozessorientierte Informationssysteme im "Plug & Play"-Stil zu realisieren, wie in Abb. 2 angedeutet. Der Prozess-Modellierer wählt aus einer Bibliothek von Anwendungskomponenten die geeignete aus und ordnet sie einem Prozessschritt zu.

Es reicht allerdings nicht aus, nur das "Zusammen-Clicken" einer Anwendung in der skizzierten Art zu unterstützen (was einige Systeme ansatzweise heute schon leisten). Die daraus resultierenden prozessorientierten Anwendungen sollten anschließend auch robust und stabil

laufen. Böse Überraschungen zur Laufzeit, etwa der Art, dass Anwendungskomponenten nicht mit den erforderlichen Aufrufparametern versorgt werden können und es deshalb zu Laufzeitfehlern kommt, müssen bereits zur Modellierungszeit durch systemseitige Konsistenzüberprüfungen (Build-Time) ausgeschlossen werden. Ähnliches gilt für prozessinterne Blockierungszustände, unerwünschte Zyklen und anderes mehr.

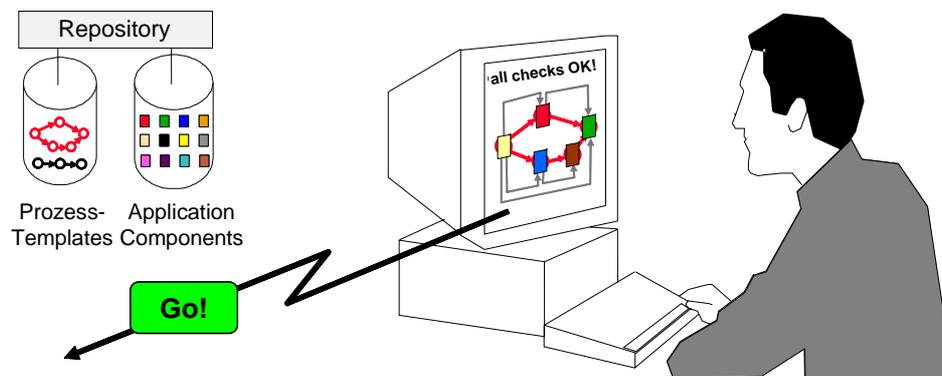


Abb. 2: Realisierung eines prozessorientierten Informationssystems mittels "Plug & Play"

Ist das Problem der raschen Entwicklung prozessorientierter Anwendungen gelöst, dann können, wenn die betrieblichen Prozesse erfasst, optimiert und im PMS hinterlegt wurden, im Prinzip alle bislang "manuellen" Prozesse nunmehr rechnergestützt ausgeführt werden. Was geschieht aber, wenn zur Ausführungszeit Fälle bzw. Probleme auftreten, die zur Modellierungszeit des Prozesses nicht bekannt waren oder nicht bedacht worden sind? Erklären wir dann z.B. unserem Kunden, nachdem bereits fünf Aufträge "schiefgelaufen" sind, dass noch zehn weitere seiner Aufträge leider ebenfalls fehlschlagen werden, weil wir jetzt PMS-Technologie einsetzen und an den bereits angestoßenen Prozessen leider nichts ändern können? Oder arbeiten wir am PMS vorbei, d.h. wir tun so, als ob wir den hinterlegten Prozess ausführen, machen in Wirklichkeit aber etwas anderes (falls wir das überhaupt noch können)?

Die Herausforderung ist, prozessorientierte Informationssysteme zu realisieren, die mit solchen Ausnahmesituationen umgehen können, d.h. bei Bedarf auch Ad-hoc-Abweichungen im Einzelfall vom vorgeplanten Ablauf zulassen. Ganz wichtig ist hierbei, dass der *Anwender* diese Änderungen durchführen kann, und zwar ohne hierfür über irgendwelche tiefgehenden Systemkenntnisse verfügen zu müssen. Hierzu muss eine semantisch hohe Sicht auf Prozesse und auf Änderungen angeboten werden.

In Abb. 3.a - 3.f wird in Anlehnung an das ADEPT-System [ReDa98] skizziert, wie eine solche Benutzerschnittstelle aussehen könnte: Der Benutzer bearbeitet einen Fall, der eine Ausnahmebehandlung erfordert (a). Er betätigt einen "Ausnahme-Button" (nicht dargestellt) und das System erfragt in einem Dialog die Art der gewünschten Ausnahmebehandlung (z. B. Aktivität einfügen, löschen oder verschieben). Der Benutzer entscheidet sich in diesem Beispiel für "Aktivität einfügen" und bekommt eine Liste der in diesem Kontext prinzipiell wählbaren Aktivitäten angeboten (b). Anschließend erfragt das System, ab wann im Prozessverlauf der neue Schritt den zuständigen Bearbeitern angeboten und bis wann im Prozessverlauf dessen Bearbeitung abgeschlossen werden soll (c). Das System prüft nun, ob die gewünschte

Änderung in diesem Kontext durchgeführt werden kann und meldet ein positives Ergebnis (d, e). Anschließend fährt der Benutzer mit der Bearbeitung des Falles fort (f). – Bitte beachten: Geändert wurde nur dieser eine Fall, d.h. diese eine Prozess-Instanz. Alle anderen Prozesse dieses Typs sind von dieser Änderung nicht betroffen.

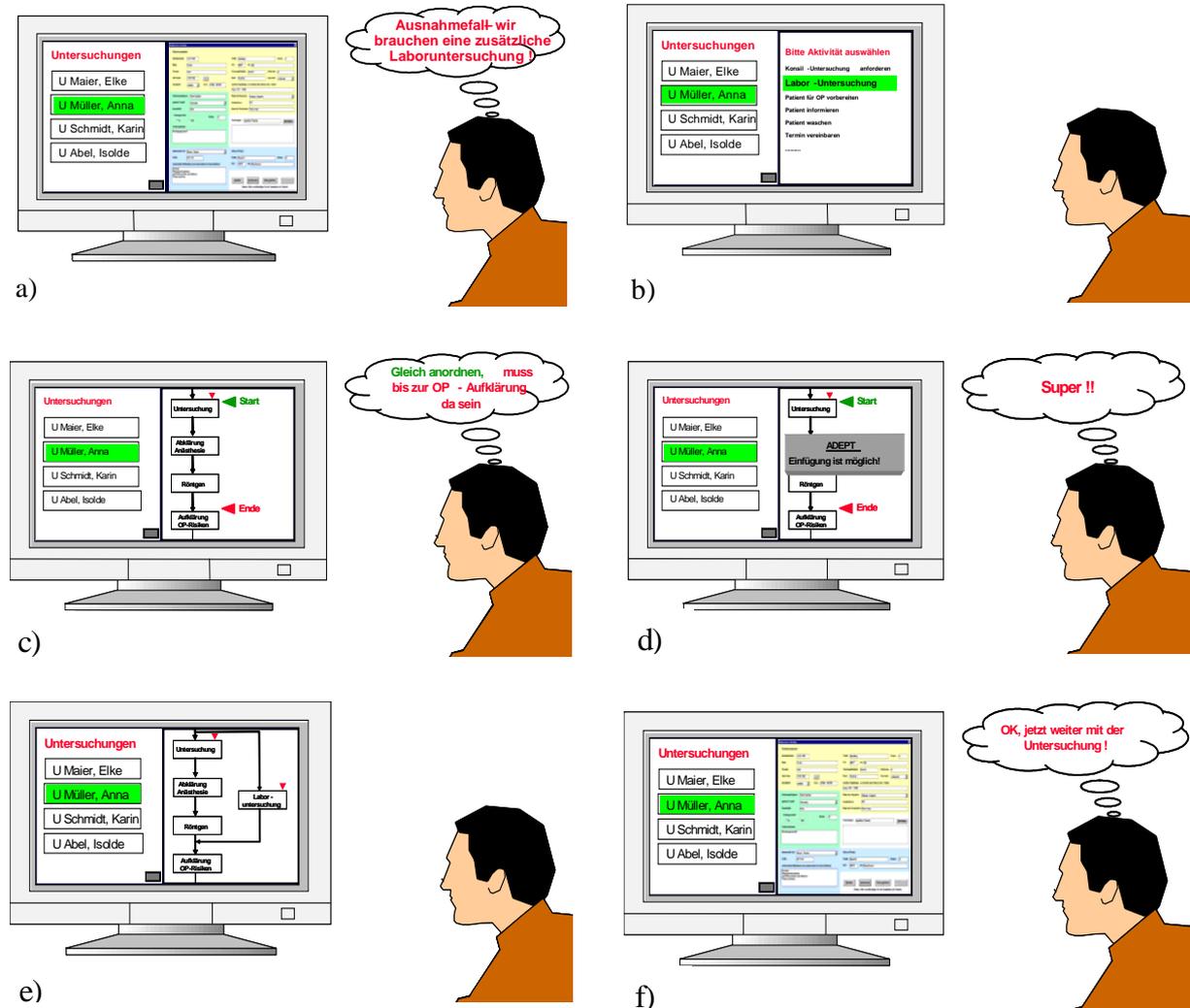


Abb. 3: Ad-hoc-Abweichung aus Benutzersicht

Möglichkeiten zu Ad-hoc-Änderungen der beschriebenen Art sind ein sehr mächtiges Mittel, um auf individuelle Sonderfälle reagieren zu können oder um übergangsweise Modellierungsfehler im Prozessablauf auf Einzelfallbasis zu korrigieren. Diese Fähigkeit wird derzeit von kommerziellen Systemen bislang entweder überhaupt nicht oder nur in stark eingeschränkter Form angeboten (oder sie wird angeboten, kann aber zu gravierenden Konsistenzproblemen bzw. Laufzeitfehlern führen). – Die Gründe hierfür werden ersichtlich, wenn wir auf die notwendigen Voraussetzungen für Fähigkeiten dieser Art im nächsten Kapitel zu sprechen kommen.

(Geschäfts-)Prozesse sind naturgemäß im Zeitablauf Änderungen unterworfen: Gesetzliche Rahmenbedingungen ändern sich, Interaktionsformen mit Kunden oder Zulieferern werden

geändert, es wird unternehmensintern umorganisiert usw. Man wird also irgendwann den Prozess(typ) ändern und neue Prozesse ab diesem Zeitpunkt nach dem neuen Schema ausführen wollen. Die Frage ist in diesem Fall, was mit bereits gestarteten Prozessinstanzen geschehen soll, die auf Basis des alten Prozessschemas erzeugt wurden. Bei kurz laufenden Prozessen wird es in vielen Fällen genügen, dass das PMS eine Koexistenz von Prozessinstanzen, basierend auf verschiedenen Versionen des Prozessschemas, unterstützt. D.h. die "alten" Prozessinstanzen werden einfach noch nach dem alten Schema zu Ende geführt. In vielen Fällen, insbesondere in Verbindung mit Prozessen, die über mehrere Monate oder Jahre laufen, reicht das jedoch nicht aus. Es müssen deshalb Mittel und Wege gefunden werden, die bereits laufenden Prozessinstanzen auf das neue Schema zu migrieren.

Im Prinzip könnte man dieses Problem zwar auch mit den oben skizzierten Möglichkeiten für Ad-hoc-Abweichungen lösen, dies wäre jedoch für die betroffenen Benutzer sehr aufwendig und insgesamt – insbesondere wenn viele Instanzen zu ändern sind – auch sehr fehlerträchtig. Die ideale Lösung wäre deshalb, die Migration der Prozessinstanzen auf das PMS zu übertragen. Der Prozessverantwortliche würde das Prozessschema ändern (genauer: eine neue Version des Prozessschemas erzeugen), er würde das PMS prüfen lassen, welche der laufenden Instanzen systemseitig migrierbar sind und würde deren Migration dann quasi per "Knopfdruck" erledigen. Für die nicht migrierbaren Instanzen kann systemseitig ein entsprechendes Analyseprotokoll erstellt werden, so dass anschließend entschieden werden kann, welche Instanzen ggf. inspiziert und evtl. manuell mittels Ad-hoc-Änderung migriert werden.

Die Realisierung prozessorientierter Informationssysteme mittels Plug & Play, die Unterstützung von Ad-hoc-Abweichungen, von Prozessschemaevolution und einer Reihe weiterer wünschenswerter Systemfähigkeiten wird jedoch nur dann Realität werden können, wenn die gesamte damit verbundene Komplexität nicht zu Lasten des Anwendungs- bzw. Komponentenentwicklers geht. Wie bei den relationalen DBMS, wo viele der komplizierten Dinge, wie z.B. Concurrency Control, Logging, Recovery, Caching, Anfrageoptimierung, komplett ins DBMS verlagert wurden, muss dies entsprechend auch bei den PMS geschehen.

4. Wie können Prozess-Management-Systeme dies leisten? – Die Systemsicht

Die große Herausforderung ist, dass viele verschiedene Aspekte, wie formales Prozessmodell, Ad-hoc-Flexibilität, Unterstützung temporaler Aspekte, Prozessschemaänderungen, Propagation von Schemaänderungen auf laufende Prozessinstanzen, Benutzerschnittstellen, transaktionale Eigenschaften, komponentenorientierte Softwareentwicklung, Systemarchitekturfragen, effiziente Speicherung und Verwaltung von Prozessinstanzen, Antwortzeitverhalten und anderes mehr nicht isoliert, sondern im Zusammenhang und Zusammenwirken betrachtet und verstanden werden müssen.

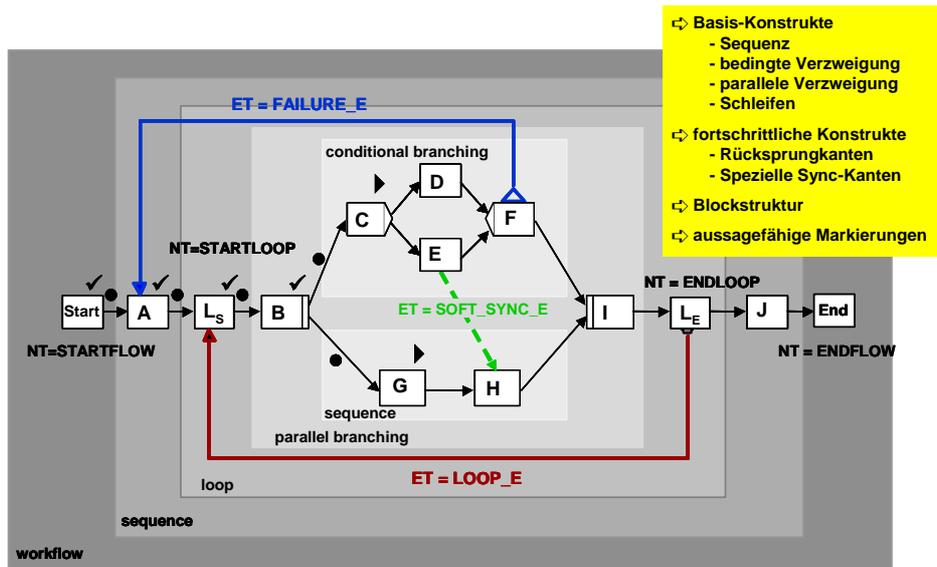


Abb. 4: Konstrukte des ADEPT-Prozess-Metamodells

So sollte z.B. das Prozess-Metamodell, also die Art und Weise, wie und mit welchen Konstrukten Prozesse modelliert werden (Knotentypen, Kantentypen, Arten von Markierungen, Art der Datenflussmodellierung usw.) einerseits ausdrucks mächtig genug sein, um alle relevanten Anwendungsfälle möglichst adäquat darstellen zu können, andererseits sollte es aber so strikt sein, dass es formalen Analysen (z.B. alle Datenflüsse korrekt, blockierungsfrei usw.) zugänglich ist. – Man kann auch hier wieder eine Parallele zu den relationalen DBMS ziehen. Auch dort hat man ein formales Datenmodell (die Relation) sowie Operationen auf diesem Modell (Relationenalgebra: Selektion, Projektion, Join, ...). Erst durch ein solches Modell mit seinen formalen Eigenschaften wurde die systemseitige Anfrageoptimierung möglich. Nur wenn eine solche Basis vorhanden ist, kann man eine Anfrage Q in eine andere Anfrage Q' transformieren und formal zeigen (mit Satz und Beweis), dass bzw. unter welchen Vorbedingungen diese Transformation korrekt ist, d.h. die Ausführung von Q zum selben Anfrageergebnis wie Q' führt.

Dies gilt, im übertragenen Sinne, auch für Ad-hoc-Änderungen und Prozessschemaevolution in PMS. Darüber hinaus wird man in diesem Kontext aber noch fordern müssen, dass die jeweiligen Analysen effizient durchführbar sind. Sehr hilfreich ist in diesem Zusammenhang, wenn man den Bereich, in dem die Prozessinstanz analysiert werden muss, systemseitig einzuschränken vermag. Aus diesem Grund ist z.B. das von uns entwickelte ADEPT-Metamodell den an sich sehr mächtigen Petri-Netzen in dieser Hinsicht überlegen. Durch Blockstrukturierung, verschiedene Kantentypen, explizite Schleifenkanten sowie durch Markierungen, die erkennen lassen, auf welchem Weg ein bestimmter Zustand erreicht worden ist, kann man im ADEPT-Modell (siehe Abb. 4) in vielen Fällen bereits durch Analyse eines recht begrenzten Bereichs Korrektheits- und Konsistenzaussagen treffen (siehe [ReDa98, SRD04] für eine ausführliche Beschreibung).

Noch dramatischer sind die Effekte und Unterschiede im Kontext *Prozessschemaevolution*. Das Problem ist an sich schon recht verzwickelt (siehe Abb. 5 und 6 zur Illustration des Prob-

lems sowie Abb. 7 für die diversen Fälle, die (mindestens) unterschieden werden müssen). Mit dem "richtigen" Prozess-Metamodell kann man diese Komplexität jedoch zu einem großen Teil vor dem Benutzer verbergen. Mit dem "falschen" Prozess-Metamodell kann Prozessschemaevolution jedoch leicht zum Alptraum werden. Unter Umständen muss der Prozessverantwortliche selbst Abbildungsregeln vom alten in das neue Prozessschema definieren, mit all den damit verbundenen Risiken (siehe [RRD04a] für eine Diskussion der verschiedenen Ansätze und entsprechende Beispiele).

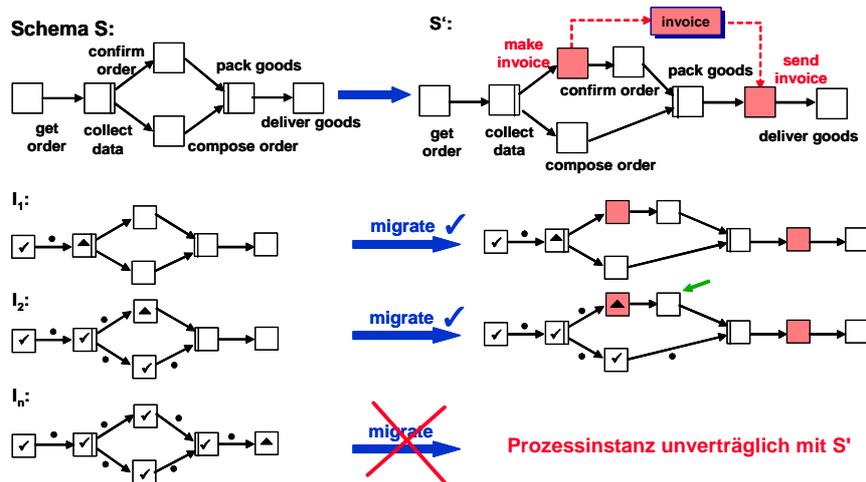


Abb. 5: Prozessschemaevolution mit Instanzen ohne Ad-hoc-Änderungen

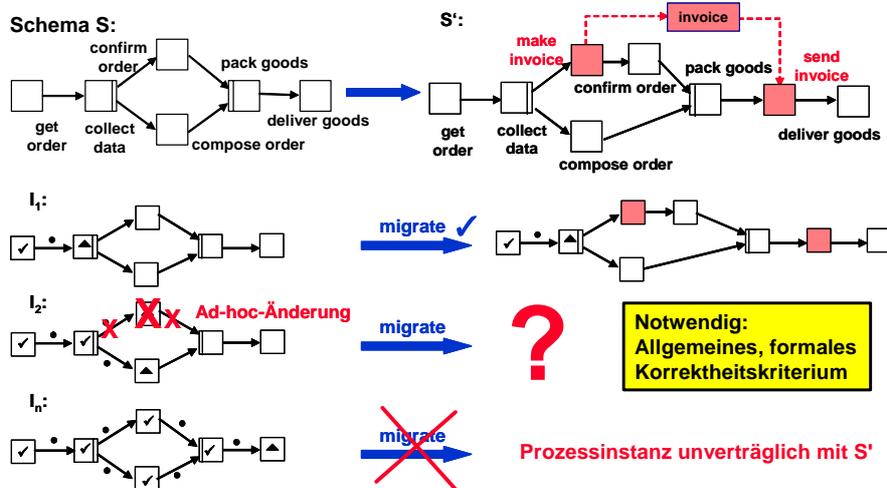


Abb. 6: Prozessschemaevolution bei Instanzen mit Ad-hoc-Änderungen

Neben den notwendigen formalen Grundlagen ist jedoch noch eine weitere Voraussetzung für Ad-hoc-Flexibilität und Prozessschemaevolution unabdingbar: *Einzeln manipulierbare Prozessinstanzen!* Dies dürfte für die meisten der am Markt befindlichen Systeme für Prozess-Management (Workflow-Management-Systeme, EAI-Systeme) derzeit nicht der Fall sein. Sie verwenden mittlerweile zwar alle ein graphisches Prozessmodell zur Modellierung, generieren aus diesem jedoch Programmcode, Datenbank-Trigger oder eine Kombination aus beiden. D.h. sie erzeugen aus dem Graphmodell letztlich einen sog. endlichen Automaten (zentral

oder verteilt realisiert), der Prozesse bzw. Prozessinstanzen "hart verdrahtet" ausführt.

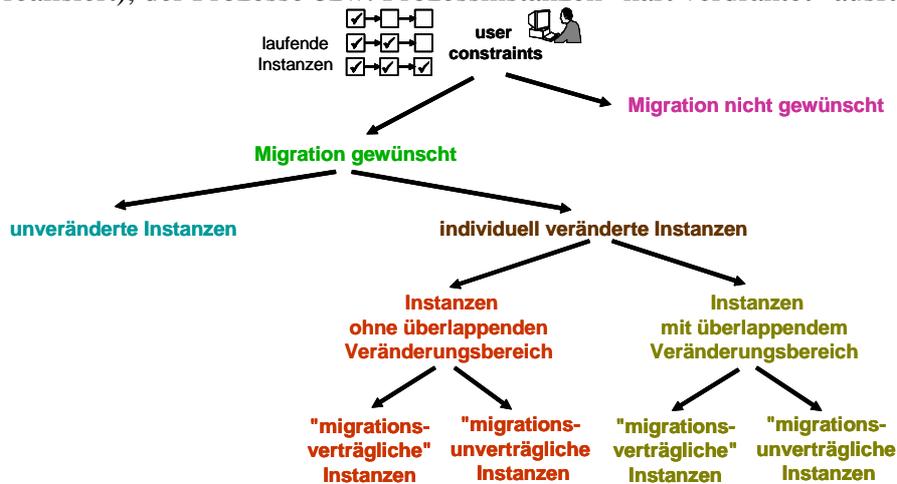


Abb. 7: Fallunterscheidungen bei Prozessschemaevolution

Auf einer solchen Technologiebasis sind echte Ad-hoc-Abweichungen gar nicht und Prozessschemaevolution nur in extrem eingeschränkter Weise möglich. – Diese PMS entsprechen in gewisser Weise (als Analogie gesehen) den DBMS der ersten Generation, wo aus dem Datenbankschema letztlich Records mit Verkettungsstrukturen erzeugt und Datenbankoperationen im Anwendungsprogramm direkt in Routinen übersetzt wurden, um bestimmte Zeigerketten zu durchlaufen. Wurde an der physischen Struktur der Datenbank etwas geändert, so mussten meist auch die Anwendungsprogramme neu übersetzt werden.

5. Abschließende Bemerkungen und Empfehlungen

Prozessorientierung wird für die Unternehmens-IT in den nächsten Jahren ein "Mega-Thema" werden. Das Problem wird von vielen IT-Verantwortlichen derzeit noch völlig unterschätzt. Sie sehen es immer noch vor allem als Problem der Anwender und überlassen die Entwicklung von Lösungen (fast) völlig den Fachabteilungen. Hierdurch entstehen oft sehr heterogene Lösungen, die in den nächsten Jahren zu immensen Integrationsproblemen führen werden. Die zentrale Forderung wird sein, Prozesse sehr viel rascher und kostengünstiger als heute realisieren zu können. Prozessmanagementsysteme werden deshalb (ob man sie mag oder nicht) verstärkt realisiert werden müssen. Aber: Die Flexibilität des Unternehmens darf durch den Übergang zu dieser Technologie nicht leiden. Mit heutiger Technologie besteht diesbezüglich ein erhebliches Risiko.

PMS der vorgestellten Art haben ein sehr großes technologisches Potenzial und können eine Reihe der oben beschriebenen Probleme auf sehr elegante und effiziente Weise lösen. Sie haben das Potenzial, die Entwicklung flexibler, prozessorientierter Informationssysteme ganz entscheidend zu vereinfachen und zu verbessern. Aber: Diese Funktionalität muss von den Anwendern bei den Herstellern auch eingefordert werden. Die Hersteller entwickeln seit einigen Jahren zunehmend sehr (kunden)anforderungsbezogen und mit immer kleineren zeitlichen Entwicklungshorizonten. Sie treiben den Aufwand zur Entwicklung echter technologi-

scher Innovationen nur dann, wenn die Kunden diese Innovationen explizit nachfragen. Die Anwender kennen in vielen Fällen jedoch überhaupt nicht die technologischen Alternativen und fordern deshalb häufig nur Verbesserungen entlang des eingeschlagenen Weges. Der Verzicht auf die Erarbeitung eigener vorausschauender Perspektiven auf Anwenderseite kann sich hier auf mittlere Sicht bitter rächen bzw. zu einem späten Zeitpunkt sehr teure Kurskorrekturen notwendig machen. – Deshalb die Empfehlung:

- Mit den Entwicklungen und Fragestellungen im Forschungsbereich auseinandersetzen.
- Eine eigene Meinung bilden, was Ihr Unternehmen mittel- und langfristig braucht.
- Bisherige Ansätze ggf. ruhig auch einmal grundsätzlich in Frage stellen.
- Sich auf den Kenntnisstand bringen, die Anforderungen in adäquater Weise zu artikulieren.

Literatur und weiterführende Information

- [ReDa98] Reichert, M.; Dadam, P.: ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. *Journal of Intelligent Information Systems*, Kluwer Academic Publ., Vol. 10, No. 2, March/April 1998, pp. 93-129
- [RRD04] Rinderle, S.; Reichert, M.; Dadam, P.: Flexible Support of Team Processes by Adaptive Workflow Systems. *Distributed and Parallel Databases*, Vol. 16, No. 1, 2004, pp. 91-116
- [RRD04a] Rinderle, S.; Reichert, M.; Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems - A Survey. *Data and Knowledge Engineering*, Vol. 50, No. 1, 2004, pp. 9-34

Auf unseren Webseiten finden sich in den Rubriken Forschung und Publikationen eine Vielzahl weiterer Veröffentlichungen zu diesem Thema zum Download.

Kontaktadresse:

Prof. Dr. Peter Dadam

Universität Ulm
Fakultät für Informatik
Abt. Datenbanken und Informationssysteme
D-89061 Ulm

Telefon: +49(0)731-50-24130
Fax: +49(0) 731-50-24134
E-Mail dadam@informatik.uni-ulm.de
Internet: www.informatik.uni-ulm.de/dbis