

# ProCycle – Integrierte Unterstützung des Prozesslebenszyklus

Barbara Weber, Werner Wild, Manfred Reichert, Peter Dadam

Aufgrund häufiger Änderungen in ihrem Geschäftsumfeld müssen Unternehmen in der Lage sein, ihre Geschäftsprozesse und die sie unterstützenden Informationssysteme (IS) rasch und flexibel anzupassen. In jüngerer Vergangenheit ist eine neue Generation von IS entstanden, die eine umfassende IT-Unterstützung für Prozesse zum Ziel haben. Diese Systeme haben sich jedoch in der Praxis oftmals als zu starr erwiesen. Um wirklich breit einsetzbar zu sein, muss es auch möglich sein, in Ausnahmefällen flexibel vom definierten Prozess abzuweichen sowie Prozessimplementierungen rasch an sich ändernde Rahmenbedingungen, wie beispielsweise neue gesetzliche Anforderungen, anzupassen. Das prozessorientierte IS sollte den Benutzer im Änderungsfall durch die Wiederverwendung von Änderungen unterstützen sowie den Prozessverantwortlichen bei der Ableitung von verbesserten Prozessmodellen helfen. Dieser Beitrag stellt mit ProCycle einen implementierten Ansatz vor, der all diese Funktionalitäten bietet und der aus der Integration adaptiver Prozess-Management-Technologie mit Techniken des Case-Based Reasoning hervorgegangen ist. Ziel ist es, den kompletten Prozesslebenszyklus in integrierter Form zu unterstützen.

## 1 Einleitung

Die adäquate IT-Unterstützung ihrer Geschäftsprozesse stellt für viele Unternehmen eine große Herausforderung dar. Aufgrund häufiger Änderungen in ihrem Geschäftsumfeld müssen sie in der Lage sein, ihre Informationssysteme (IS) an den betrieblichen Prozessen auszurichten. Weiters sollten die IS eine flexible Prozessunterstützung ermöglichen, gleichzeitig aber ein gewisses Maß an Kontrolle bieten [1]. So müssen zur Laufzeit Ad-hoc-Änderungen einzelner Prozessinstanzen (z.B. Einfügen oder Verschieben von Prozess-Schritten) möglich sein, ohne dass dies die Robustheit und Stabilität des Gesamtsystems beeinträchtigt. Nur dann ist eine realitätsnahe Prozessunterstützung erzielbar und eine Diskrepanz zwischen rechnergestützten *Workflows* und tatsächlich gelebten Prozessen vermeidbar. Eine neue Generation prozessorientierter IS (engl. *Process-Aware IS*; PAIS) versucht dieser Herausforderung zu begegnen, indem Prozesse explizit beschrieben und damit Prozesslogik und Anwendungscode separiert werden [2]. Unternehmen versprechen sich dadurch eine raschere Anpassbarkeit ihrer Prozesse, d.h. mehr Agilität.

In der Vergangenheit konnte diese Flexibilität aber nur ansatzweise realisiert werden, da existierende Technologien (z. B. Workflow-Management-Systeme) entweder keine oder nur stark eingeschränkte Prozessänderungen zur Laufzeit zulassen. Einen viel versprechenden Ansatz bieten allerdings adaptive Prozess-Management-Systeme wie ADEPT [1] oder WASA2 [3], die sich speziell der Änderungsproblematik von Prozessen und deren Behandlung zur Laufzeit verschrieben haben. Sie erlauben es autorisierten Benutzern zur Laufzeit, flexibel vom vordefinierten Prozess abzuweichen, etwa durch dynamisches Einfügen oder Überspringen eines Prozess-Schrittes (z. B. zwecks Behandlung eines Ausnahmefalles). Weiters gestatten es diese Systeme, die Implementierung eines PAIS rasch an veränderte Gegebenheiten anzupassen. Im letztgenannten Fall können unter gewissen Voraussetzungen sogar bereits laufende Prozessinstanzen noch auf die neue Abwicklung umgestellt werden.

Obwohl adaptive Prozess-Management-Technologie eine flexible Änderung von Prozessen sowohl auf Typ- als auch Instanzebene ermöglicht und somit die Grundlage für die systemseitige

Umsetzung von Prozessänderungen schafft, bietet sie noch keine Hilfestellung für die Nutzung vorhandenen Änderungswissens oder gar die Ableitung von Prozessverbesserungen. Hier eröffnet eine Integration mit Techniken des Case-Based Reasoning (CBR), wie sie z.B. in CBRFlow [4] vorgenommen wird, interessante Perspektiven. Konkret erlaubt es CBRFlow, Abweichungen vom vordefinierten Prozess mit Hilfe von CBR in annotierter Form in einer Fallbasis zu speichern und die resultierenden Fälle in ähnlichen Situationen wiederzuverwenden. Dabei dienen häufig auftretende, ähnliche Abweichungen als Ansatzpunkt für die Ermittlung von Prozessverbesserungen.

In diesem Beitrag zeigen wir anhand der von uns im Projekt *ProCycle* vorgenommenen Integration von ADEPT und CBRFlow exemplarisch auf, wie durch Zusammenführung von adaptiver Prozess-Management-Technologie und CBR-Techniken eine vollständige Unterstützung des Prozesslebenszyklus möglich wird. Neben der Durchführung von Prozessänderungen erlaubt das integrierte System (kurz: ProCycle), aus Erfahrungen zu lernen. Einerseits werden Benutzer auf der Ebene einzelner Prozessinstanzen durch Wiederverwendung von Änderungsfällen unterstützt, andererseits werden auf Typ-Ebene die aus solchen Fällen gewonnenen Erfahrungen genutzt, um zu verbesserten Prozessen zu gelangen.

Dieser Artikel gliedert sich wie folgt: Kapitel 2 bietet Hintergrundinformationen in Bezug auf Prozessmanagement, die für das weitere Verständnis hilfreich sind. Kapitel 3 beschreibt die Unterstützung des Prozesslebenszyklus in ProCycle, eine Darstellung des implementierten Systems folgt in Kapitel 4. Kapitel 5 diskutiert verwandte Ansätze und Kapitel 6 fasst die erzielten Ergebnisse zusammen.

## 2 Grundlagen

PAIS erlauben es Benutzern, ihre Prozesse explizit zu modellieren, auszuführen und zu überwachen. Wir skizzieren nachfolgend die in diesem Zusammenhang gebotene Systemunterstützung.

## 2.1 Prozessmodellierung

Die Orchestrierung der einzelnen Schritte eines Prozesses (sog. *Aktivitäten*) basiert auf einem vordefinierten *Prozess-Schema*. Dieses legt die durchzuführenden Aktivitäten, deren Kontroll- und Datenflussbeziehungen, zugehörige Mitarbeiterzuordnungen sowie Geschäftsobjekte (zur Versorgung von Aktivitäteneingaben bzw. Übernahme von Aktivitätenausgaben) fest. Für jeden durch das PAIS zu unterstützenden Geschäftsprozess (z.B. Buchen einer Geschäftsreise oder Bearbeiten eines Kreditantrags) wird ein *Prozess-Typ* definiert, zu dem es verschiedene *Schema-Versionen* geben kann, welche die zeitliche Entwicklung des Prozess-Typs reflektieren. In Abb. 1 etwa sind Prozess-Schemata *S1* und *S1'* zwei unterschiedliche Versionen desselben Prozess-Typs.

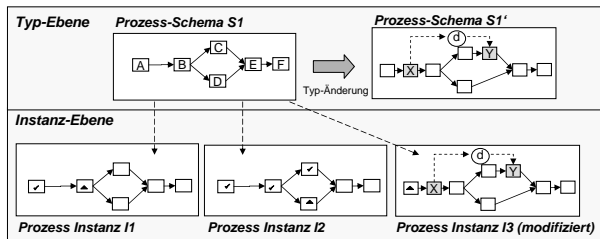


Abbildung 1: Typ- und Instanzebene in PAIS

## 2.2 Prozessausführung

Basierend auf einem Prozess-Schema können neue (*Prozess-*) *Instanzen* erzeugt, verwaltet und ausgeführt werden. Um zur Laufzeit angemessen auf Ausnahmesituationen reagieren zu können, müssen – wie eingangs erwähnt – Prozessteilnehmer zudem in der Lage sein, vom vordefinierten Ablauf abzuweichen [1]. Die Auswirkungen solcher instanzspezifischen Änderungen sollten lokal bleiben, d.h. es dürfen keine Effekte bezogen auf andere laufende Instanzen desselben Typs resultieren. In Abb. 1 wurde z.B. Instanz *I3* zur Laufzeit modifiziert, indem zwei Aktivitäten *X* und *Y* sowie eine Datenabhängigkeit zwischen ihnen eingefügt wurden. Durch diese Änderung weicht das Ausführungsschema der Instanz *I3* vom Ausgangsprozess-Schema *S1* ab.

Da in der Praxis dieselben oder ähnliche Abweichungen öfter auftreten können, sollte ein PAIS die Wiederverwendung von Änderungen unterstützen. Dies ist sinnvoll, weil die Neudefinition von Änderungen immer mit gewissem Aufwand verbunden ist und entsprechende Erfahrung seitens des Benutzers bedingt. Um Änderungen wiederverwenden und somit existierende Erfahrungen nutzen zu können, sollten die Instanz-Änderungen mit geeigneter *Kontextinformation* (z.B. zum Grund der Abweichung) annotiert werden. Diese Kontextinformation ist vom PAIS zu verwalten und kann dann dazu verwendet werden, um in einer aktuellen Problemsituation Fälle zu ähnlichen, bereits gelösten Problemen zu suchen. Ferner stellt Kontextwissen sicher, dass jeweils nur die für die aktuelle Situation des Benutzers relevanten Prozess-Änderungen zur Auswahl angeboten werden [4, 5].

## 2.3 Prozessmonitoring

Um die Ausführung von Prozessinstanzen und eventuelle Instanz-Änderungen nachvollziehbar zu machen, verwaltet ein PAIS ent-

sprechende *Ausführungs-* und *Änderungslogs*. Dadurch lassen sich das Ausführungsschema und der Zustand einer bestimmten Prozessinstanz für jeden beliebigen Zeitpunkt rekonstruieren. Ferner können die Log-Daten auch dazu genutzt werden, Vorschläge für Prozessverbesserungen abzuleiten [6, 7], etwa wenn eine bestimmte Ad-hoc-Änderung wiederkehrend aufgetreten ist. Die Umsetzung solcher Prozessverbesserungen erfolgt durch einen Prozessverantwortlichen auf Typebene und führt in der Regel zu einer neuen Schema-Version des entsprechenden Prozess-Typs (siehe Abb. 1). Üblicherweise basieren zukünftig erzeugte Instanzen dann auf dieser neuen Schema-Version. Im Falle von langlaufenden Prozessinstanzen kann es darüber hinaus sinnvoll bzw. notwendig sein, bereits laufende Instanzen auf die neue Schema-Version zu migrieren (siehe [8] für Details).

## 3 Unterstützung des Prozesslebenszyklus mit Case-Based Reasoning

Abb. 2 zeigt wie in ProCycle der komplette Prozesslebenszyklus durch Kombination von adaptiver Prozess-Management-Technologie und CBR durchgängig unterstützt wird.

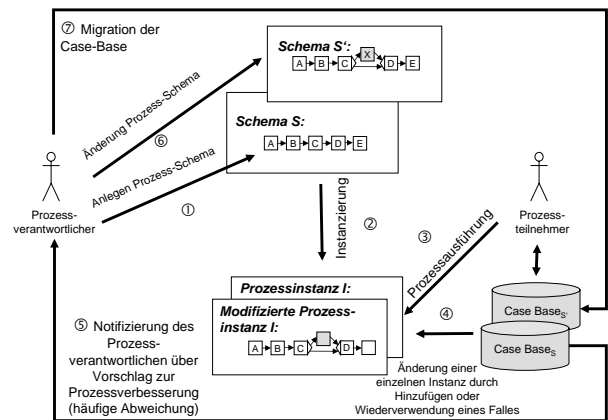


Abbildung 2: Integrierter Prozesslebenszyklus

Zum Modellierungszeitpunkt wird eine initiale Repräsentation des betreffenden Geschäftsprozesses erstellt. Dies kann entweder durch *Prozessanalyse* oder durch Anwendung von *Process Mining* (d.h. durch die Beobachtung von Prozessausführungen und deren Aktivitäten) erfolgen (1). Ausgehend von einem so gewonnenen Prozess-Schema können dann zur Laufzeit neue Instanzen erzeugt werden (2). Im Regelfall werden diese gemäß ihrem Ausgangsschema ausgeführt und die jeweils anstehenden Aktivitäten den Prozessteilnehmern über Arbeitslisten verfügbar gemacht (3). Treten Ausnahmen auf, die eine bestimmte Instanz betreffen, können Prozessteilnehmer instanzbezogen vom vordefinierten Schema abweichen. Benutzer können dabei entweder neue Abweichungen definieren und mit Angabe von Gründen in der Fallbasis (engl. *Case Base*; CB) speichern, oder sie können eine zu einem früheren Zeitpunkt in dieser CB abgelegte Abweichung wiederverwenden (4). Des Weiteren überwacht das PAIS, wie häufig ein bestimmtes Schema instanziiert wird und wie oft Abweichungen auftreten. Tritt eine bestimmte Abweichung sehr häufig auf, wird der Prozessverantwortliche darüber

informiert, dass eine Änderung auf Typ-Ebene evtl. Sinn macht (5). Er kann dann das Prozess-Schema entsprechend modifizieren und als neue Schema-Version abspeichern (6). Zusätzlich werden Fälle, die auch für die neue Schema-Version relevant bleiben, in eine neue Version der CB übergeführt (7).

### 3.1 Repräsentation und Wiederverwendung von Fällen

In diesem Abschnitt beschreiben wir, wie CBR in ProCycle eingesetzt wird, um die Semantik von Instanz-Änderungen zu beschreiben, und wie diese Änderungen gespeichert, gesucht und in ähnlichen Problemsituationen wiederverwendet werden. Zudem gehen wir auf Maßnahmen zur Qualitätssicherung von Fällen ein.

#### 3.1.1 Fallrepräsentation

Ein Fall repräsentiert in unserem Ansatz eine konkrete Laufzeitabweichung, die für eine oder mehrere Prozessinstanzen erfolgt ist. Neben der vorgenommenen Änderung selbst beschreibt der Fall auch den Kontext und die Gründe für die Abweichung. In der Regel werden neue Fälle immer dann ins System eingefügt, wenn ein Benutzer eine Abweichung vornehmen will und kein für eine Wiederverwendung geeigneter Fall im System vorliegt.

Ein konkreter Fall  $c$  umfasst eine textuelle Problembeschreibung  $pd_c$ , welche kurz die Ausnahmesituation erklärt, die zur Abweichung geführt hat. Die Gründe bzw. der Kontext für die Abweichung werden in unserem Ansatz als Menge von Frage-Antwort-Paaren  $qaSet_c$  repräsentiert, wobei jedes einzelne Frage-Antwort-Paar  $qa_i$  eine konkrete Bedingung (z.B. das Zutreffen eines medizinischen Problems) darstellt. Frage-Antwort-Paare werden von uns auch verwendet, um in einer gegebenen Situation die bekannten Fälle mit ähnlichem (Problem-)Kontext zu finden. Der Lösungsteil eines Falles  $sol_c$  schließlich besteht aus einer Liste mit Änderungsoperationen, die ausgeführt worden sind, um die vorliegende Ausnahmesituation zu beheben.

**Beispiel:** Nehmen wir an, dass ein Arzt die zur Diagnostik eines Kreuzbandrisses normalerweise durchzuführende Magnetresonanztomographie nicht anwenden kann, da der Patient einen Herzschrittmacher hat. Die Gründe für die Abweichung werden dann als Frage-Antwort-Paar beschrieben (Frage: Hat der Patient einen Herzschrittmacher?– Antwort: Ja). Der Lösungsteil umfasst die Änderungsoperationen am Prozess, die zur Behebung des Problems angewendet werden (z.B. Auslassen der Magnetresonanztomographie und Ausführen einer alternativen Untersuchung).

Die Frage eines Frage-Antwort-Paars ist immer Freitext, die Antwort kann entweder Freitext sein oder Ausdrücke enthalten. Ausdrücke erlauben es uns bereits im System vorhanden Kontextinformation zu nutzen und verhindern, dass Daten mehrfach erfasst werden müssen. Frage-Antwort-Paare, die Ausdrücke enthalten, werden automatisch vom System ausgewertet, indem Werte für Kontextattribute aus vorhandenen Datenquellen befüllt werden. Da diese Kategorie von Frage-Antwort-Paaren keine Benutzerinteraktion erfordert, können Fehler reduziert werden. Außerdem entfällt der Zeitaufwand für die Beantwortung der Fragen durch den Benutzer. Freitextantworten werden verwendet, wenn keine geeigneten Kontextattribute im System definiert sind

oder der Benutzer zu wenig erfahren ist, um Ausdrücke zu formulieren. Alle Informationen über Instanz-Änderungen, die ein bestimmtes Prozess-Schema betreffen, werden in einer schema-spezifischen CB gespeichert.

#### 3.1.2 Suchen und Wiederverwenden von Fällen

Wird zur Laufzeit eine Abweichung vom vordefinierten Prozess-Schema notwendig, initiiert der Benutzer einen Suchdialog in der CBR-Komponente, um Fälle mit ähnlicher Problemsituation zu finden. Für das Suchen wird von uns *Conversational Case-based Reasoning* (CCBR) verwendet. CCBR ist eine Erweiterung des CBR-Paradigmas und bindet den Benutzer aktiv in den Suchprozess ein [9]. Ein CCBR-System stellt ein interaktives System dar, welches den Benutzer durch das Stellen von Fragen und das Anzeigen relevanter Antwortmöglichkeiten bei der Suche unterstützt. Im Unterschied zu strukturiertem CBR muss keine vollständige Anfrage vor Beginn des Suchprozesses spezifiziert werden. Stattdessen assistiert das CCBR-System den Benutzer die aktuelle Situation zu erfassen und relevante Fälle zu finden, auch wenn er unter Umständen nur vage Vorstellungen darüber hat wonach er eigentlich sucht. Benutzer können vorhandene Informationen auf ihre Initiative hin jederzeit ins System eingeben. Aus diesem Grunde ist CCBR besonders geeignet, um unerfahrene Benutzer in Ausnahmesituationen zu unterstützen, die nicht automatisch abgehandelt werden können.

Beim Suchen nach ähnlichen Fällen präsentiert das System dem Benutzer eine Liste von Fragen; Fragen, für die es vordefinierte Antworten mit Ausdruck gibt, werden automatisch durch das System beantwortet. Andernfalls kann der Benutzer eine Freitextantwort auswählen. Für jeden Fall  $c$  in der CB wird die Ähnlichkeit zur aktuellen Problemsituation  $q$  berechnet. Die  $n$  ähnlichsten Fälle werden dann dem Benutzer, nach Ähnlichkeit absteigend sortiert, angezeigt. Die Ähnlichkeit wird dabei als Anzahl der korrekt beantworteten Fragen minus der Anzahl der inkorrekt beantworteten Fragen, dividiert durch die Gesamtanzahl der Fragen des Falls, berechnet [6].

$$sim_1(q, c) = \frac{same(qaSet_q, qaSet_c) - diff(qaSet_q, qaSet_c)}{|qaSet_c|}$$

Damit die Ähnlichkeit einen Wert zwischen 0 und 1 annimmt, nehmen wir noch folgende Normalisierung vor:

$$sim_2(q, c) = \frac{1}{2} * (sim_1(q, c) + 1)$$

Im Anschluss an die Neuanzeige gefundener Fälle kann der Benutzer (1) weitere Fragen in beliebiger Reihenfolge beantworten, (2) die Fallbasis filtern (z.B. nur Fälle mit einer bestimmten Änderungsoperation werden betrachtet), oder (3) einen der angezeigten Fälle zur Wiederverwendung selektieren. Entscheidet sich der Benutzer für *Option 1* (Beantwortung weiterer Fragen) wird die Ähnlichkeit entsprechend neu berechnet und die Liste mit den  $n$  ähnlichsten Fällen neu aufgebaut. Wählt der Benutzer *Option 2* (Filtern der CB) werden alle Fälle, die nicht dem Filterkriterium entsprechen, aus der Anzeigeliste entfernt. *Option 3* (Fall wiederverwenden) führt dazu, dass die Änderungsoperationen im Lösungsteil des selektierten Falls an das PAIS weitergeleitet werden und von diesem ausgeführt werden.

### 3.1.3 Qualität von Fällen

Die Wiederverwendbarkeit von Fällen wird maßgeblich durch die Qualität der in der Fallbasis gespeicherten Informationen bestimmt [11]. Die Fälle werden in der Regel durch normale Prozess Teilnehmer und nicht durch erfahrene Benutzer (z.B. Prozessverantwortliche) erfasst, so dass Maßnahmen zur Qualitätssicherung unumgänglich sind. Aspekte, die sich kritisch auf die Benutzerakzeptanz unseres CCBR-Ansatzes und damit auf die resultierende Informationsqualität auswirken können, sind die ggf. sehr langen Benutzerdialoge sowie inkonsistente oder fehlerhafte Fallinformationen.

**Fehlerfreiheit von Fällen.** Die Fallbasis wächst inkrementell; immer wenn neue Ausnahmesituationen auftreten, werden neue Fälle ins System hinzugefügt. Unser Ansatz garantiert dabei syntaktische Korrektheit des Lösungsteils des Falles, d.h. die Anwendung der Änderungsoperationen eines Falles auf das Ausführungsschema einer Prozessinstanz führt im Anschluss wieder zu einem korrekten Ausführungsschema (mit konsistentem Zustand) [1]. Syntaktische Korrektheit alleine reicht allerdings nicht aus. Da Fallinformationen oftmals von unerfahrenen Benutzern eingefügt werden, können auch (semantisch) fehlerhafte Fälle in das System gelangen. Deshalb muss sichergestellt werden, dass solche fehlerhaften Fälle keine hohe Wiederverwendung erfahren. In unserem Ansatz können die Fälle daher nach ihrer Anwendung auch hinsichtlich ihrer Brauchbarkeit bewertet werden. Basierend auf diesen Evaluierungsergebnissen wird ein *Reputation Score* errechnet, der Aufschluss darüber gibt, wie erfolgreich der Fall in der Vergangenheit angewendet wurde [10]. Der *Reputation Score* wird Benutzern im Suchdialog angezeigt (siehe Abb. 4). Fälle mit schlechten Evaluierungsergebnissen bzw. niedrigem *Reputation Score* können vom Prozessverantwortlichen deaktiviert werden.

**Refactoring von Fällen.** Um den Suchprozess zu beschleunigen und mögliche Fehleingaben auszuschalten, werden Antworten mit (aus dem Datenkontext automatisch auswertbaren) Ausdrücken den Freitextantworten vorgezogen. Allerdings wird es in der Praxis immer wieder Situationen geben, in denen die Eingabe von Freitext erforderlich wird, etwa wenn keine entsprechenden Kontextattribute im System definiert sind oder der Benutzer zu unerfahren ist, um Ausdrücke zu schreiben. Bei unserem Ansatz wird der Prozessverantwortliche informiert, wenn die Antworthäufigkeit einer Frage einen bestimmten Grenzwert überschreitet. Er wird dann darin unterstützt, ein Refactoring von Freitextantworten zu Ausdrücken vorzunehmen, soweit sinnvoll und möglich.

**Zusammenlegen von Fällen.** Generell ist es möglich, dass mehrere Fälle auf eine Prozessinstanz angewendet werden. Durch Analyse der Log-Daten können insbesondere Abhängigkeiten zwischen Fällen erkannt werden. Treten zwei Fälle immer in Kombination auf, kann die Problemlösungseffizienz des Systems durch Zusammenlegen der Fälle verbessert werden. Treten Fälle zwar nicht immer, aber sehr häufig in Kombination auf, sollte das System dem Benutzer bei der Auswahl eines Falles auf die Existenz eventuell abhängiger Fälle hinweisen (für Details siehe [11]).

### 3.1.4 Ableitung von Prozessverbesserungen aus Fällen

Tritt eine bestimmte Instanzmodifikation häufig auf, kann es sinnvoll sein, diese Abweichung auf Typ-Ebene zu übertragen. Dies bietet den Vorteil, dass das entsprechende Wissen für zukünftige

Instanzen sofort verfügbar wird und nicht mehr einzeln abgerufen und angewendet werden muss. Daher wird bei Überschreiten einer gewissen Wiederverwendungsrate der Prozessverantwortliche entsprechend benachrichtigt. Entscheidet er sich für die Übernahme der Instanzmodifikation(en), erstellt er eine neue Schema-Version; durch den Einsatz von ADEPT können auf Wunsch auch bereits laufende Instanzen auf das neue Schema migriert werden.

Die Erstellung einer neuen Schema-Version selbst ist kein automatisierter Prozess, sondern wird vom Prozessverantwortlichen unter Zuhilfenahme der CB vorgenommen. Das darin gesammelte Erfahrungswissen zu Ad-hoc-Abweichungen erleichtert den Änderungsprozess. Soll eine in einem Fall beschriebene Änderung immer (d.h. für alle zukünftigen Instanzen) wirksam sein, kann der Lösungsteil des Falls (d.h. die entsprechenden Änderungsoperationen) in der Regel ohne Anpassungen auf das Schema angewandt werden. Aufwendiger wird es, wenn eine Änderung nur in gewissem Kontext (d.h. nur für bestimmte Instanzen) wirksam werden soll. Hier muss der Lösungsteil des Falls um entsprechende Informationen (z.B. Auswahlprädikate) ergänzt werden. Wichtige Informationen hierzu kann der Prozessverantwortliche aus den Frage-Antwort-Paaren des Falles ziehen. Weiters wird der Prozessverantwortliche bei der Übernahme von Instanzmodifikationen auf ein Typ-Schema auch dadurch unterstützt, dass das System nur syntaktisch korrekte Änderungen zulässt. Alle verfügbaren Änderungsoperationen besitzen formale Vor- und Nachbedingungen, welche deren korrekte Anwendung sicherstellen (z.B. im Hinblick auf die Korrektheit von Datenflüssen).

## 3.2 Schemaevolution, Instanzmigration und CB-Migration

Nehmen wir an, dass der Prozessentwickler sich, wie in Abbildung 1 dargestellt, dazu entschieden hat die beiden Aktivitäten *X* und *Y* in ein Typ-Schema einzufügen. Durch den Einsatz von ADEPT können dann laufende Instanzen auf Wunsch des Prozessverantwortlichen noch auf die neue Schema-Version *S1'* migriert werden. Dabei muss sichergestellt werden, dass die migrierten Instanzen weiterhin korrekt ausgeführt werden, d.h. dass Verklemmungen oder fehlerhafte Datenflüsse ausgeschlossen werden. Dazu überprüft ADEPT zunächst, welche der laufenden Instanzen mit der neuen Schema-Version *verträglich* sind. Vereinfacht ausgedrückt ist dies der Fall, wenn die bisherige Ausführungshistorie der jeweiligen Instanz (d.h. das Logfile mit Einträgen zu Start- und Endereignissen von Instanz-Aktivitäten) auch auf dem neuen Schema erzeugbar ist. ADEPT stellt dies durch einfache und effizient überprüfbare Regeln (sog. *Compliance-Rules*) sicher, die auf einer wohldefinierten, formalen Basis fußen (für Details siehe [12]). Im Anschluss werden die verträglichen Instanzen migriert, während alle anderen Instanzen weiterhin gemäß ihres ursprünglichen Schemas ausgeführt werden. Ferner nimmt ADEPT bei der Migration von Instanzen auf das neue Schema die erforderlichen Zustandsanpassungen vor, sodass eine konsistente Fortführung der Ausführung gewährleistet ist. Anzumerken ist schließlich noch, dass Schemaevolution und Instanzmigrationen in ADEPT auch bei großer Instanzzahl (d.h. tausenden von Prozessinstanzen) effizient bewerkstelligt werden.

Die Änderung eines Prozess-Schemas macht in der Regel

nicht nur die Migration laufender Instanzen notwendig, sondern führt auch dazu, dass Teile der CB für Instanzen der neuen Schema-Version obsolet werden. Insbesondere kann eine Typ-Änderung dazu führen, dass Teile des bisher in der CB vorhandenen Wissens nun bereits in der neuen Schema-Version berücksichtigt sind. Deshalb muss für Instanzen der neuen Schema-Version auch eine neue Version der CB erstellt werden. Dabei werden nur diejenigen Fälle in die neue CB-Version migriert, die weiterhin relevant sind. Fälle, die von der Typ-Änderung betroffen sind, müssen entweder durch den Prozessverantwortlichen angepasst oder – falls sie nicht mehr gebraucht werden – von der Migration ausgeschlossen werden. Details zur Evolution von Prozess-Schemata und die damit verbundene Migration der CB finden sich in [10, 12].

## 4 Realisierung

Im Projekt *ProCycle* haben wir einen fortschrittlichen Prototyp entwickelt, der den beschriebenen Prozesslebenszyklus abbildet und die skizzierte Funktionalität bereitgestellt.

Das ProCycle-System besteht aus den zwei Komponenten ADEPT und CCBR-Tool. Die ADEPT-Technologie [4] unterstützt die Modellierung, Ausführung und das Monitoring von Prozessen. Darüber hinaus gestattet ADEPT Ad-hoc-Änderungen auf Instanzebene sowie Prozess-Änderungen auf Typ-Ebene mit der damit verbundenen Migration laufender Prozessinstanzen. CCBR-Tool erlaubt es, die Instanz-Änderungen mit Kontextinformation zu annotieren. Die resultierenden Fälle werden in der CB gespeichert und können später mittels CCBR-Tool wiederverwendet werden. Im Rahmen von ProCycle wurden ADEPT und das Open Source System *CCBR-Tool* (CCBR-Komponente von CBRFlow) zu einem integrierten System kombiniert [13].

Im Folgenden beschreiben wir wie ADEPT und CCBR-Tool zusammenwirken. Im Normalfall arbeitet der Benutzer mit der für ihn vom ADEPT-System generierten Arbeitsliste, d.h. er führt die dort aufgeführten Tätigkeiten aus. Wird es für eine bestimmte Prozessinstanz erforderlich, die geplante Reihenfolge von Aktivitäten zu ändern, neue Aktivitäten einzufügen oder Aktivitäten zu löschen, aktiviert der Benutzer die CBR-Komponente (vgl. Abb. 3).

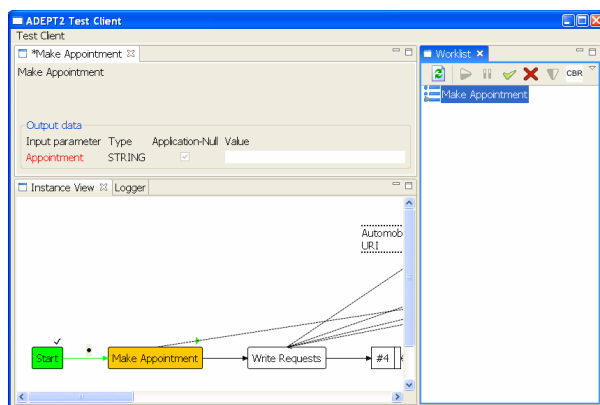


Abbildung 3: ADEPT-Worklist mit CBR-Aufruf

**Suchen von Fällen.** Nach dem Start von CCBR-Tool öffnet

sich die graphische Benutzeroberfläche zum Suchen von Fällen (siehe Abb. 4). In der oberen Bildschirmhälfte werden die Fragen angezeigt, die den Benutzer bei der Suche nach geeigneten Fällen unterstützen sollen. Beliebige Fragen können direkt über die jeweils nebenstehenden Auswahlboxen beantwortet werden. Damit einhergehend wird jeweils die angezeigte Liste der ähnlichen Fälle (untere Bildschirmhälfte) neu berechnet und sortiert; die relevantesten Fälle werden immer an oberster Stelle angezeigt.

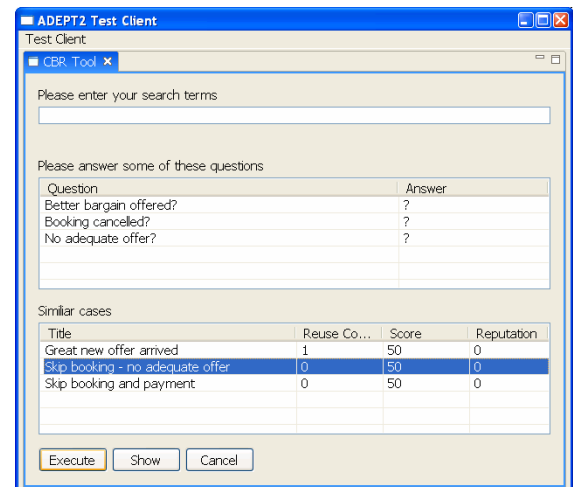


Abbildung 4: Suchen von Fällen

Will der Benutzer einen Fall wiederverwenden, selektiert er diesen und führt ihn aus. Dabei werden die Änderungsoperationen aus dem Lösungsteil des Falls an das ADEPT-System weitergeleitet und auf die betreffende Prozessinstanz angewendet. Zum Zwecke der Qualitätssicherung hat der Benutzer die Möglichkeit, den Fall später mittels Feedbackformular zu bewerten. Falls kein ähnlicher Fall gefunden wird, besteht die Möglichkeit den Suchdialog abzubrechen und einen neuen Fall anzulegen.

**Anlegen eines neuen Falls.** Die gewünschten Prozess-Änderungen werden zunächst im graphischen Editor von ADEPT vorgenommen. Die Änderungen werden dann an CCBR-Tool weitergeleitet, wo sie annotiert werden können (siehe Abb. 5). Dazu müssen eine Problembeschreibung erfasst und die Gründe für die Abweichung in Form von Frage-Antwort-Paaren angegeben werden. Der Fall wird dann in der CB gespeichert, bevor die Änderungsoperationen im ADEPT-System ausgeführt werden. Auch hier kann der Benutzer den Fall später mittels Feedbackformular bewerten.

## 5 Verwandte Ansätze

Dieser Beitrag hat aufgezeigt, wie durch Kombination von CBR mit adaptiver Prozess-Management-Technologie eine integrierte Unterstützung des Prozesslebenszyklus realisiert werden kann. Unser Anliegen ist es, aus vergangenen Prozessausführungen zu lernen und die gewonnenen Erfahrungen für Prozessverbesserungen zu nutzen.

Einen alternativen Ansatz zur Verbesserung der Qualität von Prozessen bieten *Process Mining* Techniken [14]. Mittels *Pro-*

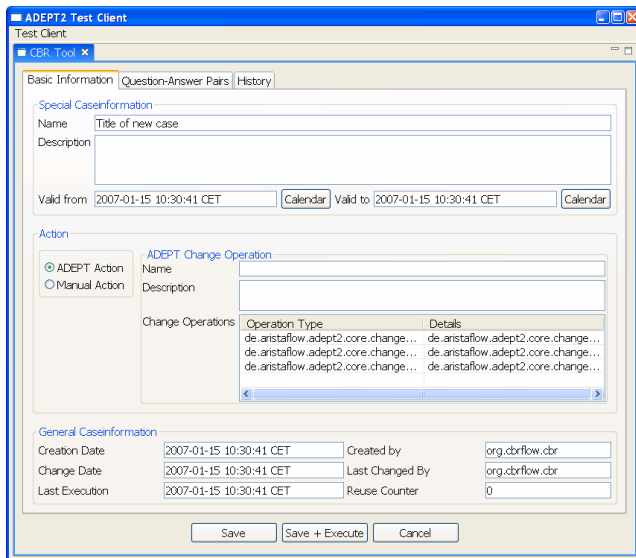


Abbildung 5: Anlegen neuer Fälle

cess Mining lassen sich Diskrepanzen zwischen modellierten Prozessen und tatsächlichen Prozessausführungen rasch erkennen. Bei der *Delta-Analyse* [15] etwa werden die modellierten Prozesse mit einem aus den Ausführungshistorien generierten Modell verglichen. Im Unterschied dazu vergleicht *Conformance Testing* [15] die Ausführungshistorie direkt mit einem gegebenen Prozess-Schema, d.h. es wird überprüft, ob eine vorliegende Prozessausführung konform zu einem gegebenen Prozess-Schema ist. Obwohl beide Techniken sich dazu eignen, Schwachstellen und Flaschenhalse aufzuzeigen, bleiben die Gründe für Abweichungen unklar. Wissen über diese ist allerdings unabdingbar, wenn das Prozessmodell geeignet verbessert werden soll.

Der Einsatz von CBR-Techniken zwecks verbesserter Prozessunterstützung ist nicht neu, allerdings fokussieren bisherige Ansätze auf einzelne Teilaspekte des Prozesslebenszyklus.

Ähnlich wie unser Ansatz verwendet URANOS [16] CBR in Verbindung mit der Anpassung von Prozessinstanzen. Um den Benutzer hierbei zu unterstützen, werden bereits erfolgte Anpassungen ähnlicher Instanzen (bzgl. Struktur und Kontext) präsentiert. Im Unterschied zu ProCycle verwendet URANOS allerdings klassisches CBR. Weiters liegt der Fokus in URANOS auf Änderungen einzelner Instanzen, Typ-Änderungen werden momentan nicht berücksichtigt.

CAKE [17] kombiniert Prozess-Management-Systeme, Agententechnologie und CBR-Techniken, um kurzlebige, zeitkritische Prozesse im Notfallbereich (z.B. Feuerwehr) zu unterstützen. CBR-Techniken werden dazu verwendet, Unterprozesse dynamisch auszuwählen und zuzuweisen. Im Unterschied zu CAKE ist unser Ansatz durch die Unterstützung von Schemaevolution und Instanzmigrationen auch für langlebige Prozesse geeignet.

CBR wird auch zur Unterstützung der Prozessmodellierung [18, 19, 20], der Konfiguration komplexer Prozesse [21] sowie der Behandlung von Ausnahmen [22] verwendet.

Planungsansätze aus der Künstlichen Intelligenz (KI), im Speziellen *Mixed Initiative Case-Based Planning* (z.B., NaCoDAE/HTN [23], MI-CBP [24], SiN [25] und HICAP [26]) sind komplementär zu unserem Ansatz. Unser Fokus liegt weniger

auf der Modellierung oder Planung einzelner Instanzen, sondern vielmehr auf der Ausführung einer großen Zahl von Instanzen zur Laufzeit. Prozess-Management-Ansätze basieren deshalb in der Regel auf einem vordefinierten Schema (d.h. Plan), das zur Laufzeit in hohen Zahlen instanziiert wird. Im Unterschied dazu werden bei der KI-Planung die Benutzer bei der Erstellung von Plänen unterstützt, wobei diese zumeist instanzspezifisch sind und somit die Änderung eines Plans keine weitere Änderung laufender Instanzen (desselben Plans) nach sich zieht. Im Unterschied dazu unterstützt unser Meta-Modell auch komplexe Kontrollflusselemente (z.B. bedingte Verzweigungen, Schleifen- und Fehlerrücksprünge und Synchronisation paralleler Pfade).

Der Bereich des prozessorientierten Wissensmanagements adressiert wissensintensive Prozesse. Ziel ist es, Benutzer bei der Ausführung von Aktivitäten zu unterstützen, indem ihnen für den aktuellen Prozesskontext relevantes Wissen zur Verfügung gestellt wird. FRODO TaskMan [27] etwa unterstützt die integrierte Modellierung und Ausführung von schwach strukturierter Prozesse. Wie in ProCycle erlaubt FRODO TaskMan Instanz-Änderungen zur Laufzeit, Typ-Änderungen werden hingegen nicht unterstützt. Modelle in FRODO müssen zum Zeitpunkt der Instanziiierung nicht vollständig sein, sondern können zur Laufzeit ergänzt bzw. verfeinert werden (*Late Modelling*).

## 6 Zusammenfassung und Diskussion

Aus der Integration von ADEPT und CBRFlow resultiert eine neue Generation adaptiver Prozess-Management-Technologie, welche die Implementierung neuer sowie die Adaption existierender Prozesse signifikant erleichtert und beschleunigt. Einerseits ermöglicht das realisierte ProCycle-System rasche Anpassungen von Prozessen und die Propagation von Typ-Änderungen auf laufende Instanzen. Andererseits erleichtert das System Benutzern die Durchführung von Instanz-Änderungen durch Verwendung von CBR-Techniken. Weiters werden die Prozessverantwortlichen bei der Ableitung von Prozessverbesserungen unterstützt.

Derzeit arbeiten wir in ProCycle an der weiteren Implementierung des integrierten Systems, welches die Methoden und Konzepte von ADEPT und CBRFlow kombiniert. Eine Evaluation ist geplant. Zukünftig soll der vorgestellte Ansatz dahingehend erweitert werden, dass PAIS agil entwickelt und verwaltet werden können. Insbesondere sollen die Aufwände, die mit Prozessänderungen verbunden sind, weiter reduziert werden.

## 7 Danksagung

Die Autoren bedanken sich beim Tiroler Wissenschaftsfond, der die Entwicklung des integrierten Prototyps mit finanziert hat. Der Dank gilt außerdem Ulrich Kreher, Markus Lauer, Jakob Pinggera und Stefan Zugal für die Diskussionen und die Mitarbeit an der Implementierung des integrierten Prototyps.

## Literatur

- [1] Reichert, M., Dadam, P.: ADEPTflex - Supporting Dynamic Changes of Workflows without Losing Control. *JIS* 10 (1998) 93-129.
- [2] Dumas, M., ter Hofstede, A., van der Aalst, W., eds.: *Process Aware Information Systems*. Wiley Publishing (2005).

- [3] Weske, M.: Workflow management systems: Formal Foundation, Conceptual Design, Implementation Aspects. Habilitationsschrift, Universität Münster (2000).
- [4] Weber, B., Wild, W., Breu, R.: CBRFlow: Enabling Adaptive Workflow Management through Conversational CBR. In: ECCBR'04 (2004) 434-448.
- [5] Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating Process Learning and Process Evolution - A Semantics Based Approach. In: BPM'05 (2005) 252-267.
- [6] Günther, C., Rinderle, S., Reichert, M., van der Aalst, W.: Change Mining in Adaptive Process Management Systems. In: CoopIS'06 (2006) 309-326.
- [7] Weber, B., Reichert, M., Rinderle, S., Wild, W.: Towards a Framework for the Agile Mining of Business Processes. In: BPM 2005 International Workshops (2006) 191-202.
- [8] Rinderle, S., Reichert, M., Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems - A Survey. Data and Knowledge Engineering 50 (2004) 9-34.
- [9] Aha, D.W., Muñoz-Avila, H.: Introduction: Interactive Case-Based Reasoning. Applied Intelligence 14 (2001) 7-8.
- [10] Weber, B., Reichert, M., Wild W.: Case-Base Maintenance for CCBR-based Process Evolution. In: ECCBR'06 (2006) 106-120.
- [11] Weber, B., Wild, W., Lauer, M., Reichert, M.: Improving Exception Handling by Discovering Change Dependencies in Adaptive Process Management Systems. In: Business Process Management Workshops 2006 (2006) 93-104.
- [12] Rinderle, S., Reichert, M., Dadam, P.: Flexible Support of Team Processes by Adaptive Workflow Systems. In: Distributed and Parallel Databases 1 (2004) 91-116.
- [13] <http://www.cbflow.org>.
- [14] <http://www.processmining.org>.
- [15] van der Aalst, W.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. In: Requirements Engineering Journal 10 (2005) 198-211.
- [16] Minor, M., Koldehoff, A., Schmalen, D., Bergmann, R.: Flexible Workflows for Digital Design in the Nano Era. In: WETICE'06 (2006) 273-278.
- [17] Freßmann, A.: Adaptive Workflow Support for Search Processes within Fire Service Organisations. In: WETICE'06 (2006) 291-296.
- [18] Kim, J., Suh, W., Lee, H.: Document-Based Workflow Modeling: A Case-Based Reasoning Approach. Expert Systems with Applications 23 (2002) 77-93.
- [19] Madhusudan, T., Zhao, J.: A Case-Based Framework for Workflow Model Management. In: BPM'03, Eindhoven (2003) 354-369.
- [20] Lu, R., Sadiq, S., Governatori, G.: Utilizing Successful Work Practice for Business Process Evol.. In BIS'06 (2006) 58-76.
- [21] Wargitsch, C.: Ein Beitrag zur Integration von Workflow- und Wissensmanagement unter besonderer Berücksichtigung komplexer Geschäftsprozesse. Dissertation, Universität Erlangen-Nürnberg (1998).
- [22] Luo, Z., Sheth, A., and J. Miller, K.K.: Exception Handling in Workflow Systems. Applied Intelligence 13 (2000) 125-147.
- [23] Muñoz-Avila, H., McFarlane, D., Aha, D., Ballas, J., Breslow, L., Nau, D.: Using Guidelines to Constrain Interactive Case-Based HTN Planning. In: ICCBR'99 (1999) 288-302.
- [24] Veloso, M., Mulvehill, A., Cox, M.: Rationale-Supported Mixed-Initiative Case-Based Planning. In: IAAI'97 (1997) 1072-1077.
- [25] Muñoz-Avila, H., Aha, D., Nau, D., Breslow, L., Weber, R., Yamal, F.: Sin: Integrating Case-Based Reasoning with Task Decomposition. In: Proc. IJCAI'01 (2001) 99-104.
- [26] Muñoz-Avila, H., Gupta, K., Aha, D., Nau, D.: Knowledge Based Project Planning. In: Knowledge Management and Organizational Memories. Kluwer Academic Publishers (2002)
- [27] Elst, L., Aschoff, F., Bernardi, A., Maus, H., Schwarz, S.: Weakly-Structured Workflows for Knowledge-Intensive Tasks: An Experimental Evaluation. In: WETICE'03 (2003) 340-345.

## Kontakt

Dr. Barbara Weber  
 Universität Innsbruck  
 Technikerstraße 21a, A-6020 Innsbruck  
 Tel.: +43 (0)512-507-6474  
 Fax: ++43 (0)512-507-9871  
 Email: Barbara.Weber@uibk.ac.at

Bild

**Barbara Weber** ist wissenschaftliche Mitarbeiterin am Institut für Informatik an der Uni Innsbruck. Ihre Dissertation hat sie zum Thema 'Integration of Workflow Management and Case-Based Reasoning' verfasst. Ihr primäres Forschungsinteresse gilt Methoden, Werkzeugen und Architekturen für das agile Management von Geschäftsprozessen. Sie ist Mitglied zahlreicher Programmkomitees und Organisator mehrerer internationaler Workshops.

Bild

**Werner Wild** ist seit mehr als 25 Jahren selbständiger IT Berater mit Schwerpunkt auf objektorientierten Software-Technologien und agilen Entwicklungsmethoden. Er hat Lehrverpflichtungen an den Universitäten Innsbruck und Bozen sowie am Management Center Innsbruck (FH) und beschäftigt sich in der Forschung mit Geschäftsprozessen, CBR und Agilen Methoden. Er leitet die Expertengruppe *Forschung und Innovation* der Wirtschaftskammer Tirol (UBIT).

Bild

**Manfred Reichert** ist seit 2005 Associate Professor am Lehrstuhl für Informationssysteme an der Uni Twente. Dort beschäftigt er sich intensiv mit Prozessmanagement- und Integrationstechnologien sowie deren Einbettung in Service-orientierte Architekturen. Daneben leitet er den Forschungsschwerpunkt E-health, der sich insbesondere mit der Unterstützung von Prozessen im Gesundheitswesen befasst. Zuvor war er Juniorprofessor an der Uni Ulm.

Bild

**Peter Dadam** ist seit 1990 Professor für Informatik an der Uni Ulm und leitet dort das Institut für Datenbanken und Informationssysteme. Seit mehreren Jahren beschäftigt er sich intensiv mit den technologischen Grundlagen von Prozess-Management-Systemen sowie deren Einsatz in fortschrittlichen Anwendungen und komplexen Arbeitsumgebungen. Davor leitete er am Wissenschaftlichen Zentrum der IBM in Heidelberg die Forschung über fortschrittliches Informationsmanagement.