

Schema Evolution in Prozess-Management-Systemen*

Stefanie Rinderle

Abt. DBIS, Fakultät für Informatik, Universität Ulm
rinderle@informatik.uni-ulm.de

Abstract: Eine der wichtigsten Anforderungen an Prozess-Management-Systeme in der Praxis ist die Unverstützung von Prozessänderungen zur Laufzeit. Dies beinhaltet sowohl die Modifikation einzelner Prozessinstanzen (z.B. als Reaktion auf Ausnahmesituationen) als auch Änderungen auf Prozessstypenebene (z.B. Anpassung von Prozessen an neue gesetzliche Rahmenbedingungen). Bei Prozessstypänderungen ist es oft unerlässlich die Modifikationen auch auf die bereits laufenden Prozessinstanzen zu übertragen. Eine solche Änderungspropagation muss korrekt, effizient und benutzerfreundlich durchgeführt werden, d.h. ohne Laufzeitfehler zu verursachen, ohne das System unnötig lange zu blockieren und ohne die Benutzer mit komplexen Instanzadaptionen zu belasten. Diese Anforderungen werden noch deutlich verschärft, wenn die Änderungen auf Prozessinstanzen propagiert werden sollen, die selbst schon Gegenstand einer Ad-hoc-Änderung waren. Dann müssen zusätzlich potentielle Wechselwirkungen zwischen Prozessstyp- und Prozessinstanzänderungen analysiert und fortschrittliche Migrationsstrategien angeboten werden. Hierzu bieten wir ein formales Rahmenwerk für das komplette Spektrum des Zusammenspiels von Prozessstyp- und Prozessinstanzänderungen an unter besonderer Berücksichtigung der Aspekte Korrektheit, Vollständigkeit, Effizienz und Benutzerfreundlichkeit. Zusätzlich demonstriert ein prototypisch implementiertes Prozess-Management-System die Machbarkeit der entwickelten Konzepte.

1 Motivation

Die fortschreitende Globalisierung der Märkte in Verbindung mit E-Business stellt in Zukunft hohe Anforderungen an alle, die am Marktgeschehen teilnehmen. Die optimale Gestaltung und Beherrschung der Geschäftsprozesse sowie die Fähigkeit, diese rasch und kostengünstig an neue Gegebenheiten anzupassen, wird für viele Unternehmen zur Überlebensfrage werden. Fundamental hierfür ist die konsequente prozessorientierte Ausrichtung der betrieblichen Informationssysteme, d.h. die strikte Trennung von Prozesslogik und Anwendungscode sowie die explizite Steuerung der Prozesse durch ein Prozess-Management-System.

Sollen Prozess-Management-Systeme in umfassender Weise für die rechnerbasierte Verwaltung und Steuerung von Geschäftsprozessen einsetzbar sein, müssen die von ihnen verwalteten Prozessschemata und Prozessinstanzen bei Bedarf rasch anpassbar sein [vB02,

*Der Originaltitel der Dissertation lautet „Schema Evolution in Process Management Systems“.

RRD04a]. Solche Änderungen können einen Prozesstyp (bzw. sein Schema) als Ganzes (z.B. bei Änderung gesetzlicher Rahmenbedingungen) oder auch nur einzelne Instanzen (z.B. Gabe eines zusätzlichen, für den Normalfall nicht vorgesehenen Medikaments bei Patient Müller) betreffen. Abbildung 1 zeigt einen medizinischen Behandlungsprozess, dessen Prozessschema durch das Einfügen der Aktivität **Labortest** und das Löschen der Aktivität **Patient aufklären** modifiziert wird.

Bei solchen Änderungen auf Prozesstyp-Ebene wird man in der Regel fordern, dass die auf Basis des alten Prozessschemas erzeugten Instanzen auch nach Änderung dieses Schemas ungestört weiterlaufen können. Dies lässt sich z. B. durch geeignete Versionierungskonzepte erreichen. Dieser einfache Ansatz ist für Prozesse kurzer Dauer meist ausreichend, wirft aber im Zusammenhang mit lang laufenden Prozessen, wie sie z. B. im Krankenhaus-, Engineering- oder Finanz-Sektor auftreten, Probleme auf. Hier ist in vielen Fällen eine Fortführung der Prozesse auf Grundlage des alten Prozessschemas nicht akzeptabel, etwa wenn dadurch gesetzliche Vorschriften oder Geschäftsregeln des Unternehmens (z. B. Behandlungsrichtlinien eines Krankenhauses) verletzt werden.

Aus diesen Gründen besteht von Anwenderseite der Wunsch, die auf Prozesstyp-Ebene festgelegten Änderungen - wo sinnvoll und möglich - auch auf die bereits (vielleicht in großer Zahl) laufenden Prozessinstanzen zu übertragen [RRD04c]. Wir sprechen in diesem Zusammenhang auch von der Propagation einer Prozesstyp-Änderung auf laufende Prozessinstanzen bzw. von der Migration verträglicher Prozessinstanzen auf das geänderte Prozessschema (siehe Abb. 1). Dies bei Bedarf zu können, und zwar ohne, dass es dadurch auf Instanzebene in der Folge zu Inkonsistenzen oder Fehlern kommt, ist ungemein wichtig, wenn ein Prozess-Management-System breit und umfassend einsetzbar sein soll.

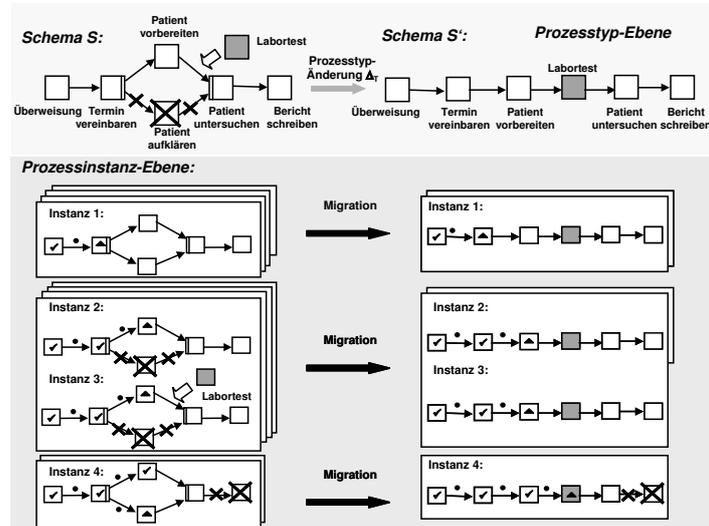


Abbildung 1: Prozessschema-Evolution mit Migration unverzerrter und verzerrter Prozessinstanzen

Neben Änderungen auf Prozesstyp-Ebene muss ein Prozess-Management-System, wenn es flexibel (z.B. in Ausnahmesituationen) einsetzbar sein soll, auch die (Ad-hoc) Modifikation einzelner Prozessinstanzen zur Laufzeit erlauben. Beispielsweise wurde für Instanz 4 in Abb. 1 die Aktivität **Bericht schreiben** dynamisch zur Laufzeit der Instanz gelöscht. Kommt es nachfolgend zu einer Prozesstyp-Änderung, stellt sich die grundlegende Frage, ob und wann die bereits individuell modifizierten Prozessinstanzen auf das geänderte Prozessschema migriert werden können. Ein flexibles Prozess-Management-System muss auch das Zusammenspiel solcher nebenläufig auftretenden Prozesstyp- und Prozessinstanz-Änderungen adäquat unterstützen.

Heutige auf dem Markt verfügbare Prozess-Management-Systeme erlauben es jedoch entweder gar nicht, die Änderungen eines Prozesstyps auf bereits laufende Prozessinstanzen zu übertragen, oder aber dies kann in der Folge zu Inkonsistenzen oder gar Systemabstürzen führen. Dieser Mangel ist ein wesentlicher Grund für die immer noch geringe Verbreitung dieser Systeme. Auch Ansätze aus der Forschung springen in vielerlei Hinsicht zu kurz, etwa hinsichtlich Benutzerfreundlichkeit oder Effizienz. Außerdem existiert weder ein kommerzielles System noch ein theoretischer Ansatz, der das Zusammenspiel von Prozesstyp- und Prozessinstanz-Änderungen erlaubt.

2 Zusammenspiel von Prozesstyp- und Prozessinstanzänderungen

Die vorliegende Arbeit bietet zum ersten Mal ein umfassendes formales Rahmenwerk für die Unterstützung von Prozesstyp- und Prozessinstanz-Änderungen im laufenden Betrieb. Darauf basierend können Prozesstyp-Änderungen in korrekter und effizienter Weise „per Knopfdruck“, auf die möglicherweise große Zahl von sich in Ausführung befindlichen Prozessinstanzen propagiert werden und zwar völlig unabhängig davon, ob diese bereits individuell modifiziert wurden oder nicht.

Grundlegend hierbei ist die Unterscheidung zwischen zwei Arten von Prozessinstanzen: solchen, die noch basierend auf ihrem Original-Prozessschema laufen (unverzerrte Prozessinstanzen) und solchen, die bereits individuell modifiziert worden sind (verzerrte Prozessinstanzen). Beispielsweise läuft Instanz 1 in Abb. 1 unverzerrt auf Schema *S* während die Instanzen 2, 3 und 4 im Verlauf ihrer Ausführung individuell modifiziert worden sind. Als Fundament stellt die vorliegende Dissertation zunächst einen Ansatz zur Migration unverzerrter Prozessinstanzen auf das geänderte Prozessschema vor. Kernstück bildet ein umfassendes Korrektheitskriterium, das sowohl für ausdrucksmächtige Prozessmetamodelle¹ als auch für eine vollständige Menge an Änderungsoperationen angewendet werden kann. Die Zusicherung dieses Kriteriums verhindert, dass es nach der Migration der Prozessinstanzen auf das geänderte Prozessschema zu Problemen (z.B. Programmabstürzen infolge unversorgter Eingabeparameter oder inkonsistente Prozessinstanz-Zustände) kommt. Solche Probleme treten häufig in Verbindung mit komplexen Änderungen auf. Ein Beispiel hierfür zeigt Abb. 2: Auf Prozesstypebene werden zwei Aktivitäten inklusive einer Daten-

¹z.B. für die von uns entwickelten WSM-Netze, aber auch für die von kommerziellen Systemen verwendeten Aktivitäten- oder BPEL-Netze

abhängigkeit zwischen ihnen eingefügt (Aktivität **Fragebogen versenden** schreibt das Datenelement **Wunsch**, welches wiederum von Aktivität **Geschenk versenden** als obligater Eingabeparameter gelesen wird). Propagiert man diese Prozesstyp-Änderung ohne weitere Prüfung der Korrektheit auf Instanz *I*, resultiert dies in einem inkonsistenten Prozessinstanz-Zustand: Die Aktivität **Fragebogen versenden** wird nicht ausgeführt, weshalb das Datenelement **Wunsch** nicht geschrieben wird. Infolgedessen kommt es zu einer fehlerhaften Versorgung der Eingabedaten von Aktivität **Geschenk versenden** was schlimmstenfalls zu einem Absturz der mit dieser Aktivität assoziierten Anwendungskomponente führen kann. Das in der Arbeit vorgestellte Kriterium würde für das in Abb. 2 dargestellte Beispiel die Migration von Instanz *I* korrekterweise verbieten.

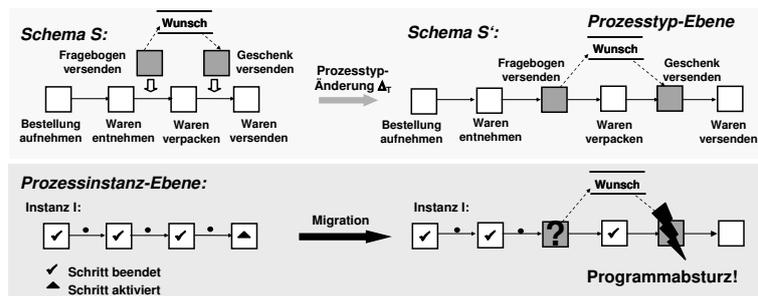


Abbildung 2: Inkonsistenter Zustand nach unkontrollierter Prozessinstanz-Migration

Damit die Korrektheitsprüfungen zur Laufzeit auch bei einer sehr großen Anzahl laufender Prozessinstanzen (beispielsweise sind in bestimmten Umgebungen, etwa Großkliniken, oft mehrere zehntausend Prozessinstanzen aktiv) effizient zugesichert werden kann, bietet unser Ansatz präzise Zustandsbedingungen, welche für alle Arten von Änderungsoperationen (additive, subtraktive und ordnungsverändernde Operationen sowie komplexe Kontroll- und Datenflussänderungen) die effiziente Überprüfung des Korrektheitskriteriums ermöglichen. Insbesondere werden sowohl teure Log Scans als auch komplexe Erreichbarkeitsanalysen (wie in vielen Ansätzen aus der Literatur vorgeschlagen) vermieden. Zusätzlich werden in der Arbeit Algorithmen zur automatischen Anpassung der verträglichen Prozessinstanzen nach ihrer Migration auf das geänderte Prozessschema präsentiert [Rin04] (was für bestimmte Formalismen, z. B. Petri-Netze, im Allgemeinen nicht möglich ist).

Ausgehend vom Basisfall der Migration unverzerrter Prozessinstanzen sollen Prozesstyp-Änderungen auch auf verzerrte Prozessinstanzen anwendbar sein. Dazu werden die verzerrten Prozessinstanzen zunächst entlang des Überlappungsgrades zwischen instanz-spezifischer Änderung („Verzerrung“) und der Prozesstyp-Änderung klassifiziert [RRD04b]. Der Grund hierfür ist, dass Prozessinstanzen, für welche instanz-spezifische Änderung und Prozesstyp-Änderung disjunkt sind, eine andere Migrationsstrategie erfordern als Prozessinstanzen, für welche diese Änderungen überlappende „Effekte“ auf das ursprüngliche Prozessschema haben. Für Prozessinstanzen der ersten Kategorie (disjunkte Prozessinstanz- und Prozesstyp-Änderungen (z.B. Instanz 4 in Abb. 1) wird eine geeignete Erweiterung des (Basis-) Korrektheitskriteriums vorgenommen, welche neben Statuskonflikten auch

strukturelle Konflikte berücksichtigt. Beispielsweise enthält das resultierende Schema für Instanz 2 in Abb. 3 nach einer unkontrollierten Propagation der Prozessstyp-Änderung einen unerwünschten Zyklus, der nachfolgend eine Verklemmung bei der Ausführung von Instanz 2 verursacht. Um die Existenz solcher struktureller Konflikte effizient auszuschließen, werden in der Arbeit schnell überprüfbare Konflikttests präsentiert. Für das Szenario aus Abb. 3 beispielsweise kann der in der Arbeit vorgestellte Verklemmungstest basierend auf dem ursprünglichen Schema S angewendet werden. Dieser Test entscheidet mittels einfacher Vorgänger- / Nachfolgerbeziehungen etwa, dass das resultierende Schema für Instanz 2 einen verklemmungsverursachenden Zyklus enthalten wird. Dadurch werden die aufwendige Materialisierung der zu prüfenden Schemata und die anschließenden, zum Teil komplexen Korrektheitstests (z.B. Erreichbarkeitsanalysen zur Erkennung potentieller Verklemmungen) vermieden.

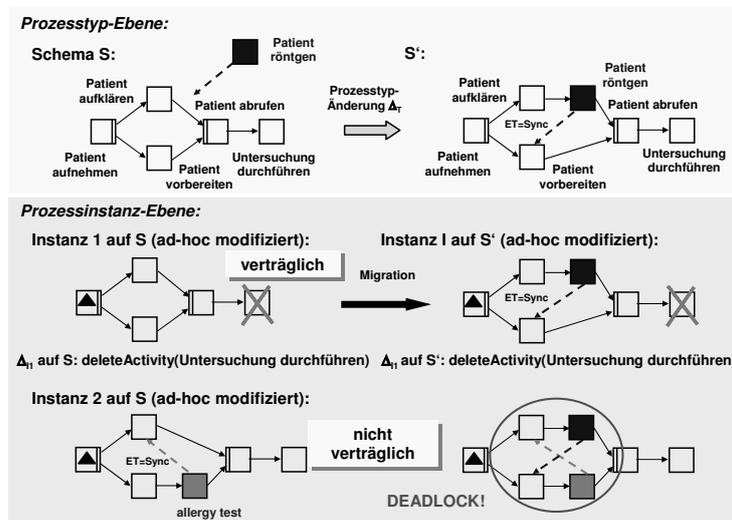


Abbildung 3: Struktureller Konflikt nach Migration einer Prozessinstanz mit disjunkter Verzerrung

Schließlich werden in der Dissertation adäquate Strategien für die Migration von Prozessinstanzen mit disjunkter Verzerrung präsentiert. Dabei wird auch in geeigneter Weise der Herausforderung begegnet, wie die jeweilige instanzspezifische Änderung nach der Migration auf das geänderte Prozessstyp-Schema korrekt bestimmt werden kann. Hierbei kommt die in der Arbeit präsentierte formale Definition disjunkter Änderungen zum Tragen, insbesondere die Eigenschaft der Kommutativität zwischen solchen Änderungen. Ein Beispiel zeigt Abb. 3. Instanz 1 wurde nach dem erweiterten Korrektheitskriterium als verträglich gemäß ihres Zustands und ihrer Struktur eingestuft. Nach der in der Dissertation ausgearbeiteten Migrationsstrategie für Instanzen mit disjunkter Verzerrung wird Instanz 1 auf das geänderte Schema S' „umgehängt“. Da die instanzspezifische Änderung Δ_{I1} und die Prozessstyp-Änderung Δ_T bezogen auf das ursprüngliche Schema S kommutativ sind, kann Δ_{I1} nach der Migration von I auf S' unverändert beibehalten werden (siehe Abb. 3).

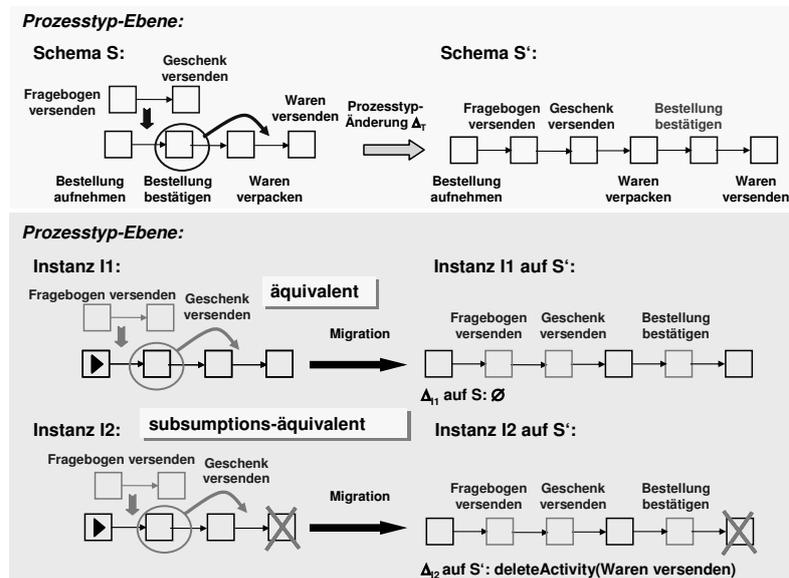


Abbildung 4: Überlappende Prozessstyp- und Prozessinstanz-Änderungen

Neben Prozessinstanzen mit disjunkter Verzerrung ist man in der Praxis häufig auch mit Prozessinstanzen konfrontiert, deren instanzspezifische Verzerrung mit der Prozessstyp-Änderung überlappt. Dies ist beispielsweise der Fall, wenn Optimierungen auf Prozessstyp-Ebene durch Ad-hoc-Änderungen auf Prozessinstanz-Ebene teilweise oder komplett antizipiert werden. Für solche Prozessinstanzen, deren Verzerrung die Prozessstyp-Änderung überlappt, wird eine ausgefeilte Klassifikation vorgestellt, die von sich nur teilweise überlappenden Änderungen (z.B. für Instanz *I2* in Abb. 4) bis zu äquivalenten Änderungen reicht (z.B. für Instanz *I1* in Abb. 4). Diese anhand des Überlappungsgrades zwischen Prozessstyp- und Prozessinstanz-Änderungen vorgenommene Klassifikation stellt die Basis für die Formulierung geeigneter Migrationsstrategien dar. Beispielsweise kann die Instanz *I1* in Abb. 4 mit ihrer zur Prozessstyp-Änderung äquivalenten Verzerrung auf das geänderte Prozessstyp-Schema S' migriert werden. Nach der Migration wird die für Instanz *I1* auf S' resultierende instanzspezifische Verzerrung Δ_{I1} leer. Dagegen subsumiert die instanzspezifische Verzerrung von Instanz *I2* die Prozessstyp-Änderung und erweitert sie um das dynamische Löschen der Aktivität *Waren versenden* (siehe Abb. 4). Deshalb muss für Instanzen mit Verzerrungen dieses Überlappungsgrades eine andere Migrationsstrategie gewählt werden. Wie in der Dissertation vorgeschlagen, wird Instanz *I2* auf das geänderte Prozessstyp-Schema migriert und die Differenz zwischen Prozessstyp- und Prozessinstanz-Änderung als instanzspezifische Verzerrung auf S' gespeichert (auch für die Berechnung der Differenzen zwischen Änderungen wird in der Arbeit ein Algorithmus geboten).

Auch für alle anderen Klassen von Überlappungsgraden zwischen Prozessstyp- und Prozessinstanzänderungen bietet der Ansatz (automatische) Migrationsstrategien oder zumin-

dest Verhaltensregeln zur Benutzerunterstützung, falls keine eindeutige Migrationsstrategie vorgeschlagen werden kann. Die Migrationsstrategien beinhalten nicht nur das automatische „Umhängen“ der verträglichen Prozessinstanzen auf das geänderte Prozessschema, sondern auch die erforderlichen Struktur- und Zustandsanpassungen. Um den Grundstein für eine fehlerfreie Implementierung der Konzepte zur Prozessschema-Evolution zu legen, wurden in der Dissertation alle Begriffe und Aussagen formal definiert und bewiesen.

3 Prototypische Implementierung

Der im Rahmen der Dissertation entstandene Software-Prototyp demonstriert eindrucksvoll die Vollständigkeit, Effizienz und Benutzerfreundlichkeit der entwickelten Konzepte. Abbildung 5 zeigt die Funktionsweise des Demonstrators im Zusammenhang mit der Migration unverzerrter Prozessinstanzen. Auf dem ursprünglichen Prozesstyp-Schema wurden 2502 Prozessinstanzen gestartet und befinden sich in Ausführung. Nach Änderung des Prozesstyp-Schemas durch Einfügen der Aktivität `Labortest` und Löschen der Aktivität `Patient` aufklären (vgl. Abb. 1) bestimmt das System automatisch, welche Prozessinstanzen korrekt auf das geänderte Prozesstyp-Schema migriert werden können. In Abb. 5 laufen die verträglichen Instanzen nach ihrer Migration auf der geänderten Version V2 des Prozesstyp-Schemas. Die nicht verträglichen Prozessinstanzen werden basierend auf der ursprünglichen Version V1 des Prozesstyp-Schemas fortgeführt.

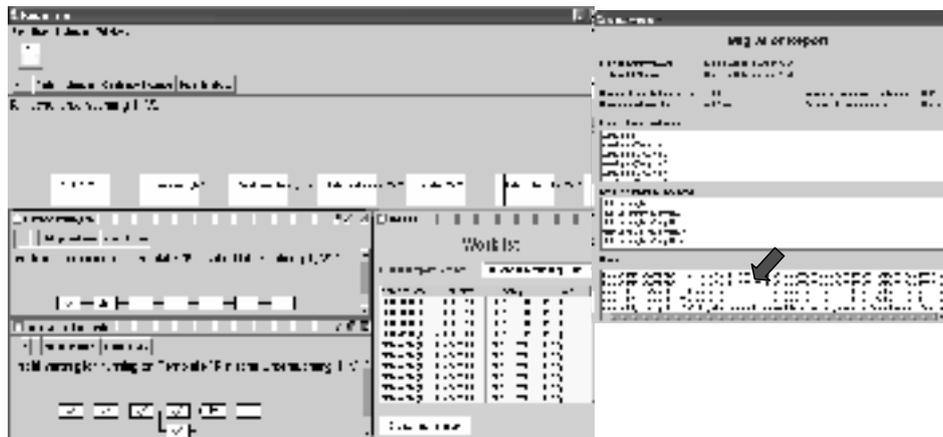


Abbildung 5: Migration unverzerrter Prozessinstanzen und Migrationsreport

Bei jeder Prozessschema-Evolution wird vom Demonstrator ein sogenannter Migrationsreport generiert. Dieser bietet dem Benutzer umfangreiche Statistiken, etwa zur Anzahl der verträglichen und nicht verträglichen Prozessinstanzen und zur Zeitdauer der Verträglichkeitsprüfungen und Prozessinstanz-Anpassungen. Der Migrationsreport für das vorgestellte Szenario ist in Abb. 5 dargestellt. Beispielsweise benötigen die Verträglichkeitsprü-

fungen für 2502 Prozessinstanzen nur 10 ms. Des Weiteren gibt der Migrationsreport Aufschluss darüber, warum Prozessinstanzen von einer Migration ausgeschlossen wurden: unverzerrte Prozessinstanzen können nicht auf das geänderte Prozesstyp-Schema migriert werden, wenn sie in ihrer Ausführung bereits zu weit fortgeschritten sind.

The image shows a software interface with two main panels. The left panel, titled 'Demonstrator', displays a process flow diagram with steps like 'Start (ID: 0)', 'admit patient (ID: 2)', 'instruct patient (ID: 4)', and 'prepare patient (ID: 5)'. A dialog box 'Change History of Instance' is open over a 'compliant (adhoc modified) (adhoc modified) running' instance, with a 'delete' button circled. The right panel, titled 'MigrationReport', contains the following data:

MigrationReport			
From TemplateVersion:	clinical_synctest, V1		
To TemplateVersion:	clinical_synctest, V2		
Number of migrated instances:	751	Number of not migrated instances:	5
Duration of all migration:	1582 ms	Duration of compliance check:	6
List of migrated instances:			
compliant (adhoc modified)			
compliant (adhoc modified)(Copy No. 1)			
compliant (adhoc modified)(Copy No. 2)			
compliant (adhoc modified)(Copy No. 3)			
compliant (adhoc modified)(Copy No. 4)			
List of not migrated instances:			
noncompliant (adhoc modified)			
noncompliant (adhoc modified)(Copy No. 1)			
noncompliant (adhoc modified)(Copy No. 2)			
noncompliant (adhoc modified)(Copy No. 3)			
noncompliant (adhoc modified)(Copy No. 4)			
Notes:			
[noncompliant (adhoc modified)(Copy No. 496)] Migration canceled, because there is a deadlock cau			
[noncompliant (adhoc modified)(Copy No. 497)] Migration canceled, because there is a deadlock cau			
[noncompliant (adhoc modified)(Copy No. 498)] Migration canceled, because there is a deadlock cau			
[noncompliant (adhoc modified)(Copy No. 499)] Migration canceled, because there is a deadlock cau			
[noncompliant (adhoc modified)(Copy No. 500)] Migration canceled, because there is a deadlock cau			
Overlap classes:			
compliant (adhoc modified)(Copy No. 55): DISJOINT			
compliant (adhoc modified)(Copy No. 257): DISJOINT			
compliant (adhoc modified)(Copy No. 361): DISJOINT			
compliant (adhoc modified)(Copy No. 689): DISJOINT			
compliant (adhoc modified)(Copy No. 654): DISJOINT			

Abbildung 6: Migration von Prozessinstanzen mit disjunkter Verzerrung

Der Demonstrator unterstützt neben Änderungen auf Prozesstyp-Ebene auch Ad-hoc-Modifikationen einzelner Prozessinstanzen. In Abb. 6 ist das Szenario für die Migration von Prozessinstanzen mit disjunkter Verzerrung aus Abb. 3 dargestellt. Wie aus dem Migrationsreport zu erkennen ist, werden die verzerrten Prozessinstanzen zunächst korrekt als Prozessinstanzen mit disjunkter Verzerrung eingestuft (siehe Abb. 6). Dann prüft das System, ob für die Prozessinstanzen strukturelle und statusbezogene Verträglichkeit vorliegt. Wie die abstrakte Darstellung des Szenarios in Abb. 3 zeigt, ergibt sich für Instanz 2 nach ihrer Migration eine Verklemmung. Dies wird vom Demonstrator korrekt erkannt (siehe Report in Abb. 6) und die entsprechenden Prozessinstanzen werden weiterhin nach ihrem ursprünglichen Schema ausgeführt. Die verträglichen Prozessinstanzen werden auf das geänderte Prozesstyp-Schema migriert und behalten ihre instanzspezifische Änderung bei.

Auch die Migration von Prozessinstanzen, deren instanzspezifische Verzerrung die Prozesstyp-Änderung überlappt, wird von dem im Rahmen der Dissertation implementierten Demonstrator korrekt und umfassend unterstützt. Ein Beispiel zeigt Abb. 7. Die Prozesstyp-Änderung und die Verzerrung von Prozessinstanz 1 sind äquivalent (wie auch durch die abgebildeten Änderungshistorien angedeutet), während die Prozesstyp-Änderung die Verzerrung von Prozessinstanz 2 subsumiert. Diese Überlappungsgrade werden korrekt im

Migrationsreport dokumentiert. Außerdem werden die entsprechenden Prozessinstanzen gemäß der in der Dissertation ausgearbeiteten Migrationsstrategien auf das geänderte Prozessstyp-Schema migriert (siehe Migrationsreport in Abb. 7).



Abbildung 7: Migration von Prozessinstanzen mit überlappender Verzerrung

Zu erwähnen bleibt, dass die in der vorliegenden Dissertation ausgearbeiteten Migrationsstrategien dem Benutzer nur „per default“ vorgeschlagen werden, d.h., der Benutzer wird vom System nicht gezwungen einer bestimmten Strategie zu folgen. Es ist z.B. möglich, bestimmte Prozessinstanzen von einer Migration auszuschließen (z.B. solche, die einen bestimmten Meilenstein bereits überschritten haben) oder eine andere Anpassungsstrategie nach der Migration auf das geänderte Prozessstyp-Schema vorzunehmen.

Die bei der Implementierung des Demonstrators gewonnenen Erfahrungen werden in die Entwicklung des neuen ADEPT2 Prozess-Management-Systems einfließen. Der erste in der Abteilung DBIS entwickelte ADEPT-Prototyp, ein mächtiges und flexibles Prozess-Management-System, wurde im Jahr 2003 mit dem doIT-Software-Award des Landes Baden-Württemberg ausgezeichnet. Aktuell wird im Rahmen des AristaFlow-Projektes - einer vom Land Baden-Württemberg geförderten Kooperation der Universitäten Ulm und Mannheim sowie diversen Industriepartnern (DaimlerChrysler AG, SAP AG, Wilken GmbH, All-for-One Systemhaus AG) - das Nachfolgesystem (ADEPT2) mit der integrierten Fähigkeit zur Prozessschema-Evolution implementiert. Die Grundlage hierzu hat die vorliegende Dissertation geschaffen.

4 Zusammenfassung

Insgesamt ermöglicht das von uns entwickelte Rahmenwerk, laufende Prozessinstanzen unabhängig davon, ob sie bereits individuell modifiziert wurden oder nicht, korrekt, effizient und unter adäquater Einbeziehung des Benutzers „per Knopfdruck“ auf ein geändertes Prozessstyp-Schema zu migrieren. Lediglich Prozess-Instanzen, deren Migration zu einem inkonsistenten Instanzschema führen würde, werden von der Migration ausgenommen, d.h. sie werden auf Basis des alten Prozessschemas zu Ende geführt.

Die Umsetzung dieser Konzepte eröffnet ganz neue Möglichkeiten in Bezug auf die raschere Realisierung und Adaption von prozessorientierten Anwendungen. Zusätzlich trägt die Arbeit durch Bereitstellung einer formalen Basis zur drastischen Reduzierung der Fehleranfälligkeit von Prozess-Management-Systemen bei, wodurch eine robuste Ausführung prozessorientierter Anwendungen ermöglicht wird. Zusammenfassend wird durch die vorliegende Arbeit die Abbildung sich ständig ändernder Geschäftsprozesse in der Realität auf deren elektronische Unterstützung durch ein Prozess-Management-System ermöglicht.

Literatur

- [Rin04] S. Rinderle. *Schema Evolution in Process Management Systems*. Dissertation, Universität Ulm, 2004. (in Englisch).
- [RRD04a] S. Rinderle, M. Reichert und P. Dadam. Correctness Criteria for Dynamic Changes in Workflow Systems – A Survey. *DKE*, 50(1):9–34, 2004.
- [RRD04b] S. Rinderle, M. Reichert und P. Dadam. Disjoint And Overlapping Process Changes: Challenges, Solutions, Applications. In *Proc. Int’l CoopIS’04*, LNCS 3290, Seiten 101–120, Agia Napa, Cyprus, October 2004.
- [RRD04c] S. Rinderle, M. Reichert und P. Dadam. Flexible Support Of Team Processes By Adaptive Workflow Systems. *DPD*, 16(1):91–116, 2004.
- [vB02] W.M.P v.d. Aalst und T. Basten. Inheritance of Workflows: An Approach to Tackling Problems Related to Change. *Theoret. Comp. Science*, 270(1-2):125–203, 2002.

Stefanie Rinderle studierte Wirtschaftsmathematik an der Universität Augsburg. Seit ihrem Diplom im November 2000 ist sie wissenschaftliche Mitarbeiterin in der Abteilung Datenbanken und Informationssysteme der Fakultät für Informatik an der Universität Ulm. Dort promovierte sie im Dezember 2004 zum Dr. rer. nat. (mit Auszeichnung). Von März bis Mai 2005 absolviert Stefanie Rinderle einen PostDoc-Aufenthalt an der Universität Twente in den Niederlanden, wo sie an einem Projekt für Daimler Chrysler Research in Ulm arbeitet. Daran schließt sich ein weiterer dreimonatiger PostDoc-Aufenthalt an der Université d’Ottawa, École de gestion in Kanada mit Schwerpunkt auf der Formalisierung von E-Negotiation-Prozessen an. Ihre Interessen liegen in den Bereichen adaptives Prozess-Management, lernende Prozesse und semantische Benutzerunterstützung, Sicherheitsaspekten und Modellierung und Analyse von Geschäftsprozessen.