



FernUniversität in Hagen
Fakultät für Mathematik und Informatik
Lehrgebiet Unternehmensweite Softwaresysteme
Prof. Dr. Lars Mönch

Untersuchung von Prozessmanagementsystemen am Beispiel von AristaFlow BPM Suite

Bachelorarbeit
zur Erlangung des akademischen Grades Bachelor of Science (B.Sc.) Wirtschaftsinformatik

an der Fakultät für Mathematik und Informatik
der FernUniversität in Hagen

Betreuer:	Prof. Dr. Lars Mönch
Bearbeiter:	Michael Theis
Bearbeitungszeit:	Drei Monate
Abgabetermin:	15. August 2012

Inhaltsverzeichnis

1 Einleitung	3
1.1 Gegenstand der Arbeit	3
1.2 Zielsetzung und Vorgehensweise	3
1.3 Aufbau der Arbeit.....	4
2 Prozessmanagementsysteme.....	5
2.1 Begriffsbildung und Aufgaben.....	5
2.2 Analyse von Anforderungen für das Erreichen von Prozessflexibilität	7
2.2.1 Flexibilität zur Entwurfszeit	8
2.2.2 Flexibilität zur Ausführungszeit	10
2.2.3 Prozessschemaevolution	12
2.3 Problemstellung und Diskussion von Vorarbeiten	15
3 Realisierung von Prozessflexibilität mit der AristaFlow BPM Suite	17
3.1 Einführung in die AristaFlow BPM Suite	17
3.2 Modellierung mit AristaFlow.....	19
3.2.1 Unterstützung von Ad-hoc-Anpassungen.....	23
3.2.2 Prozessschemaevolution in AristaFlow	26
3.3 Vergleich von AristaFlow mit anderen BPM-Werkzeugen	26
4 Umsetzung eines Beispielszenarios mit der AristaFlow BPM Suite.....	29
4.1 Szenario aus der Krankenhausdomäne	29
4.2 Umsetzung des Szenarios	34
4.3 Evaluierung der Möglichkeiten von AristaFlow	41
4.3.1 Prozessdurchlauf „Patientendurchsteuerung“	42
4.3.2 Erstes Ausnahmeszenario	46
4.3.3 Zweites Ausnahmeszenario	49
4.3.4 Prozessprotokollierung	52
5 Zusammenfassung und Ausblick.....	53
Abkürzungsverzeichnis.....	54
Abbildungsverzeichnis	55
Tabellenverzeichnis	57
Literaturverzeichnis.....	58

1 Einleitung

1.1 Gegenstand der Arbeit

Unternehmen sehen sich aufgrund einer verstärkten Globalisierung und durch mehr internationalen Wettbewerb neuen Herausforderungen ausgesetzt, um weiterhin am Markt bestehen zu können. Dazu können die Unternehmen z.B. ihre Durchlaufzeiten verkürzen, den Entscheidungsprozess beschleunigen oder einen schnelleren Informationsaustausch sicherstellen, um schneller auf Marktveränderungen reagieren zu können [AHW03].

Dies erfordert ein effizientes und effektives Management der betrieblichen Abläufe (Prozesse) in einem Unternehmen, d.h. Prozesse müssen sich schnell an neue Situationen anpassen lassen. Das führte laut [AHW03] in den Unternehmen dazu, dass die papiergestützten Prozesse mehr und mehr durch das papierlose elektronische Business-Process-Management (BPM) ersetzt wurden.

Prozessmanagementsysteme (PMS) unterstützen dabei das BPM, u.a. durch aktive Bereitstellung und Weiterleitung von Aufgaben in die Arbeitslisten der zuständigen Bearbeiter, durch Automatisierung von Teilprozessen, durch Integration heterogener Anwendungssysteme, durch schnellere Anpassbarkeit der Prozesse an neue Gegebenheiten und bessere Nachvollziehbarkeit einzelner Prozessausführungen [DRR11].

Anhand eines Beispielszenarios wird mit der AristaFlow BPM Suite gezeigt, wie man mit Hilfe eines PMS ein prozessorientiertes klinisches Informationssystem umsetzen kann.

1.2 Zielsetzung und Vorgehensweise

Das Ziel dieser Bachelorarbeit ist die Analyse der funktionalen Anforderungen an PMS sowie die Analyse der technologischen Anforderungen in Hinblick auf die Flexibilität von Prozessen.

Aktuelle Prozess- und Workflow-Management-Systeme unterstützen überwiegend nur Arbeitsabläufe, die sich vollständig, d.h. inklusive ihrer Ausnahmen, vormodellieren lassen. Abweichungen zur Laufzeit sind nur sehr stark eingeschränkt möglich. Komplexere Prozesse mit flexiblen Reaktionen auf unvorhergesehene Ausnahmesituationen sind damit nicht modellierbar.

Im weiteren Verlauf der Arbeit werden die technologischen Anforderungen beschrieben und analysiert, die für die benötigte Flexibilität von anspruchsvollen und komplexen Arbeitsabläufen zur Laufzeit erfüllt sein müssen.

Die AristaFlow BPM Suite wird bzgl. der Anforderungen an Flexibilität zur Entwurfs- und Ausführungszeit sowie bzgl. der Möglichkeiten zur Migration laufender Prozesse mit anderen Werkzeugen für das BPM verglichen.

Abschließend werden mit Hilfe der AristaFlow BPM Suite ein Beispielszenario aus der Krankenhausdomäne umgesetzt und die Möglichkeiten von AristaFlow evaluiert.

1.3 Aufbau der Arbeit

In Kapitel 2 folgen die Begriffsbildung und die Aufgaben von PMS. Es werden die Anforderungen, die für das Erreichen von Prozessflexibilität erfüllt sein müssen, analysiert sowie die Problemstellung und die Vorarbeiten diskutiert.

Das anschließende Kapitel 3 zeigt, wie Prozessflexibilität mit der AristaFlow BPM Suite realisiert wurde. Dazu wird eine Einführung in die AristaFlow BPM Suite gegeben und die Modellierung mit AristaFlow dargestellt. Das Kapitel schließt mit einem Vergleich von AristaFlow mit anderen BPM-Werkzeugen ab.

Die Umsetzung eines Beispielszenarios aus der Krankenhausdomäne mit der AristaFlow BPM Suite ist Gegenstand von Kapitel 4. Dort wird auch auf die Evaluierung der Möglichkeiten von AristaFlow eingegangen.

Den Abschluss dieser Arbeit bildet Kapitel 5 mit einer Zusammenfassung und einem Ausblick in Hinblick auf flexible Prozessunterstützung.

2 Prozessmanagementsysteme

2.1 Begriffsbildung und Aufgaben

In der Fachliteratur treten oftmals im Zusammenhang mit dem Begriff PMS weitere Termini, wie z.B. BPM, Workflow-Management-System (WFMS) oder prozessorientiertes Informations- bzw. Anwendungssystem, auf. Während Business-Process-Management-System (BPMS) die englische Entsprechung für (Geschäfts-)PMS ist, ist der Übergang im Falle der anderen vorgenannten Begriffe teilweise fließend und findet daher oft eine synonyme Verwendung in der Literatur.

Das BPM beinhaltet nach [AHW03] Methoden, Techniken und Werkzeuge, um den Entwurf, die Ausführung, die Verwaltung und die Analyse der operativen Geschäftsprozesse zu unterstützen und stellt eine Erweiterung der klassischen WFMS dar, zumal auch Menschen, Organisationen, Anwendungen, Dokumente und andere Informationsquellen mit einbezogen werden.

WFMS unterstützen nach [HaNe09] die Geschäftsprozessabwicklung durch die automatische Weiterleitung von Dokumenten, Informationen oder Aufgaben zu den jeweiligen Bearbeitern sowie durch die Überwachung von Fristen. Dies geschieht über vordefinierte Regeln in einem Prozessmodell.

Bei den WFMS unterscheidet [Mert09] nach „Production/Transactional Workflows“ für streng standardisierte Prozesse wie z.B. Geschäftsreise- oder Beschaffungsanträge, nach „Administrativen Workflows“ für die teilstrukturierten Vorgänge wie z.B. die Auftragsabwicklung oder die Reklamationsbearbeitung und nach „Ad-hoc-Workflows“ für die freien, ungeplanten Vorgänge.

Ein BPMS bzw. PMS wird definiert als ein generisches Softwaresystem, das durch detaillierte Prozessentwürfe gesteuert wird, um operative Geschäftsprozesse auszuführen und zu verwalten [AHW03]. Dabei stellt das PMS ein (anwendungs-)neutrales Basissystem für ein prozessorientiertes Informationssystem dar, das zusätzlich anwendungsbezogene Daten und Funktionen umfasst [DRR04].

Anhand des BPM-Lebenszyklus, der die verschiedenen Phasen zur Unterstützung von operativen Geschäftsprozessen beschreibt, nehmen [AHW03] eine Abgrenzung von WFM zu BPM vor. In der Prozessentwurfsphase werden die Prozesse entworfen. Anschließend werden durch Konfiguration eines prozessorientierten Informationssystems die Entwürfe implementiert. Danach beginnt mit der Abarbeitung der operativen Geschäftsprozesse in dem konfigurierten System die Prozessausführungsphase. In der Diagnosephase werden die operativen Abläufe auf Fehler analysiert und Verbesserungsmöglichkeiten gesucht. Wie in Abbildung 1 dargestellt, konzentriert sich das (traditionelle) WFM auf die untere Hälfte des BPM-Lebenszyklus, wohingegen das BPM auch Unterstützung bei der Analyse und der Verbesserung von Prozessen bietet.

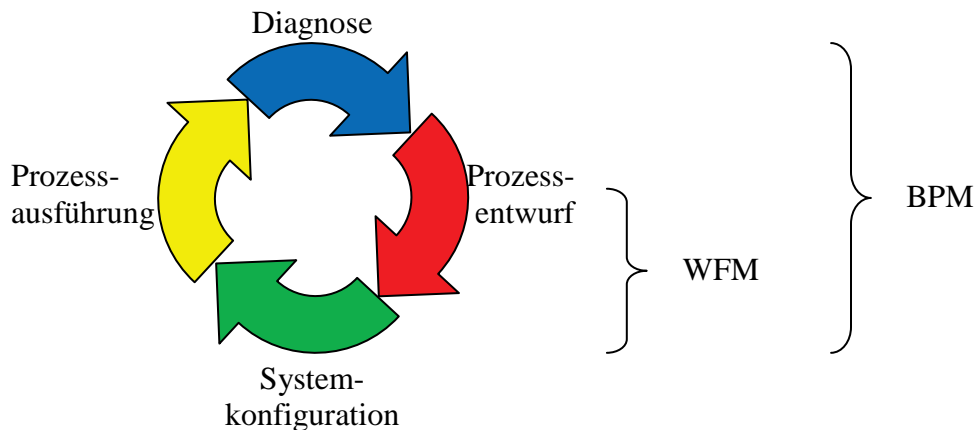


Abbildung 1: BPM-Lebenszyklus zum Vergleich von WFM und BPM nach [AHW03]

Die Aufgaben eines PMS sehen [LRR04] in der Steuerung, Verwaltung und Überwachung der betrieblichen Unternehmensabläufe und [DRR11] in der Unterstützung des BPM durch aktive Bereitstellung und Weiterleitung von Aufgaben in die Arbeitslisten der zuständigen Bearbeiter, durch Automatisierung von Teilprozessen, durch Integration heterogener Anwendungssysteme, durch schnellere Anpassbarkeit der Prozesse an neue Gegebenheiten und bessere Nachvollziehbarkeit einzelner Prozessausführungen.

Dazu müssen PMS über durchdachte Werkzeuge für die Prozessmodellierung, -überprüfung und -tests, für die Überwachung sowie für das dynamische Anpassen von laufenden Prozessinstanzen oder des Arbeitsvorrats verfügen [Reic08].

Im Rahmen der Integrierten Informationssysteme lassen sich PMS auf der Ebene der operativen Systeme am ehesten den Administrationssystemen zuordnen, deren Ziele – ähnlich der PMS – in der Rationalisierung der Massendatenverarbeitung durch Kostensenkung und Entlastung des Personals von Routineaufgaben sowie in der Prozessbeschleunigung durch Verkürzung der Durchlaufzeit liegen [Mert09].

Ein Grundprinzip eines PMS liegt in der Trennung von Prozesslogik und Anwendungscode, d.h. dass die Kontroll- und Datenflüsse sowie die organisatorischen Regelungen der unterstützten Prozesse explizit im PMS modelliert und nicht im Programmcode versteckt werden [Reic00, Rind04]. Dies wird in Abbildung 2 verdeutlicht.

Die Frage nach dem Nutzen von PMS beantworten [WaPa07] u.a. mit der Verbesserung der Ablauf- und Kostentransparenz, der klaren Zuordnung von Kompetenzen und Verantwortlichkeiten, der Produktivitätssteigerung durch kontinuierliche Prozessverbesserung sowie der Mess- und Überwachbarkeit der Prozessleistung.

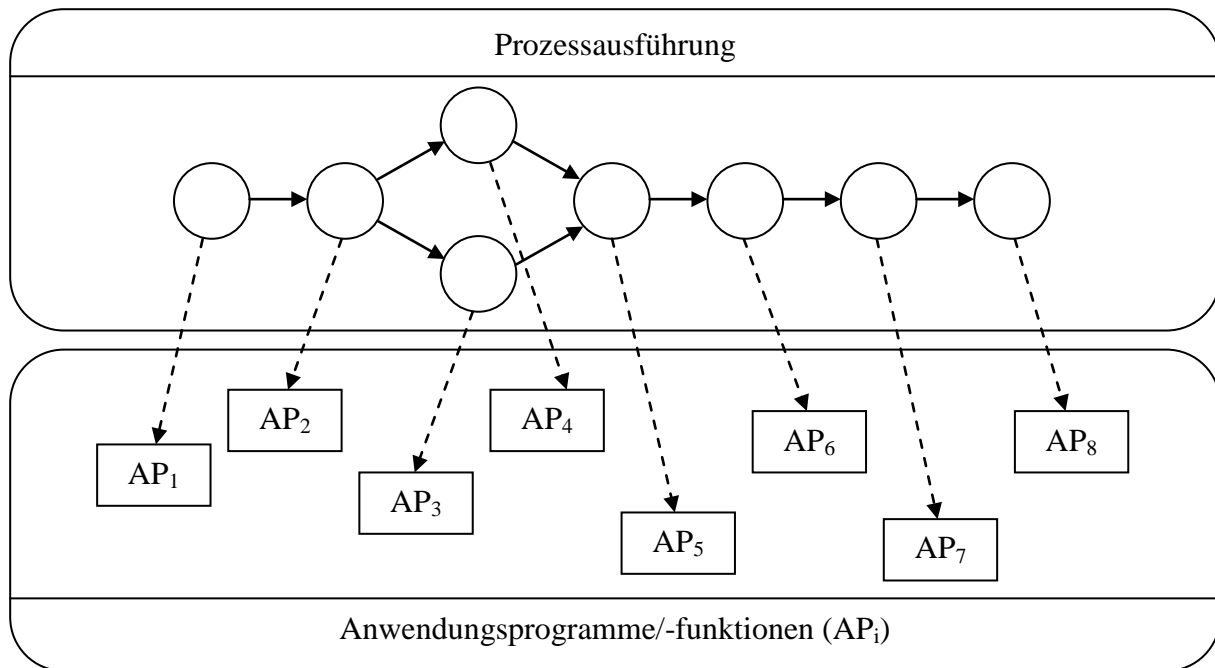


Abbildung 2: Trennung von Prozesslogik und Anwendungscode in Anlehnung an [Rind04]

2.2 Analyse von Anforderungen für das Erreichen von Prozessflexibilität

An ein PMS werden funktionale und technologische Anforderungen gestellt. Die funktionalen Anforderungen reichen von der einfachen Anzeige und Manipulation von Datenbanktabellen bei formularorientierten PMS über die Anzeige und Bearbeitung von Dokumenten bei den dokumentenorientierten PMS bis hin zur Verknüpfung von beliebig komplexen Anwendungsfunktionen bzw. Diensten mit den einzelnen Prozessschritten eines Geschäftsprozesses. Dies können z.B. der Empfang oder Versand von E-Mails, der Aufruf externer Programme oder die Nutzung von Webservices bzw. Datenbankoperationen sein. Dabei muss das PMS auch mit Aufruf- und Resultatparametern, die wiederum für nachfolgende Prozessschritte oder den Kontrollfluss benötigt werden, umgehen können [DRR11].

Einfache Geschäftsprozesse, wie z.B. Reisekostenabrechnung oder Urlaubsbeantragung, lassen sich gut zur Entwurfszeit modellieren, da hier klare Strukturen existieren und vorhersehbare (Ausnahme-)Situationen entstehen. Bei komplexer werdenden Prozessen kann eine vollständige Modellierung mit allen denkbaren Ausnahmesituationen und Sonderfällen zur Entwurfszeit nicht mehr gewährleistet werden. Dies stellt ein PMS vor die Aufgabe, auch in diesen Ausnahmesituationen – unter Einhaltung der Nachvollziehbarkeit und Fehlervermeidung – flexibel zu bleiben, damit das PMS nicht umgangen werden muss.

Dazu sind bestimmte technologische Anforderungen hinsichtlich Prozessflexibilität zu erfüllen, die – wie in Abbildung 3 dargestellt – in Flexibilität zur Entwurfs- bzw. Ausführungszeit und in die Prozessschemaevolution unterschieden werden. Die nächsten drei Unterabschnitte beschäftigen sich mit den einzelnen Flexibilitätsaspekten.



Abbildung 3: Flexibilitätsaspekte

2.2.1 Flexibilität zur Entwurfszeit

Bei der Flexibilität zur Entwurfszeit geht es um die Frage, wie schnell man neue Prozesse bzw. Prozessmodelle implementieren und anschließend in den produktiven Betrieb bringen kann. Dafür muss ein stabiler und robuster Prozessablauf gewährleistet werden. Stabil bedeutet im Allgemeinen, dass der Prozess wiederholbare und vorhersagbare Ergebnisse liefert [DiSc05, Stro09, Cim12]. Ein Prozess ist indes robust, wenn er auch gegenüber externen Störeinflüssen und Abweichungen unempfindlich ist [Gab12, Klep08, WiWa11]. In [Jen02] wird die Robustheit weiter gefasst als die Stabilität. So wird die Lösung eines dynamischen Systems als stabil bezeichnet, wenn geringfügige Störungen an der Lösung zu einer neuen Lösung führen, die nahe an der ursprünglichen bleibt. Der qualitative Aspekt der Lösung spielt dabei keine große Rolle. Robustheit dagegen spiegelt die Leistungsfähigkeit eines Systems bzw. Prozessablaufs wider, strukturellen Störeinflüssen zu widerstehen, ohne dass sich die Funktionalität oder Lösungsqualität wie z.B. die Ausführungsdauer signifikant ändern.

Moderne PMS unterstützen die Prozessmodellerstellung mit Hilfe grafischer Modellierungswerkzeuge, um neue Prozesse zu beschreiben. Dazu können die verfügbaren Anwendungsfunktionen einfach per Drag & Drop oder per Kontextmenü in die Prozessmodelle eingefügt werden. Dabei muss idealerweise durch das PMS sichergestellt werden, dass der neue bzw. angepasste Prozess technisch robust und stabil läuft, so dass keine Laufzeitfehler auftreten können. Viele PMS erfordern vom Prozessmodellierer intensiven Testaufwand, um die verschiedenen Prozess- und Ausnahmesituationen vollständig zu durchlaufen. Wünschenswert ist eine systemseitige, d.h. durch das PMS unterstützte, Fehleranalyse des Prozessmodells wie z.B. die Erkennung und Vermeidung von Modellierungsfehlern in einfachen, aber auch komplexen Prozessmodellen, die

möglicherweise zu Laufzeitfehlern oder unerwünschten Blockierungen (Deadlocks) führen. So ist z.B. in Abbildung 4 aufgrund eines Modellierungsfehlers der Kontrollfluss unterbrochen. Dies kann von einem PMS, sofern es mehrere Startknoten zulässt, fälschlicherweise als eine parallele Ausführung der beiden Prozessschritte A und C interpretiert werden, was zu unerwünschten Zuständen führen kann.

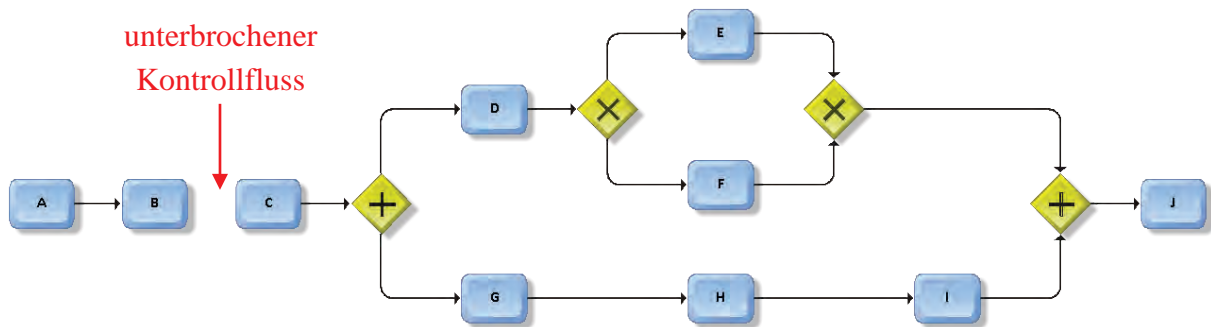


Abbildung 4: Modellierungsfehler – unterbrochener Kontrollfluss

In einem zweiten Beispiel (Abbildung 5) ist eine AND-Verzweigung mit einem XOR-Konnektor abgeschlossen, so dass der Prozessablauf fortgeführt werden kann, ohne dass beide Pfade komplett durchlaufen wurden und der Prozessschritt J eventuell nicht mit allen benötigten Informationen aus den vorherigen Prozessschritten versorgt wird. Dies kann dann zu einem ungewollten und fehlerhaften Prozessverhalten, aber auch zu einem Laufzeitfehler der Prozessinstanz führen.

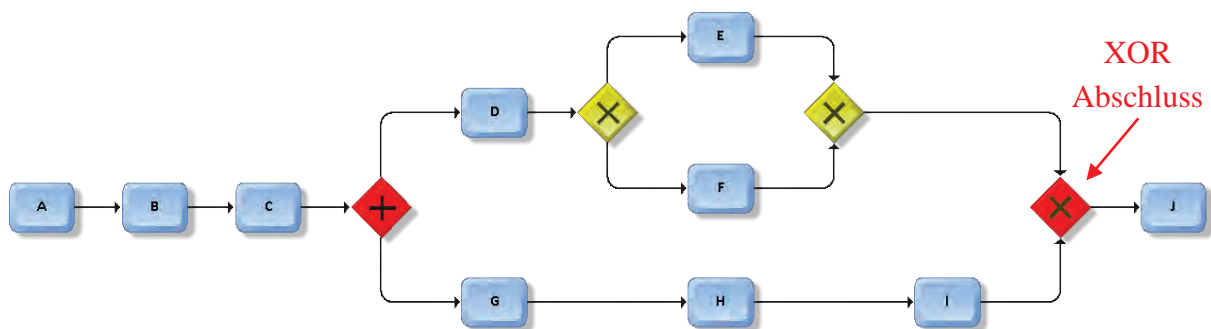


Abbildung 5: Modellierungsfehler – AND mit XOR abgeschlossen

Ein weiterer Aspekt ist die Kenntnis des PMS über die Datenflüsse zwischen den Anwendungsfunktionen, um die korrekte und vollständige Versorgung der Prozessschritte mit den benötigten Daten gewährleisten zu können. Fehlen diese Informationen, so kann das PMS zur Entwurfszeit keine Prüfung der benötigten Eingabeparameter sowie Rückgabewerte vornehmen, so dass Anwendungsfunktionen aufgerufen werden können, für die die erforderlichen Eingabedaten nicht zur Verfügung stehen und auf diese Weise mit einem Laufzeitfehler enden [DRR11].

Abbildung 6 verdeutlicht die Datenflüsse zwischen den Anwendungsfunktionen bzw. Services (S_1 bis S_{10}).

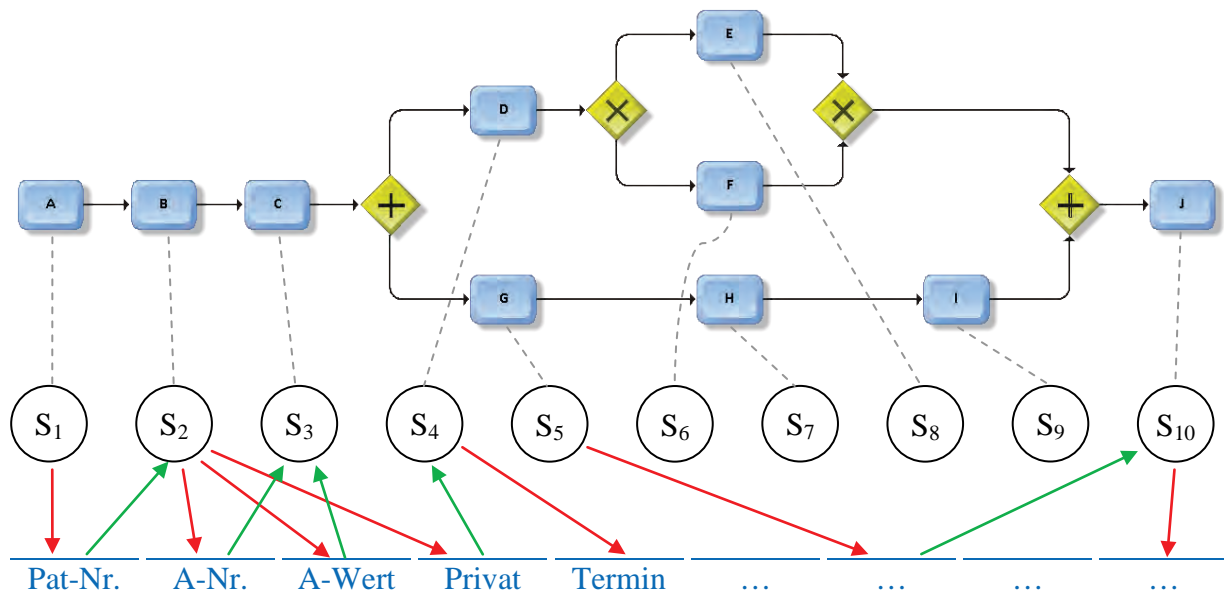


Abbildung 6: Datenflüsse zwischen Anwendungsfunktionen

2.2.2 Flexibilität zur Ausführungszeit

Die Flexibilität zur Ausführungszeit behandelt die Problemstellung, wie unvorhergesehene, d.h. nicht vormodierte, Situationen während der Prozessausführung gehandhabt werden. Dies beinhaltet auf Prozessinstanzebene die Möglichkeit, bei Bedarf individuell Prozessschritte ad hoc einzufügen, zu ändern, zu löschen oder zu verschieben, ohne dass die Robustheit und Stabilität der Prozessinstanz negativ beeinträchtigt wird. Fehlt diese Möglichkeit, so muss – unabhängig von der eingetretenen Situation – der vorgegebene Prozessablauf durchlaufen werden oder das PMS, sofern möglich, umgangen werden, indem manuell in die internen Prozesszustandsdaten eingegriffen wird.

Die Notwendigkeit von Flexibilität zur Ausführungszeit ergibt sich bereits aus bestimmten Geschäftsprozessen, die von Natur aus in ihrer Gesamtheit nicht vollständig vorhersehbar und planbar sind. So kann für Behandlungsprozesse in der Krankenhausdomäne kein einheitlicher Ablauf festgelegt werden, sondern hier muss bei Bedarf der Ablauf individuell auf den Patienten angepasst werden können. Dies kann z.B. in unbekanntem Allergien auf bestimmte Medikamente, in aufgetretenen Komplikationen oder in unzutreffenden Diagnosen begründet sein. Eine solche (Ad-hoc-)Abweichung vom geplanten Prozessablauf muss, insbesondere in den zuvor genannten Fällen, schnell und ohne Programmieraufwand umsetzbar sowie im Nachhinein nachvollziehbar sein. Dabei muss sich die Prozessinstanz auch weiterhin robust und stabil verhalten, was eine umfassende, systemseitige Korrektheitsprüfung erforderlich macht. Dazu gehört auch, dass nur berechtigte und qualifizierte Benutzer eine derartige Anpassung der Prozessinstanz vornehmen dürfen. Den Prozessspezialisten müssen hierbei mehr Anpassungsmöglichkeiten angeboten werden wie für einfach qualifizierte Anwender. Dies muss vom PMS über ein entsprechendes Berechtigungskonzept sichergestellt werden [DRR11].

In Abbildung 7 ist beispielhaft der Prozessschritt „Kontrastmittel applizieren“ als Ad-hoc-Anpassung einer laufenden Prozessinstanz des Prozessschemas S dargestellt. Die Legende der verwendeten Symbole ist in folgender Tabelle 1 zu finden. Diese gilt auch für die nachfolgenden Abbildungen.

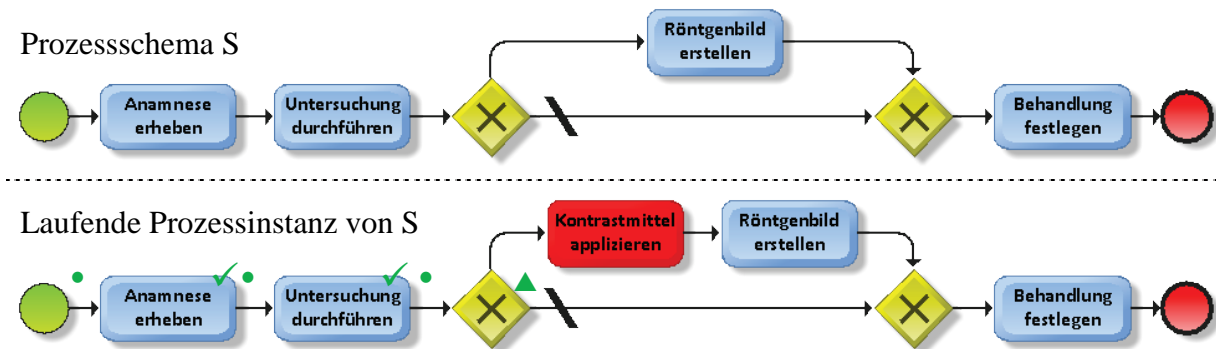


Abbildung 7: Ad-hoc-Anpassung einer laufenden Prozessinstanz

Tabelle 1: Legende der Symbole

Symbol	Beschreibung
✓	Prozessschritt ausgeführt
✗	Prozessschritt übersprungen
▲	Prozessschritt aktiviert
▶	Prozessschritt wird gerade ausgeführt
•	Kante ist wahr
●	Kante ist falsch

Derzeit bieten nur einige PMS rudimentäre Möglichkeiten für Ad-hoc-Abweichungen, wie z.B. das einfache Ersetzen von Prozessschritten durch andere im Falle einer kompatiblen Schnittstelle (analog dem Late Binding) oder das Einfügen eines Prozessschrittes an der aktuellen Position, an. Erschwerend kommt hinzu, dass viele PMS die Datenflüsse zwischen den Anwendungsfunktionen nicht kennen und so die Robustheit des Prozessablaufs selbst bei einfachen Anpassungen auf Prozessinstanzebene gefährden. Daher sollte dies nur von Prozessspezialisten durchgeführt werden. Komplexere Änderungen werden indes von den heutigen PMS nicht unterstützt.

Dabei sind das Anwendungspotenzial und die Einsatzmöglichkeiten für PMS, die weitergehende Flexibilität zur Ausführungszeit anbieten können, enorm. So kann z.B. ein PMS, das an eine Wissensbasis gekoppelt ist, auf Basis früherer Entscheidungen und Anpassungen flexibel Änderungen an einer Prozessinstanz vornehmen. Auch können Prozesse schneller in Betrieb genommen werden, da eventuell existierende Schwächen im Prozessmodell noch nachträglich durch einfache Anpassungen der laufenden

Prozessinstanzen ausgebessert werden können. Auf diese Weise können auch lang laufende Prozesse, die noch nicht bis zum Ende ausmodelliert worden sind, gestartet werden und sukzessive um weitere Prozessschritte erweitert werden [DRR11].

2.2.3 Prozessschemaevolution

Bei der Prozessschemaevolution geht es um die Fähigkeit eines PMS, bereits laufende Prozessinstanzen auf ein geändertes Prozessmodell bzw. -schema migrieren zu können [DRR11].

Im einfachsten Fall laufen im Rahmen einer Prozessschemaänderung bereits aktive Prozessinstanzen noch nach dem alten Prozessschema weiter und neu gestartete Prozesse nach dem neuen Schema. Schwieriger wird es, wenn z.B. neue gesetzliche Bestimmungen, andere geänderte Rahmenbedingungen oder ein fehlerhaftes Prozessmodell eine Anpassung auch an den laufenden Prozessinstanzen notwendig machen. Bei einigen wenigen aktiven Prozessinstanzen können PMS, die die Flexibilität zur Ausführungszeit unterstützen, diese Anpassung auf Prozessinstanzebene durchführen. Jedoch steigen der Aufwand und die Fehleranfälligkeit mit der Anzahl der laufenden Prozessinstanzen. Dies wird mit Hilfe der Prozessschemaevolution gelöst. Hierbei werden verzerrte und unverzerrte Prozessinstanzen unterschieden [Nahl05, Rind04]. Unverzerrte Prozessinstanzen entsprechen ihrem Prozessschema, d.h. sie sind nicht angepasst worden. Verzerrte Prozessinstanzen hingegen weichen aufgrund einer Ad-hoc-Änderung von ihrem Prozessschema ab. In diesem Fall muss das PMS prüfen, inwiefern eine Migration der einzelnen verzerrten Prozessinstanzen auf das neue Prozessschema sinnvoll bzw. überhaupt möglich ist, ohne die Stabilität und die Korrektheit des Prozessablaufs zu beeinträchtigen.

Für eine gewünschte Migration von Prozessinstanzen auf ein neues Schema spielen der Prozessfortschritt der laufenden Instanzen und die evtl. durchgeführten Ad-hoc-Abweichungen für einzelne Prozessinstanzen eine wesentliche Rolle. In diesem Zusammenhang werden drei Fälle für die Migration betrachtet, die in den folgenden Beispielen verdeutlicht werden. Es werden unverzerrte sowie verzerrte Prozessinstanzen mit und ohne überlappende Änderungen unterschieden.

Ausgangsbasis bildet in Abbildung 8 das Prozessschema S, das durch Einfügen des zusätzlichen Prozessschritts „Tubuslage prüfen“ nach der Aktivität „Narkose einleiten“ zum Prozessschema S' modifiziert wird.

Abbildung 9 zeigt zwei Beispiele von unverzerrten Prozessinstanzen. Im Fall der Prozessinstanz I₁, die sich bereits im Schritt „OP durchführen“ befindet, kann keine verträgliche Migration auf das neue Prozessschema S' erfolgen, da die Instanz zu weit fortgeschritten ist. Sie bleibt daher unverändert. Dagegen kann die Prozessinstanz I₂ gemäß Schema S' verträglich auf I₂' migriert werden.

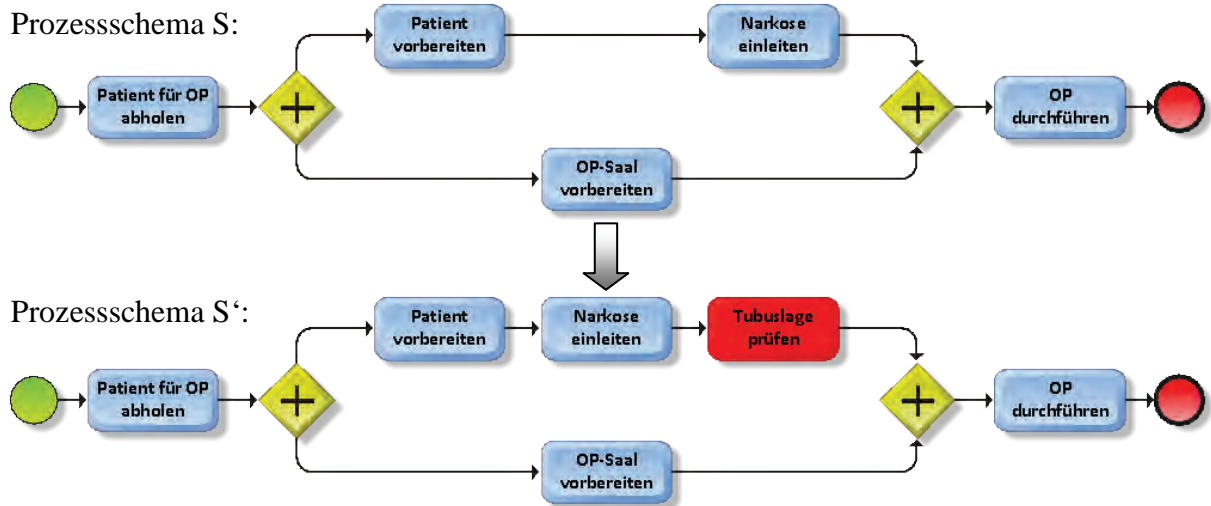
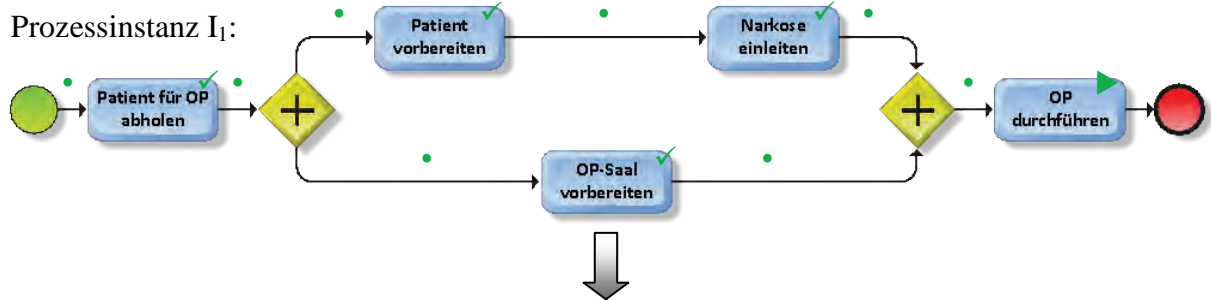
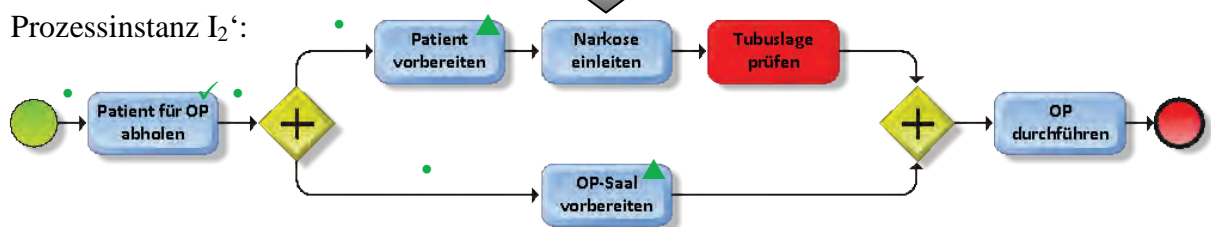
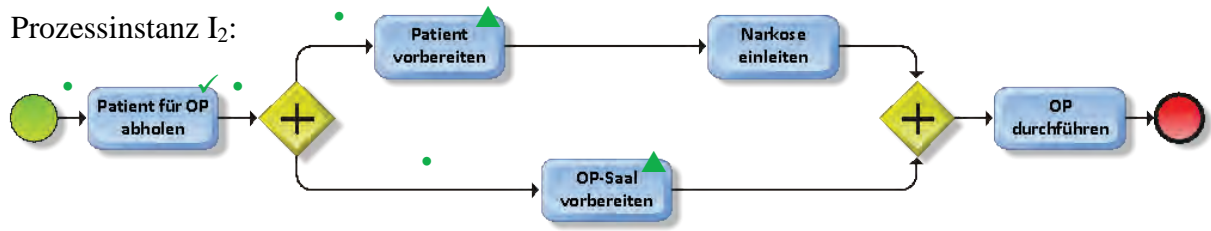


Abbildung 8: Modifikation von Prozessschema S zu S'



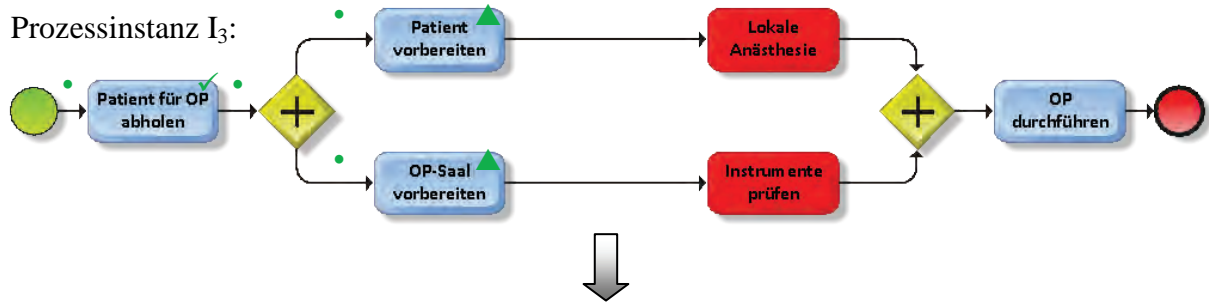
Migration ist nicht verträglich. Prozessinstanz I₁ bleibt auf Prozessschema S.



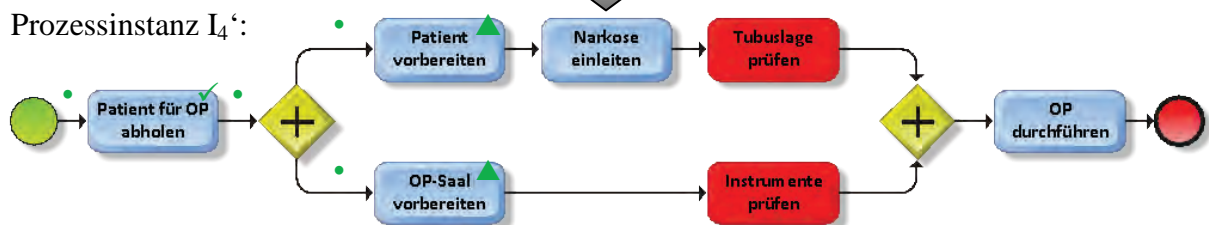
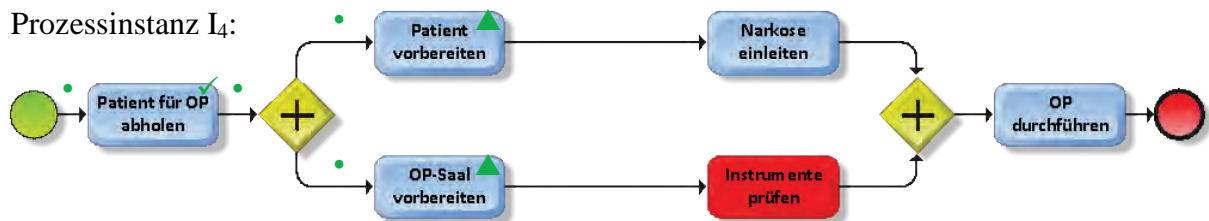
Migration ist verträglich. Instanzänderung wird propagiert.

Abbildung 9: Beispiele von unverzerrten Prozessinstanzen

Im nächsten Fall werden in Abbildung 10 zwei verzerrte Prozessinstanzen, die ad hoc angepasst sind, in Bezug auf ihre Migrationsfähigkeit betrachtet. In Prozessinstanz I_3 sind der Prozessschritt „Narkose einleiten“ durch „Lokale Anästhesie“ ersetzt und der Schritt „Instrumente prüfen“ eingefügt worden. Durch den ersetzten Prozessschritt kann eine verträgliche Migration nicht durchgeführt werden. I_3 verbleibt auf Schema S. Die Instanz I_4 ist um eine neue Aktivität „Instrumente prüfen“ erweitert worden. Eine verträgliche Migration kann für I_4 auf I_4' angewandt werden.



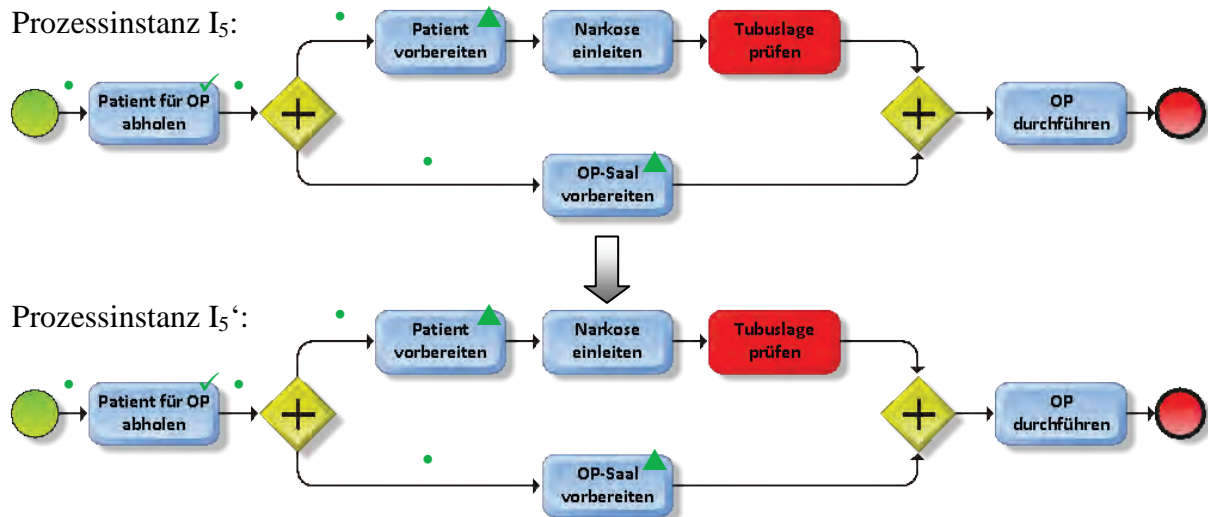
Migration ist nicht verträglich. Prozessinstanz I_3 bleibt auf Prozessschema S.



Migration ist verträglich. Instanzänderung wird propagiert.

Abbildung 10: Beispiele von verzerrten Prozessinstanzen

Eine weitere Variante für verzerrte Prozessinstanzen sind Instanzen mit überlappenden Änderungen. Abbildung 11 zeigt eine derartig ad hoc angepasste Prozessinstanz, die um die Aktivität „Tubuslage prüfen“ bereits erweitert wurde. Eine Migration auf Prozessinstanz I_5' kann verträglich durchgeführt werden.



Migration ist verträglich. Instanzänderung auf I₅' wird propagiert.

Abbildung 11: Beispiel einer verzerrten Prozessinstanz mit Überlappung

2.3 Problemstellung und Diskussion von Vorarbeiten

In der vorliegenden Arbeit wird untersucht, wie Prozessflexibilität mit der AristaFlow BPM Suite und der zugrundeliegenden ADEPT-Technologie unterstützt wird. Dazu soll ein Beispielszenario aus der Krankenhausdomäne mit der AristaFlow BPM Suite implementiert und in verschiedenen Situationen ausprobiert werden. Weiterhin sind die Möglichkeiten der AristaFlow BPM Suite mit anderen BPM-Werkzeugen zu vergleichen.

In diversen Vorarbeiten wurden bereits die Geschichte der Entwicklung und die Herausforderungen von AristaFlow und der ADEPT-Technologie beschrieben [Dada06, Dada09, DaRe09]. Die Entwicklung eines formalen Workflow-Metamodells für die Modellierung, Ausführung und die flexible Anpassung von Workflows wird in der Dissertation von [Reic00] dargestellt. Ergebnisse dieser Dissertation sind das ADEPT-Basismodell und darauf aufbauend der ADEPT_{flex}-Ansatz, der die Grundlage für das Prinzip Correctness-by-Construction in AristaFlow bildet. Correctness-by-Construction bedeutet, dass zu jeder Zeit nur die Änderungsoperationen auf ein Prozessmodell angewandt werden können, die ein strukturell konsistentes Schema in ein anderes strukturell konsistentes Schema überführen.

Die semantische Korrektheit eines Prozesses in adaptiven PMS bei der Durchführung verschiedener Änderungsoperationen sowie die Besonderheiten bei Prozesstyp- bzw. Prozessinstanz-Änderungen unter Berücksichtigung von Effizienzgesichtspunkten wurden in den Diplomarbeiten von [Nahl05] und [Zhou06] ausgearbeitet. In [KRRD09] werden Speicherkonzepte für die effiziente Speicherrepräsentation von Schema- und Instanzdaten behandelt, die im ADEPT2-PMS realisiert sind. Änderungen des Prozessschemas in PMS haben nicht nur Einfluss auf den Kontrollfluss, sondern wirken sich auch auf den Datenfluss aus. [Rind09] untersucht, wie die Korrektheit des Datenflusses zur Entwicklungs- und Laufzeit sowie nach Änderungen des Kontrollflusses effizient sichergestellt wird.

[RRD04, RRD04a] beschreiben allgemeine Korrektheitskriterien, um die Konformität von laufenden Prozessinstanzen gegen ein angepasstes Prozessschema sicherzustellen, so dass diese Änderungen korrekt, d.h. ohne Inkonsistenzen und Fehler, auf die Instanzen übertragen werden können.

Die Prozessschemaevolution in PMS ist Gegenstand der Dissertation von [Rind04] mit dem Ziel, ein umfassendes formales Rahmenwerk für die Unterstützung von Prozesstyp- und Prozessinstanz-Änderungen zu liefern, wobei die korrekte und performante sowie benutzerfreundliche Migration von laufenden Prozessinstanzen auf einen geänderten Prozesstyp im Vordergrund steht. Die darin dargestellten Grundlagen und Strategien für die Migration von unverzerrten und verzerrten Prozessinstanzen sind in die Entwicklung des ADEPT2-PMS eingeflossen.

Die Probleme und speziellen Anforderungen eines prozessorientiertes Informationssystems im klinischen Umfeld sowie die Lösungsansätze aus dem ADEPT-Projekt werden in [DRK00] ausgeführt. In der Dissertation von [Reut11] wird untersucht, wie WFMS die Einhaltung klinischer Pfade unterstützen. Dabei wird auch die AristaFlow BPM Suite auf Basis von ADEPT2 betrachtet.

In der Bachelorarbeit von [Bach10] wurde der Prozess für die Reisekostenabrechnung der Universität Ulm untersucht und mit Hilfe der AristaFlow BPM Suite implementiert. Die vorliegende Arbeit geht über die Implementierung eines Beispielszenarios hinaus, indem der modellierte Prozessablauf mit Ausnahmesituationen konfrontiert wird.

3 Realisierung von Prozessflexibilität mit der AristaFlow BPM Suite

3.1 Einführung in die AristaFlow BPM Suite

Die AristaFlow BPM Suite ist ein Produkt der AristaFlow GmbH in Ulm und ist das Ergebnis langjähriger Forschungsarbeiten des Instituts für Datenbanken und Informationssysteme (DBIS) der Universität Ulm in den Projekten ADEPT1, ADEPT2 und AristaFlow sowie den Erfahrungen von Industriepartnern aus deren Workflow- und Software-Projekten.

In Tabelle 2 werden die einzelnen Software-Komponenten der AristaFlow BPM Suite, die auf der ADEPT2-Technologie basiert, aufgeführt und kurz beschrieben [Aris12].

Tabelle 2: Software-Komponenten der AristaFlow BPM Suite

Software-Komponente	Beschreibung
Process Template Editor	<ul style="list-style-type: none"> - dient der Erstellung der Prozessvorlagen - Modellierung mit Hilfe von Plug & Play / Drag & Drop - unterstützt das Prinzip Correctness-by-Construction, das Fehler per Konstruktion während der Modellierung verhindert
Activity Repository Editor	<ul style="list-style-type: none"> - verwaltet die Aktivitäten/-vorlagen innerhalb der AristaFlow BPM Suite
Org Model Editor	<ul style="list-style-type: none"> - dient der Erstellung und Pflege des Organisationsmodells - unterstützt auch sehr komplexe Organisationsstrukturen
Test Client	<ul style="list-style-type: none"> - erlaubt die testweise Ausführung noch nicht vollständiger Prozessvorlagen - erstellt automatisch Formulare mit Ein- und Ausgabeparameter für einen schnellen Testbeginn
Standard Client	<ul style="list-style-type: none"> - Anzeige der einem Benutzer zugeordneten Aufgaben in einer Arbeitsliste - Starten neuer Prozessinstanzen über hinterlegte Prozessvorlagen - grafische Anzeige der Prozessinstanzen - bietet Funktionen wie Wiedervorlage, Delegation und Nachfrage
Automatic Client	<ul style="list-style-type: none"> - Ausführung automatischer Prozessschritte und deren Aktivitätenprogramme ohne Benutzerinteraktion
Process Monitor	<ul style="list-style-type: none"> - dient der Überwachung der laufenden Prozessinstanzen - Zurücksetzen einer fehlgeschlagenen Aktivität - erlaubt Ad-hoc-Anpassungen an einer Prozessinstanz mit systemseitiger Korrektheitsprüfung

Process Server	<ul style="list-style-type: none"> - Kernkomponente der AristaFlow BPM Suite - verwaltet die laufenden Prozessinstanzen, die instanzierbaren Prozessvorlagen, das Activity Repository und das Organisationsmodell - basiert auf einer erweiterbaren Architektur, die auf Skalierbarkeit, Performanz und Wartbarkeit ausgelegt ist
Open API	<ul style="list-style-type: none"> - deckt Entwicklungs- und Laufzeitfunktionen ab - ermöglicht die Entwicklung eigener Client-Anwendungen

Die serviceorientierte Architektur von ADEPT2 ermöglicht die Wiederverwendung der verfügbaren Funktionalität in verschiedenen Komponenten der Software und den Einsatz als API. Die Grobarchitektur von ADEPT2 mit den einzelnen Schichten wird in Abbildung 12 veranschaulicht. BT steht dabei für Buildtime (Entwurfszeit) und RT für Runtime (Ausführungszeit).

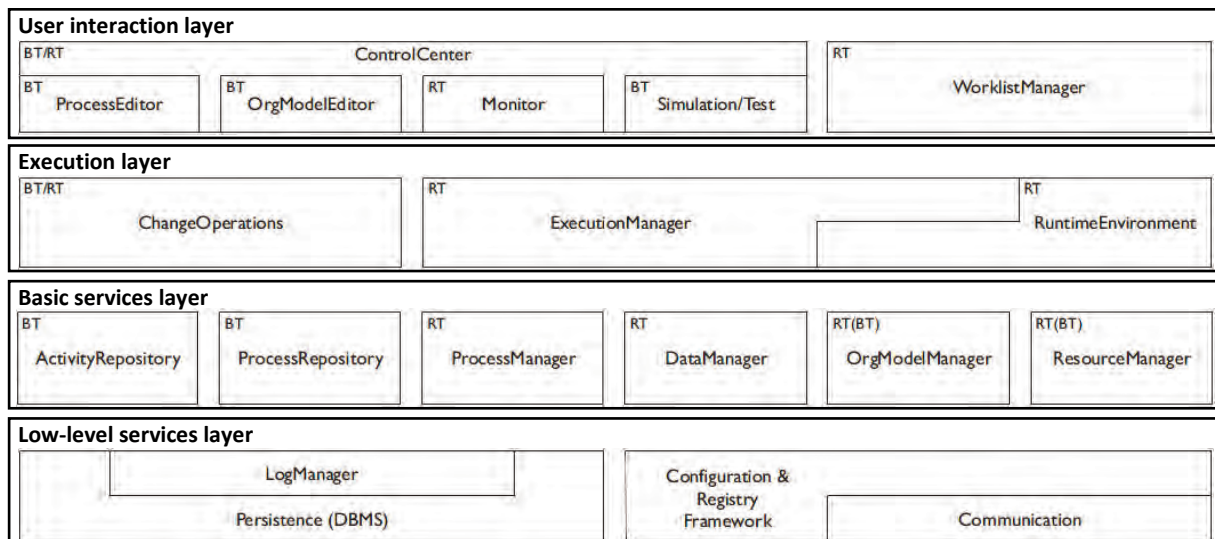


Abbildung 12: Grobarchitektur von ADEPT2 nach [Reic08]

Jede Schicht bietet die Dienste ihrer eigenen Komponenten den Komponenten darüberliegender Schichten an. In der untersten bzw. ersten Schicht (Low-level services) befinden sich systemnahe Dienste für die Protokollierung, die Persistenz, die Unterstützung der Konfiguration und Verwaltung der Systemkomponenten von ADEPT2 sowie für die Kommunikation zwischen den Komponenten.

Die Komponenten der zweiten Schicht (Basic services) stellen Dienste für die Verwaltung der unterschiedlichen Entwurfsdaten wie z.B. der Aktivitäts- und Prozessvorlagen sowie der Laufzeitdaten für die aktiven Prozessinstanzen mit den entsprechenden Werten der Datenelemente bereit.

Die operativen Komponenten von ADEPT2, die eine korrekte sowie effiziente Anpassung und Ausführung der Prozessinstanzen ermöglichen, sind in der Ausführungsschicht (Execution layer) zusammengefasst.

In der obersten Schicht (User interaction layer) werden die Entwurfs- und Laufzeitwerkzeuge für die Modellierer bzw. Administratoren von AristaFlow angeboten. Hier sind die Editoren für die Erstellung der Prozessvorlagen sowie Organisationsstrukturen als auch die Überwachungs- und Testwerkzeuge angesiedelt [Reic08].

3.2 Modellierung mit AristaFlow

Für die Modellierung mit AristaFlow stehen drei verschiedene Editoren zur Verfügung. Der Org Model Editor dient der Erstellung und Pflege des Organisationsmodells und unterstützt auch sehr komplexe Organisationsstrukturen.

Mit dem Activity Repository Editor werden die Aktivitätenvorlagen innerhalb der AristaFlow BPM Suite verwaltet. Alle Aktivitäten, die später genutzt werden sollen, müssen zuvor im Activity Repository Editor registriert und freigegeben werden.

Die eigentlichen Prozessmodelle werden mit dem Process Template Editor erstellt, der die prozessorientierte Komposition von Anwendungsfunktionen bzw. -services mit modernen Techniken wie Plug & Play und Drag & Drop unterstützt. Zur Vermeidung von Modellierungsfehlern wird der Prozessmodellierer bereits bei der Erstellung der Prozessmodelle mit Hilfe des Correctness-by-Construction-Prinzips geführt. Dadurch werden typische Fehler wie z.B. Verklemmungen während der Modellierung per Konstruktion verhindert. Dies schließt die systemseitige Prüfung der für einen korrekten Prozessablauf notwendigen nicht-optionalen Aufrufparameter der Anwendungsfunktionen mit ein, so dass auch ein konsistenter Datenfluss gewährleistet wird. Hierfür liegt der AristaFlow BPM Suite das ADEPT-Prozess-Meta-Modell zugrunde, das in Abbildung 13 dargestellt ist.

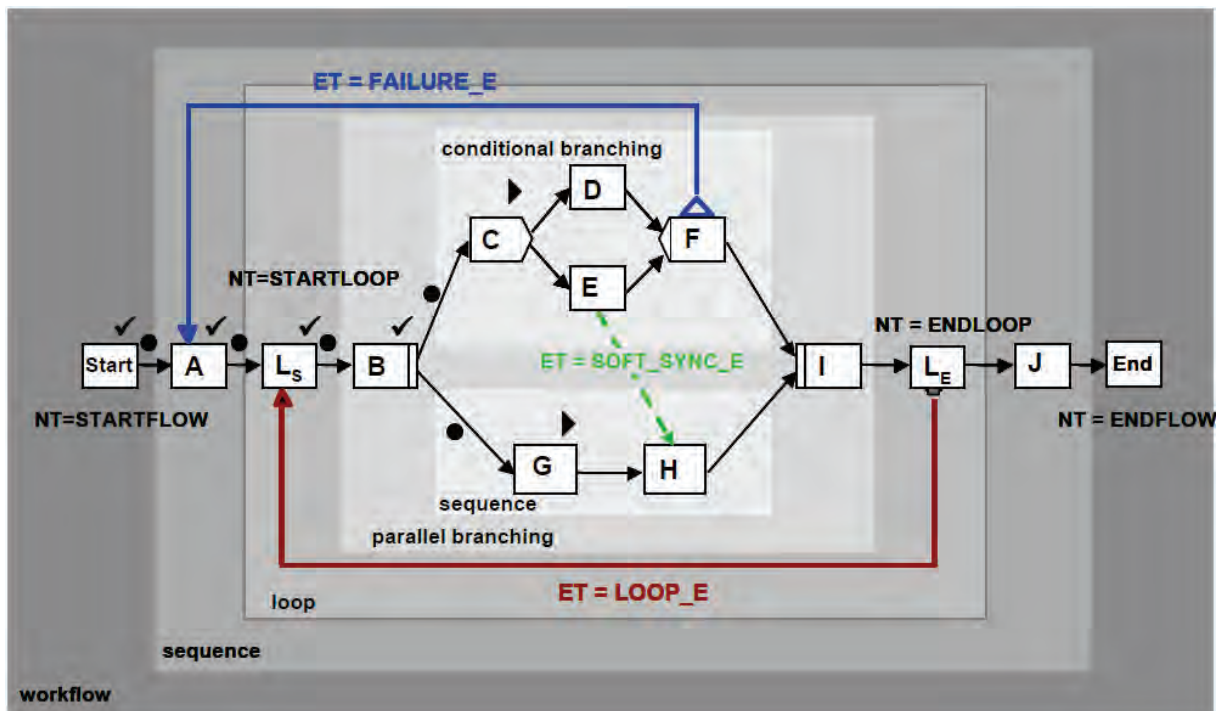


Abbildung 13: ADEPT-Prozess-Meta-Modell aus [Dada09]

Das Prozess-Meta-Modell ist trotz seiner einfachen Blockstruktur sehr ausdrucksstark und ermöglicht effiziente Korrektheitsanalysen des Prozessmodells sowohl während der Modellierung als auch bei Ad-hoc-Anpassungen.

Bei der Prozessmodellierung mit dem Process Template Editor wird direkt bei der Erstellung einer neuen Prozessvorlage ein Start- und ein Endknoten hinzugefügt, so dass initial ein strukturell konsistentes Prozessmodell vorliegt (Abbildung 14).

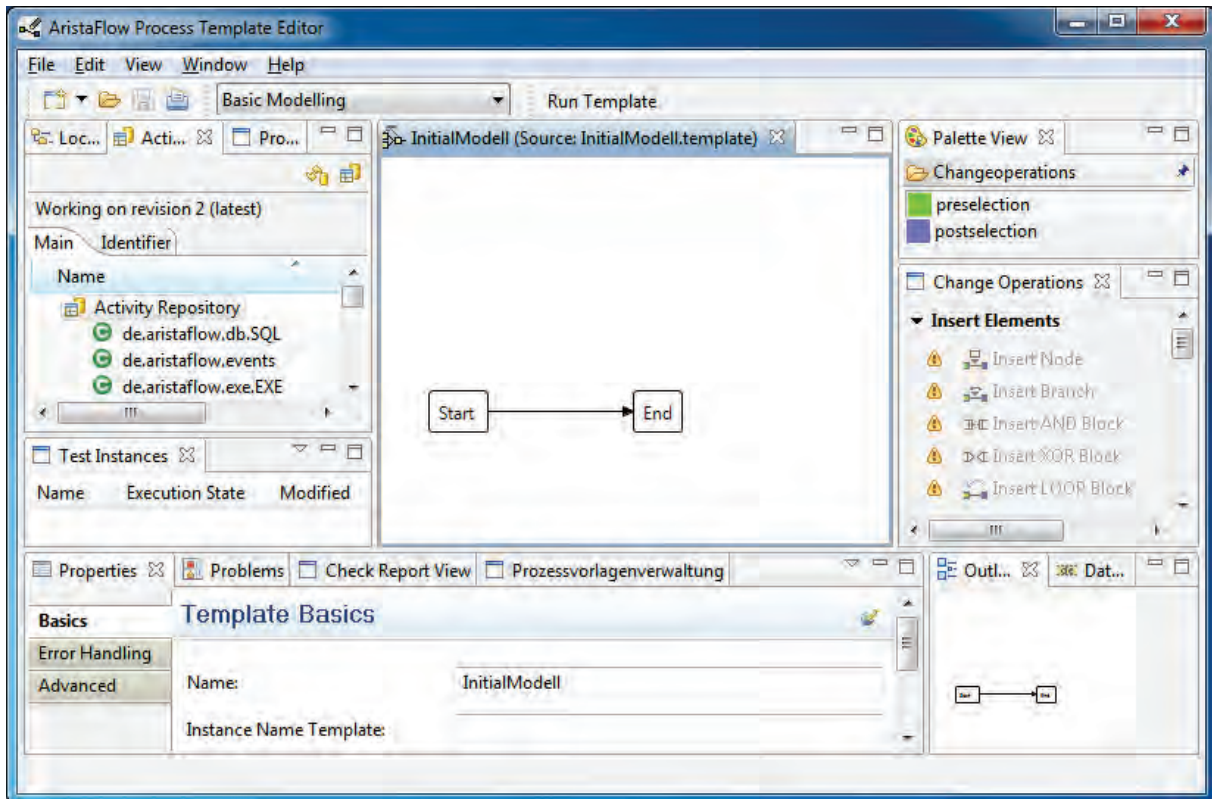


Abbildung 14: Initiale Prozessvorlage im AristaFlow Process Template Editor

Für die weitere Bearbeitung der Prozessvorlage werden dem Modellierer per Konstruktion kontextabhängig nur die Änderungsoperationen vom System angeboten, die ein strukturell korrektes Prozessschema in ein neues strukturell korrektes Schema überführen werden.

Abbildung 15 zeigt drei unterschiedliche Zustände (a) – (c) im Prozessmodell und die dazu verfügbaren Änderungsoperationen, die im Teilfenster „Change Operations“ auswählbar sind. Im Fall (a) sind keine Knoten markiert, so dass nur die Operation für das Einfügen eines Datenelements („Insert Data Element“) verwendet werden kann. Bei (b) ist der Knoten „Anamnese“ markiert. Daher können nun weitere zur Auswahl passende Operationen z.B. aus der Gruppe „Insert Surrounding Block“ oder das Löschen eines Knotens („Delete Node“) angewandt werden. Unpassende Operationen wie z.B. das Einfügen eines Knotens („Insert Node“) sind deaktiviert, da bei dem genannten Beispiel unklar ist, ob der Knoten vor oder hinter der Aktivität „Anamnese“ eingefügt werden soll. Das letzte Beispiel (c) markiert einen Bereich im Prozessmodell. Der Knoten „Anamnese“ ist mit Preselection für den Beginn und der Knoten „Untersuchung“ mit Postselection für das Ende des Bereichs ausgewählt worden.

Auch hier werden nur die Änderungsoperationen aktiviert, die in diesem Kontext eindeutig definiert und anwendbar sind, um ein weiterhin strukturell konsistentes Prozessmodell zu gewährleisten. So können u.a. ein neuer Knoten („Insert Node“) oder ein „AND, XOR bzw. Loop Block“ eingefügt werden. Die Operation zum Löschen eines Knoten steht nun nicht mehr zur Verfügung, da bei zwei markierten Knoten unklar ist, welcher gelöscht werden soll.

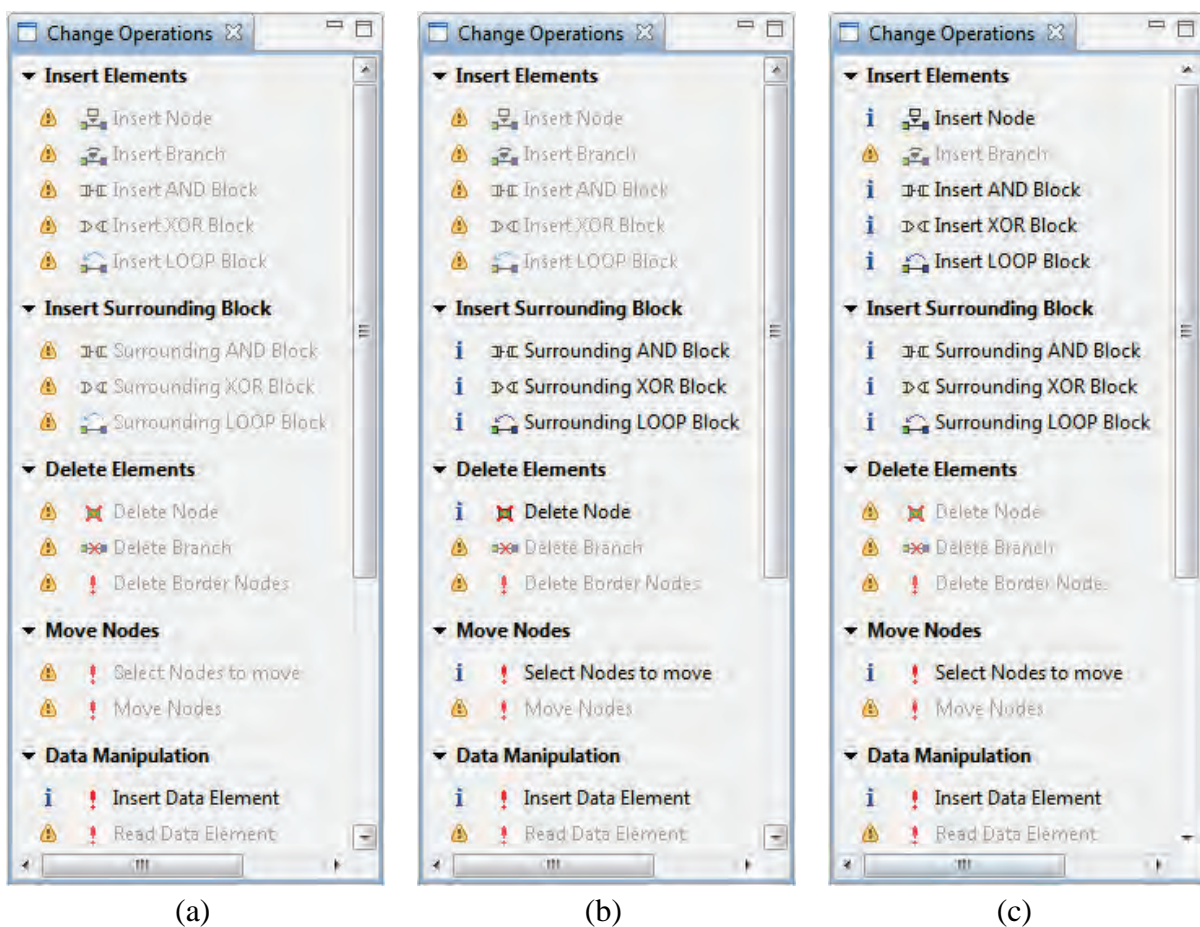
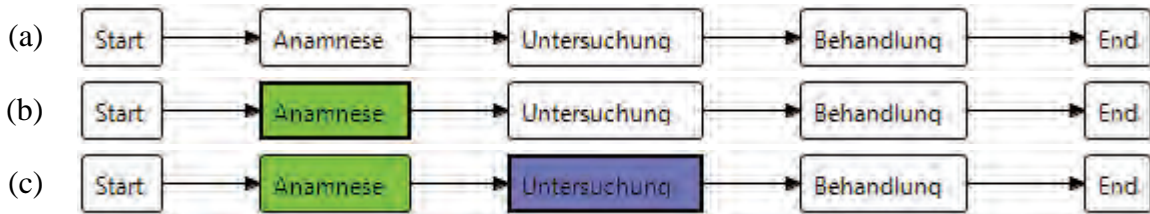


Abbildung 15: Kontextabhängige Änderungsoperationen in AristaFlow

Zusätzlich zur Sicherstellung der strukturellen Konsistenz des Prozessmodells per Konstruktion wird auch der Datenfluss nebenläufig auf Korrektheit analysiert. Gefundene Fehler bzw. Inkonsistenzen werden im Process Template Editor im Teilfenster „Problems“ und am jeweiligen Element angezeigt. So kann frühzeitig erkannt werden, ob alle benötigten Datenelemente für einen Prozessschritt bei allen Konstellationen zur Verfügung stehen oder nicht. Die Abbildung 16 zeigt, dass das Datenelement „PatientenID“ im Prozessschritt

„Behandlung“ als Eingabeparameter benötigt wird. Dieses Datenelement steht jedoch nur zur Verfügung, wenn eine neue Patientenakte angelegt wird, da bei diesem Schritt die „PatientenID“ gesetzt wird. Im anderen Fall fehlt diese Information, so dass der Prozessschritt „Behandlung“ nicht ordnungsgemäß mit seinen benötigten Parametern versorgt wird. Dieser Fehler im Datenfluss des Prozessmodells wird von AristaFlow im Teilfenster „Problems“ dargestellt. Zusätzlich erhalten die betroffenen Elemente eine Markierung.

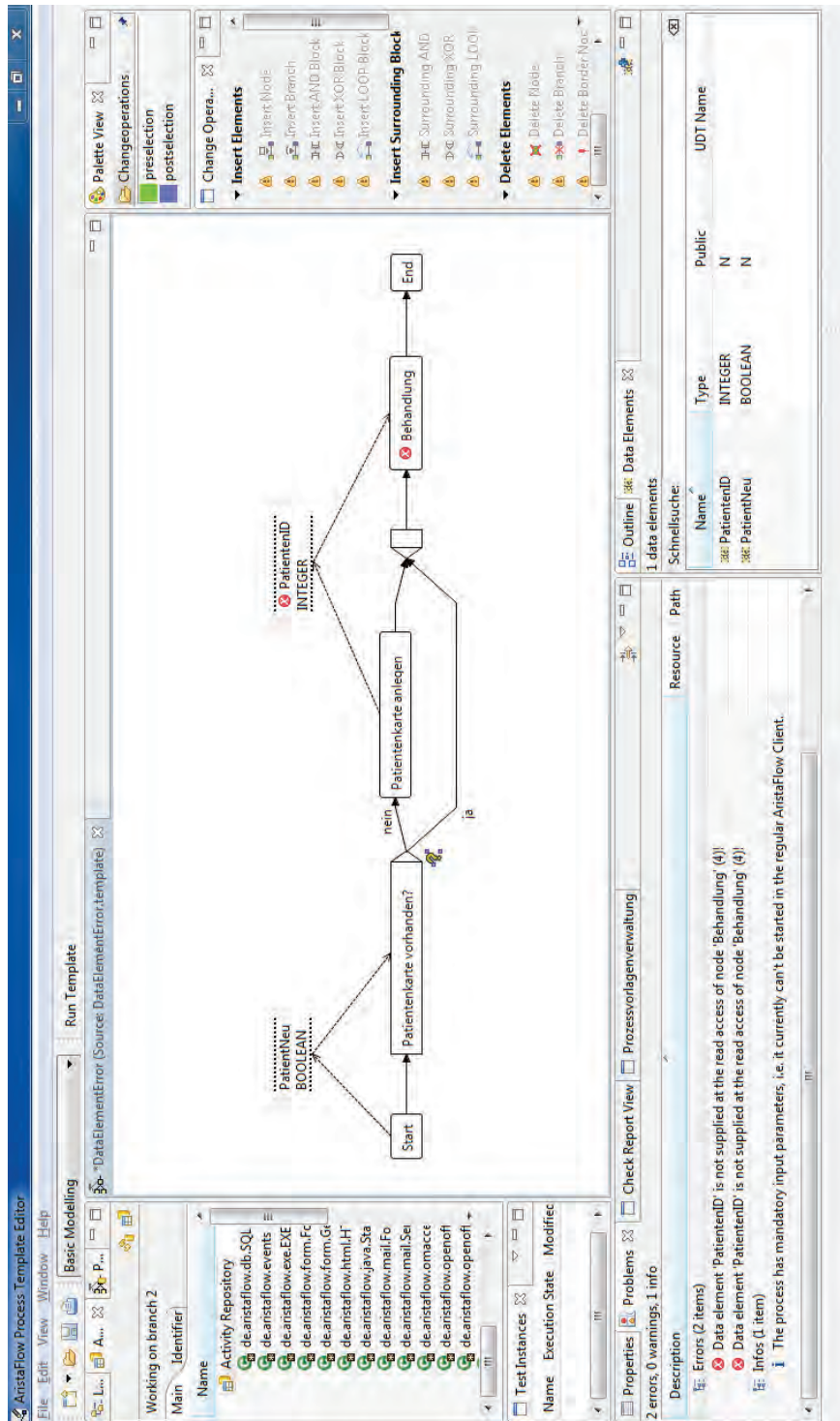


Abbildung 16: Fehlermeldung eines inkonsistenten Datenflusses

3.2.1 Unterstützung von Ad-hoc-Anpassungen

AristaFlow unterstützt den Prozessmodellierer bei Ad-hoc-Änderungen zur Laufzeit mit den gleichen Möglichkeiten und Korrektheitsanalysen wie zur Entwurfszeit. Dies umfasst sowohl die Überprüfung des Kontroll- als auch des Datenflusses hinsichtlich ihrer strukturellen Korrektheit und Konsistenz, die per Konstruktion von AristaFlow gewährleistet werden [ReDa98, Reic00]. Darüber hinaus wird auch der aktuelle Zustand der Prozessinstanz für die Zulässigkeit der gewünschten Anpassung berücksichtigt, so dass z.B. Änderungen, die in der Vergangenheit liegen, nicht statthaft sind. Jede Veränderung an einem laufenden Prozess wird von AristaFlow zur späteren Nachvollziehbarkeit in einem Änderungsprotokoll im Detail registriert.

Damit autorisierte Endanwender ebenfalls Ad-hoc-Änderungen auf Prozessinstanzebene realisieren können, lassen sich die hierfür notwendigen Funktionen in Form einer API als Systemschnittstelle nutzen. Dafür wird aus der Ausführungsschicht die Komponente „ChangeOperations“ (Abbildung 12) verwendet, die kontextabhängig die zulässigen Änderungsoperationen liefert und die Konsistenzprüfungen unter Berücksichtigung des Prozesszustandes durchführt [Dada10]. In [DRR11] wird der Einsatz dieser Systemschnittstelle skizziert, um eine Interaktion mit berechtigten Endbenutzern zu realisieren. Das Beispiel beschreibt einen Oberarzt, der die Entscheidung eines Ambulanzarztes in Bezug auf eine Operation bestätigen oder ablehnen soll. Der Oberarzt kann aufgrund seiner Berechtigung auch eingeschränkte Ad-hoc-Anpassungen an der Prozessinstanz durchführen, die nicht im Prozessschema vormodelliert sind. Dazu kann er über einen „Ausnahmeknopf“ bestimmte Änderungsoperationen wie z.B. das Einfügen eines zusätzlichen Schrittes an der Instanz vornehmen, die von der Systemschnittstelle auf Konsistenz und Korrektheit überprüft und im Prozesslog protokolliert werden.

Im folgenden Beispiel sollen zu zwei unterschiedlich weit fortgeschrittenen Prozessinstanzen gemäß dem Prozessmodell aus Abbildung 17 ad hoc ein weiterer Prozessschritt „Quittung ausstellen“ direkt nach der Aktivität „Praxisgebuehr entgegennehmen“ eingefügt werden. Da der Prozess (a) in Abbildung 18 im Gegensatz zum Prozess (b) nicht so weit fortgeschritten ist (zu erkennen an den gelb markierten Knoten), steht in AristaFlow für (a) die Änderungsoperation zum Einfügen eines Knotens („Insert Node“) zur Verfügung und für (b) nicht. Das Ergebnis der Ad-hoc-Anpassung für (a) ist in Abbildung 19 dargestellt.

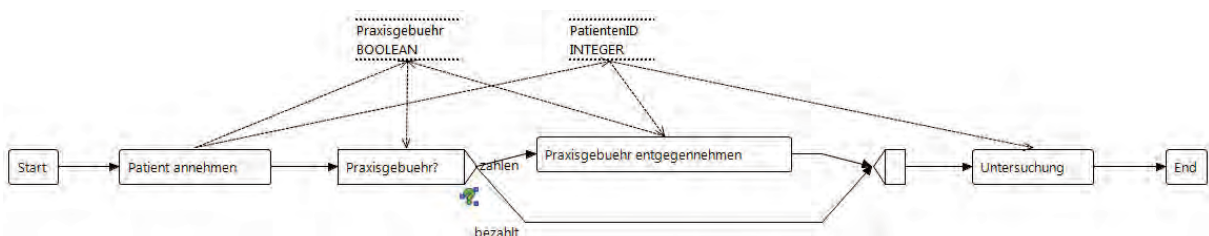


Abbildung 17: Beispiel eines Prozessmodells für eine spätere Ad-hoc-Anpassung

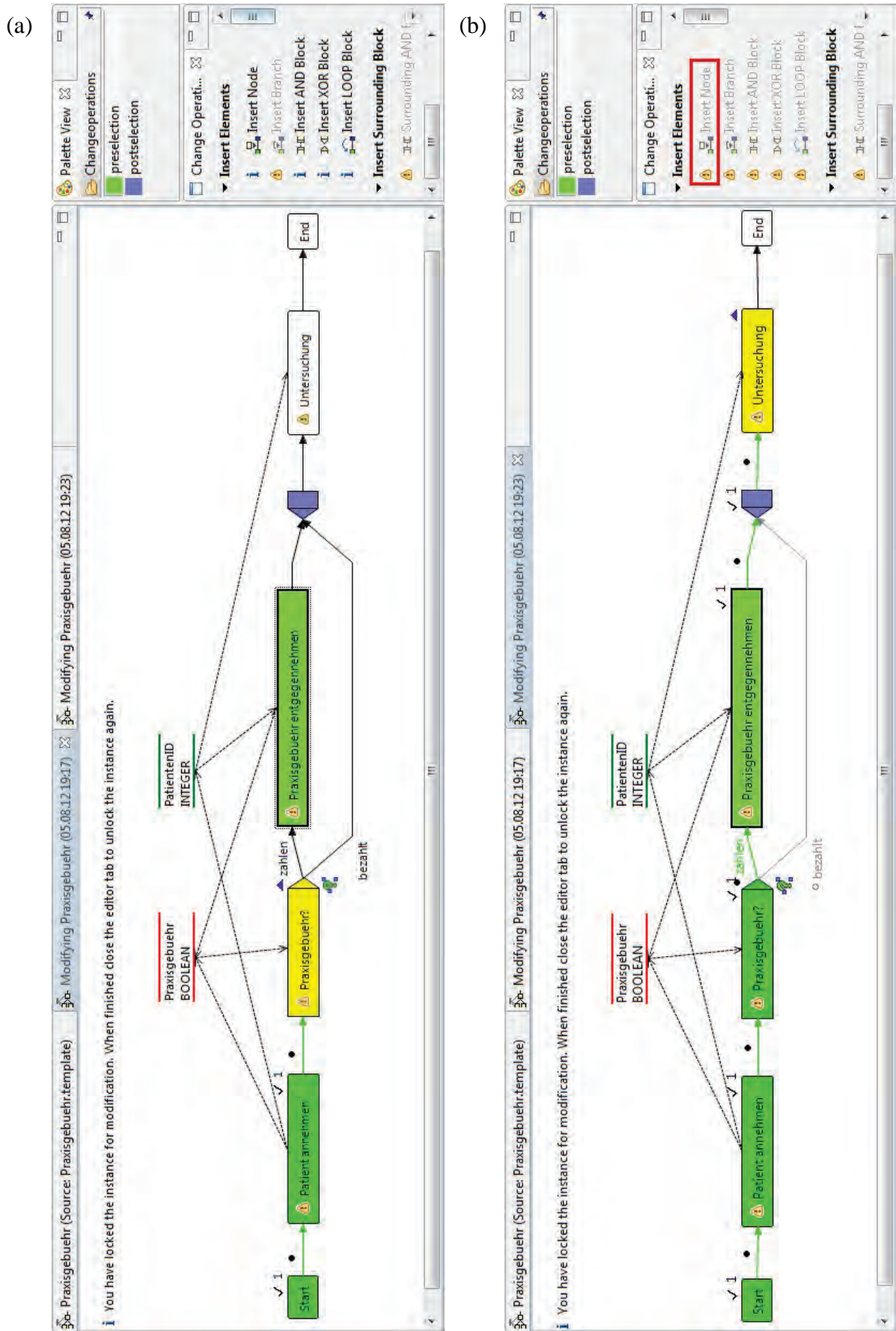


Abbildung 18: Versuch einer Ad-hoc-Anpassung an zwei Prozessinstanzen

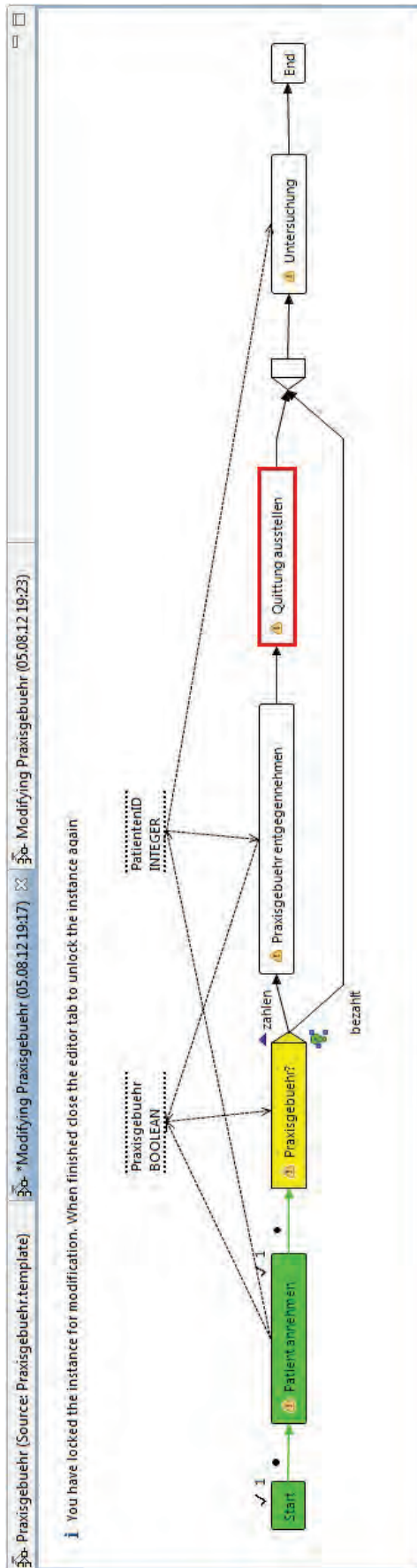


Abbildung 19: Prozessinstanz (a) nach Einfügen des zusätzlichen Prozessschrittes

3.2.2 Prozessschemaevolution in AristaFlow

Die Prozessschemaevolution ist als umfassendes Konzept im ADEPT2-Projekt entwickelt worden und erlaubt, unverzerrte als auch verzerrte Prozessinstanzen auf ein neues Prozessschema zu migrieren, sofern die systemseitigen Überprüfungen erfolgreich durchlaufen werden. Bei der Migration einer laufenden Instanz werden die Bedingungen zur Gewährleistung der Konsistenz genauso beachtet wie der entsprechende Prozessfortschritt und Ausführungszustand. Im Fall von verzerrten Prozessinstanzen wird zusätzlich die Verträglichkeit der vorgenommenen Ad-hoc-Anpassungen mit dem neuen Schema untersucht und berücksichtigt. Der Ablauf einer Prozessschemaevolution sieht wie folgt aus [DRR11]:

- Der Prozessmodellierer führt die gewünschten Modifikationen an einem bestehenden Prozessmodell aus und speichert das angepasste Schema als neue Version ab.
- Das ADEPT2-System prüft alle nach dem alten Schema gestarteten Prozessinstanzen auf ihre verträgliche Migration auf das neue Modell ab und gibt eine Ergebnisliste aller untersuchten Instanzen hinsichtlich ihrer Migrationsfähigkeit mit Begründung aus.
- Anschließend kann der Prozessmodellierer noch migrierbare Instanzen ausschließen, bevor er das ADEPT2-System mit der Migration der restlichen Prozessinstanzen auf das neue Prozessschema beauftragt.

Derzeit ist die direkte Ausführung einer Prozessschemaevolution über die Oberfläche der AristaFlow BPM Suite noch nicht möglich, sondern kann nur unter programmtechnischer Verwendung der ADEPT2-Technologie nutzbar gemacht werden.

3.3 Vergleich von AristaFlow mit anderen BPM-Werkzeugen

Zur Modellierung und Unterstützung der Geschäftsabläufe eines Unternehmens befinden sich am Markt eine Vielzahl von weiteren BPM-Werkzeugen. An dieser Stelle seien beispielhaft die Oracle BPM Suite, IBM WebSphere, die Adobe® LiveCycle® Enterprise Suite, die ARIS Platform und webMethods BPMS (beide von der Software AG) erwähnt. Für den Vergleich mit AristaFlow wird die Oracle BPM Suite 11g herangezogen. Es werden die Fähigkeiten in Bezug auf die Flexibilität zur Entwurfs- sowie Ausführungszeit und zur Prozessschemaevolution gegenübergestellt.

Alle hier aufgeführten BPM-Werkzeuge verfügen über eine grafische Oberfläche mit Drag & Drop-Funktionalität, so auch die Oracle BPM Suite, die die Erstellung eines Prozessmodells mit Hilfe des Oracle JDevelopers erlaubt. Das Projektfenster mit den auswählbaren Aktivitäten und weiteren Elementen ist in Abbildung 20 dargestellt.

Oracle verwendet für die grafische Prozessmodellierung die BPMN 2.0 und kann BPMN- und BPEL-Prozesse ausführen. Somit bietet Oracle eine ähnliche Flexibilität für die Prozesserstellung zur Entwurfszeit wie AristaFlow an. Jedoch ist die Komplexität in Oracle für die Erstellung eines Prozessmodells weitaus höher, aber auch mächtiger aufgrund der verwendeten BPMN.

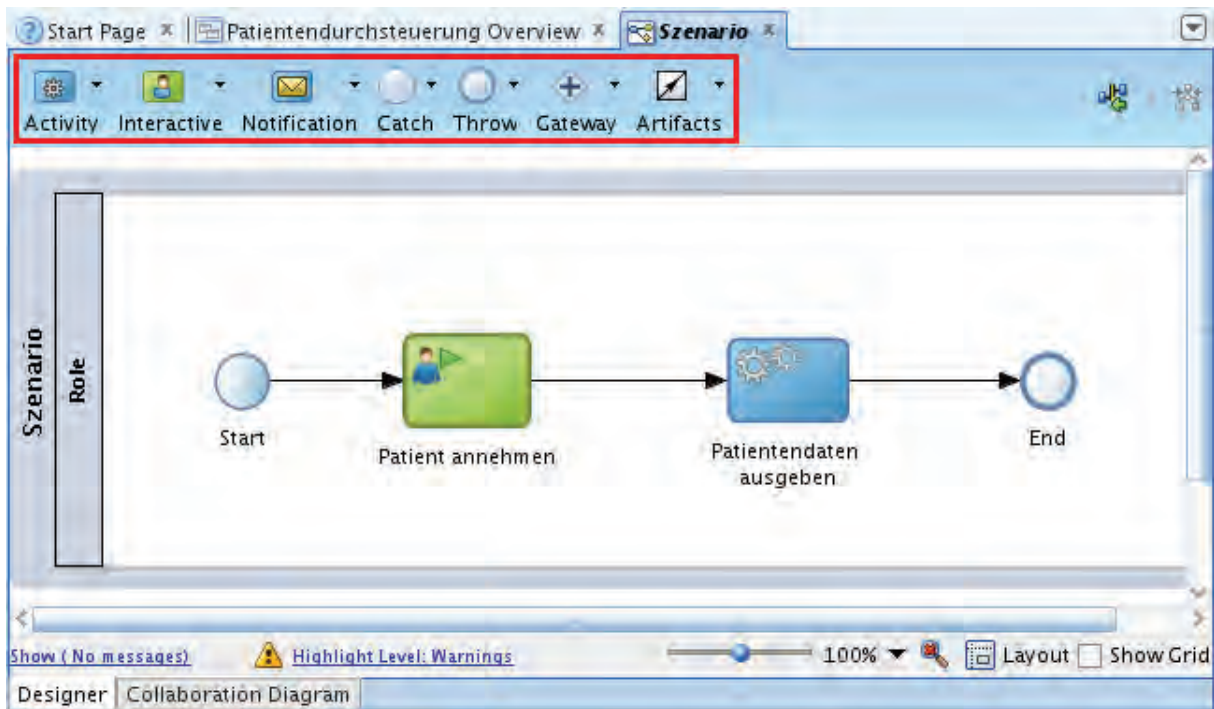


Abbildung 20: Grafische Oberfläche des Oracle Jdevelopers

Aufgetretene Fehler und Meldungen während der Modellierung werden in Oracle ebenfalls in einem Teilfenster protokolliert, das in Abbildung 21 gezeigt wird.

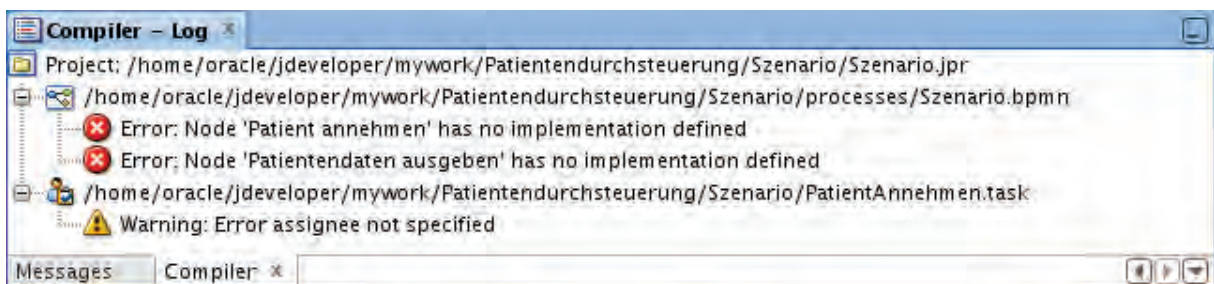


Abbildung 21: Teilfenster für aufgetretene Fehler und Meldungen

Die Oracle BPM Suite unterstützt keine Prozessmodellierung per Konstruktion, um strukturell konsistente Prozessmodelle in neue strukturell konsistente Modelle zu überführen. So kann z.B. in Oracle – wie zuvor in Abbildung 5 dargestellt – ein AND- mit einem XOR-Konnektor abgeschlossen werden.

Gemäß [Orac12] ermöglicht Oracle im „Process Workspace“ auch die Anpassung von Prozessinstanzen zur Ausführungszeit. Dort kann eine Prozessinstanz ausgewählt, angehalten, geändert und schließlich fortgesetzt werden. Die angebotenen Änderungsmöglichkeiten beschränken sich allerdings auf den Kontrollfluss und die Veränderung von Instanzattributen. So kann z.B. ein anderer Prozessschritt im Kontrollfluss aktiviert, aber im Gegensatz zur AristaFlow BPM Suite keine neue Aktivität eingefügt werden.

Ebenso lassen sich eine oder mehrere Prozessinstanzen über die Benutzeroberfläche im „Process Workspace“ auswählen und auf eine neue Prozessmodellversion migrieren. Wenn Fehler bei der Migration auftreten, so werden die Gründe für das Scheitern in einer Dialogbox angezeigt.

Die AristaFlow BPM Suite unterstützt zwar prinzipiell die Prozessschemaevolution, allerdings ist diese Funktionalität nicht über die grafische Oberfläche ausführbar, sondern kann nur programmtechnisch mit Hilfe der angebotenen API nutzbar gemacht werden.

Der Vergleich der beiden BPM Suites zeigt, dass Oracle hinsichtlich der betrachteten Flexibilitätsaspekte die technologische Weiterentwicklung von AristaFlow ebenfalls weitestgehend nachvollzogen hat. Während AristaFlow flexiblere Ad-hoc-Anpassungen zur Ausführungszeit ermöglicht, kann indes nur Oracle die Prozessschemaevolution direkt über die Benutzeroberfläche im „Process Workspace“ anbieten.

Die Ergebnisse des Vergleichs sind nochmals in Tabelle 3 zusammengefasst dargestellt.

Tabelle 3: Vergleich der AristaFlow BPM Suite mit der Oracle BPM Suite

	AristaFlow BPM Suite	Oracle BPM Suite
Flexibilität zur Entwurfszeit	<ul style="list-style-type: none"> - Grafische Modellierung (Process Template Editor) - Drag & Drop - Correctness-by-Construction - Fehlerprotokoll 	<ul style="list-style-type: none"> - Grafische Modellierung (Jdeveloper) - Drag & Drop - Ø - Fehlerprotokoll
Flexibilität zur Ausführungszeit	<ul style="list-style-type: none"> - Ad-hoc-Anpassungen werden unterstützt - Correctness-by-Construction 	<ul style="list-style-type: none"> - Ad-hoc-Anpassungen werden eingeschränkt unterstützt - Ø
Prozessschemaevolution	<ul style="list-style-type: none"> - Integriert in ADEPT2, aber nicht direkt in AristaFlow nutzbar - Fehlerprotokoll 	<ul style="list-style-type: none"> - Oracle Process Workspace - Fehlerprotokoll
Technologie	<ul style="list-style-type: none"> - ADEPT2 	<ul style="list-style-type: none"> - BPMN und BPEL

4 Umsetzung eines Beispielszenarios mit der AristaFlow BPM Suite

4.1 Szenario aus der Krankenhausdomäne

Das Beispielszenario aus der Krankenhausdomäne repräsentiert einen Ablauf in einer großen Augenarztpraxis für die „Patientendurchsteuerung“.

In diesem Szenario vereinbart ein Patient telefonisch einen Termin oder kommt ohne Termin in die Arztpraxis, die von einem Chefarzt geleitet wird.

Bei der Anmeldung durch eine Arzthelferin wird abhängig vom Versichertenstatus die Versichertenkarte des Patienten eingelesen und bei Fälligkeit die Praxisgebühr entgegengenommen. Darüber hinaus wird festgelegt, welche Vor- bzw. Zusatzuntersuchung durchgeführt werden soll. Wenn es sich um einen neuen Patienten handelt, so wird zuerst noch eine neue Patientenkarte angelegt.

Die Voruntersuchungen umfassen das Ausmessen der Brille, die Bestimmung des Visus und die Ermittlung der objektiven Refraktion. Dafür stehen zwei Voruntersuchungsräume zur Verfügung. Für die Reihenfolge und Zuweisung der Patienten auf die Voruntersuchungsräume ist die Rennerin, eine spezielle Helferin, verantwortlich. Während der Voruntersuchung kann sich die Notwendigkeit für eine Zusatzbehandlung ergeben. Neben diesen ungeplanten Zusatzbehandlungen sind auch zuvor vereinbarte Zusatzbehandlungen vorhanden.

Als Zusatzbehandlungen können das Messen des Gesichtsfeldes, die Bestimmung der Stärke der Intraokularlinse (IOL), eine optische Kohärenztomografie (OCT), die Sehschule und die Laserbehandlung durchgeführt werden. Die ersten drei Zusatzbehandlungen werden von einer Helferin ausgeführt, für die Sehschule ist eine Orthoptistin oder ein Arzt zuständig und die Laserbehandlung wird von einem Arzt vorgenommen.

Nach der Vor- bzw. Zusatzuntersuchung findet stets eine Behandlung in einem Sprechzimmer durch einen Arzt statt, die ebenfalls eine weitere Zusatzbehandlung nach sich ziehen kann. Die Verteilung auf die einzelnen Sprechzimmer wird durch eine weitere spezielle Helferin, der Verteilerin, vorgenommen. Bei Bedarf werden noch nach Abschluss der Behandlung ein Rezept ausgestellt und evtl. ein Folgetermin vereinbart.

Der Prozessablauf ist in Abbildung 22 bis Abbildung 28 als ereignisgesteuerte Prozesskette (EPK) grafisch dargestellt.

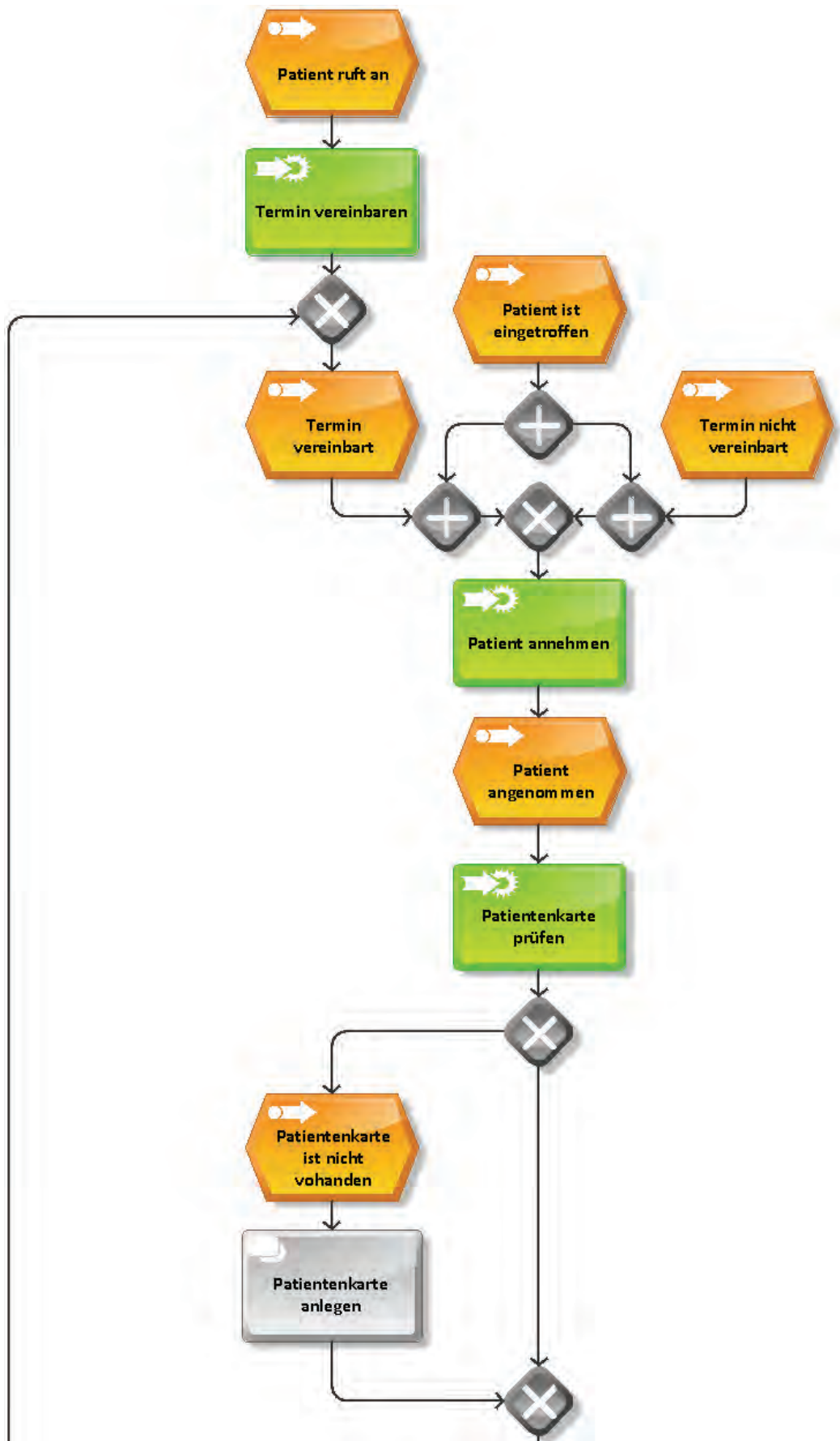


Abbildung 22: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 1 von 7)

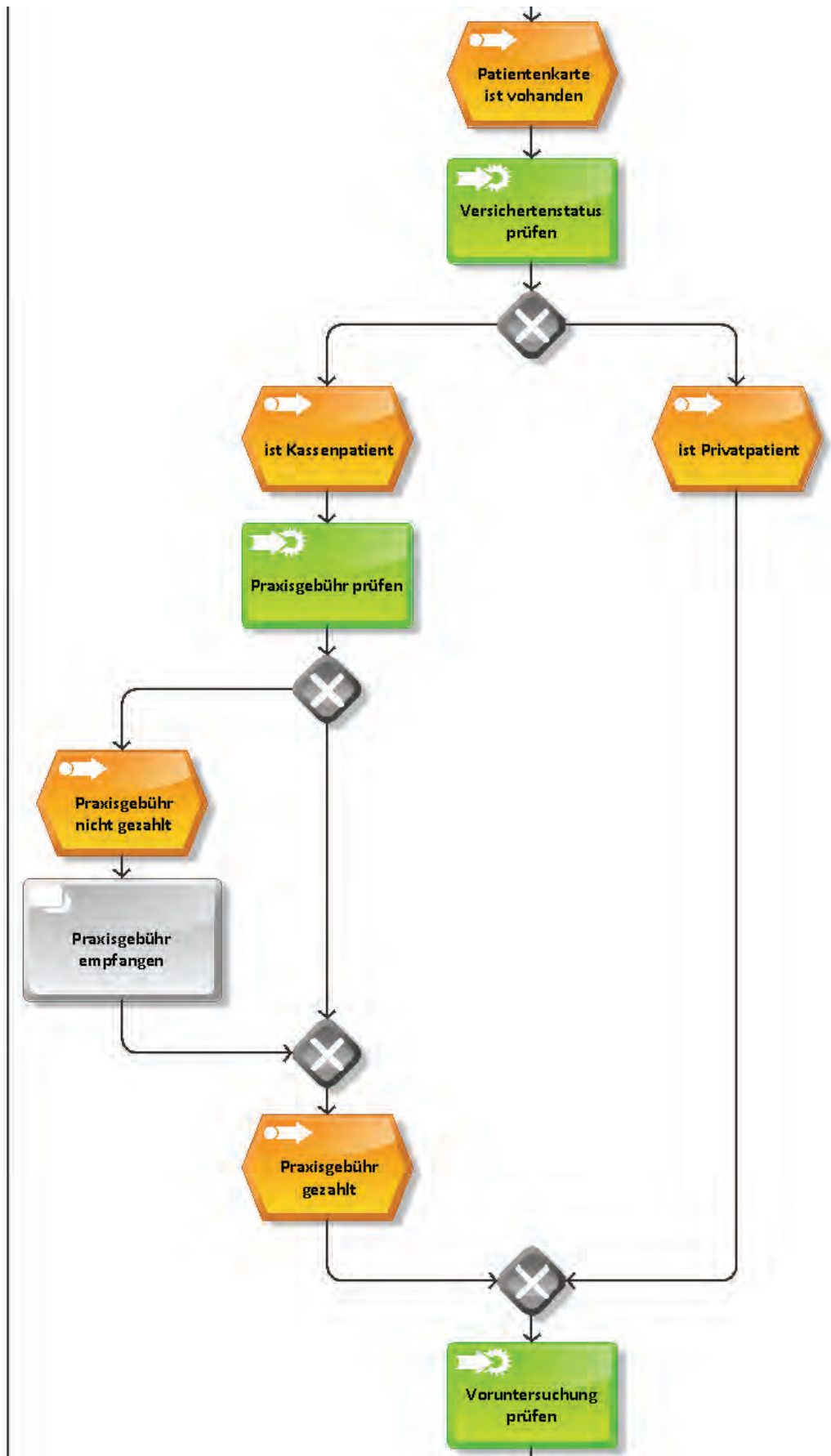


Abbildung 23: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 2 von 7)

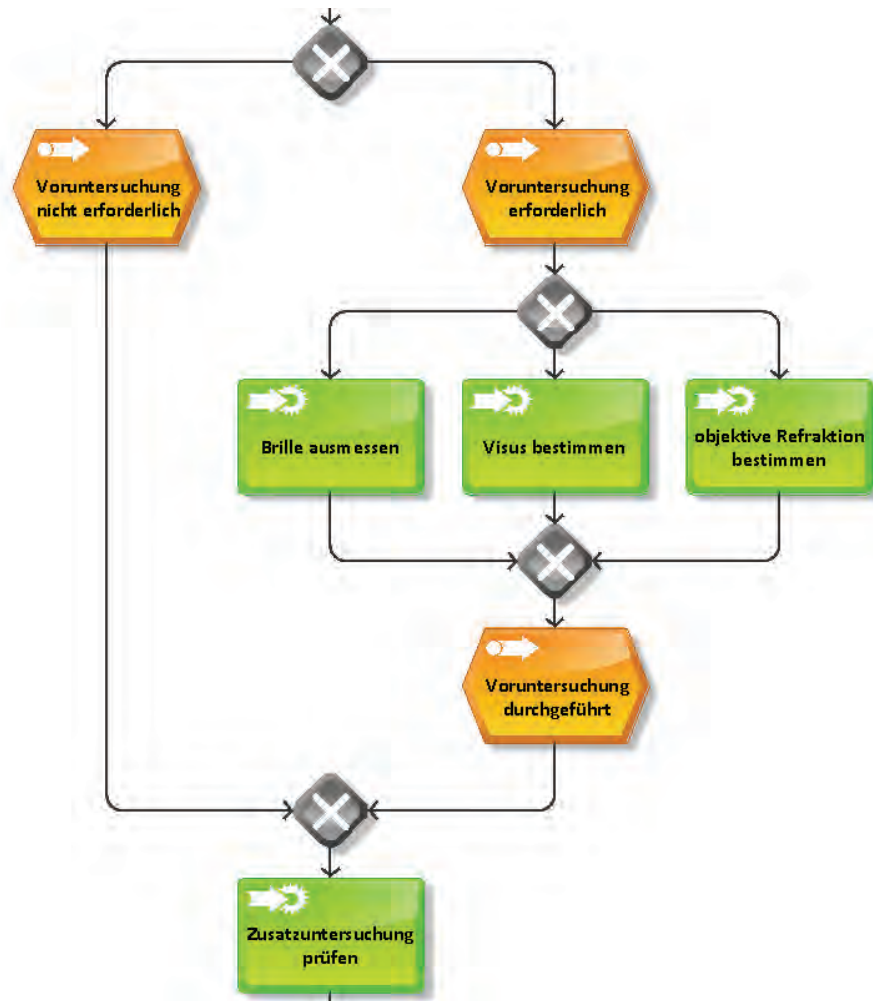


Abbildung 24: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 3 von 7)

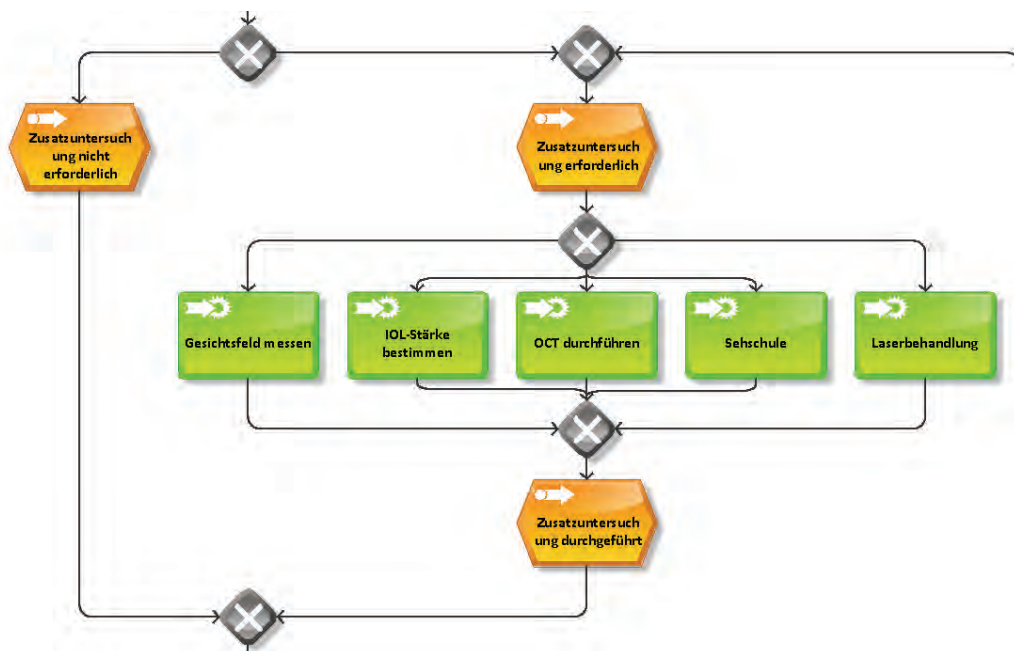


Abbildung 25: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 4 von 7)

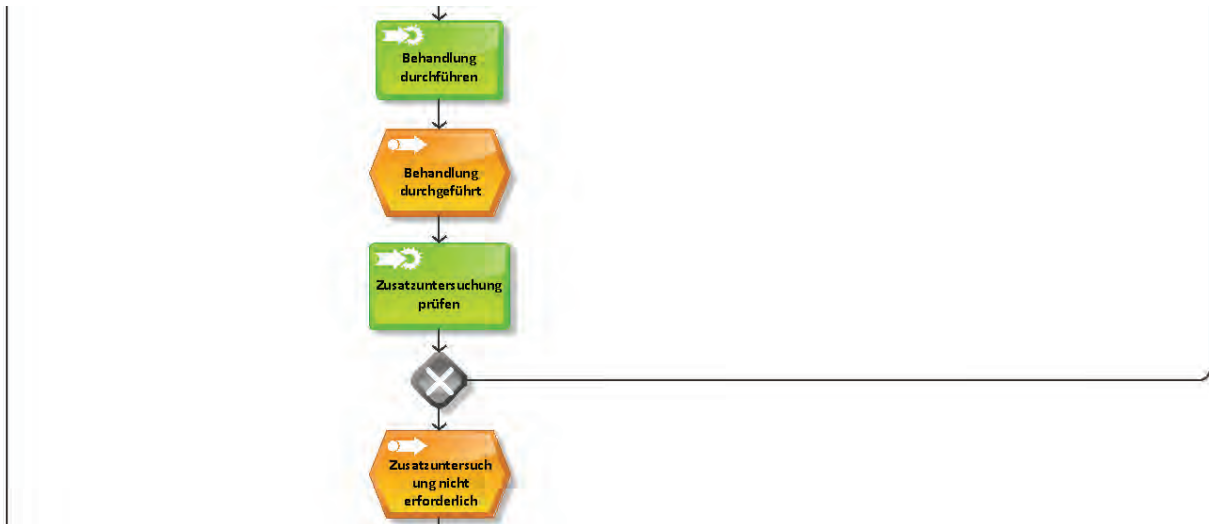


Abbildung 26: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 5 von 7)

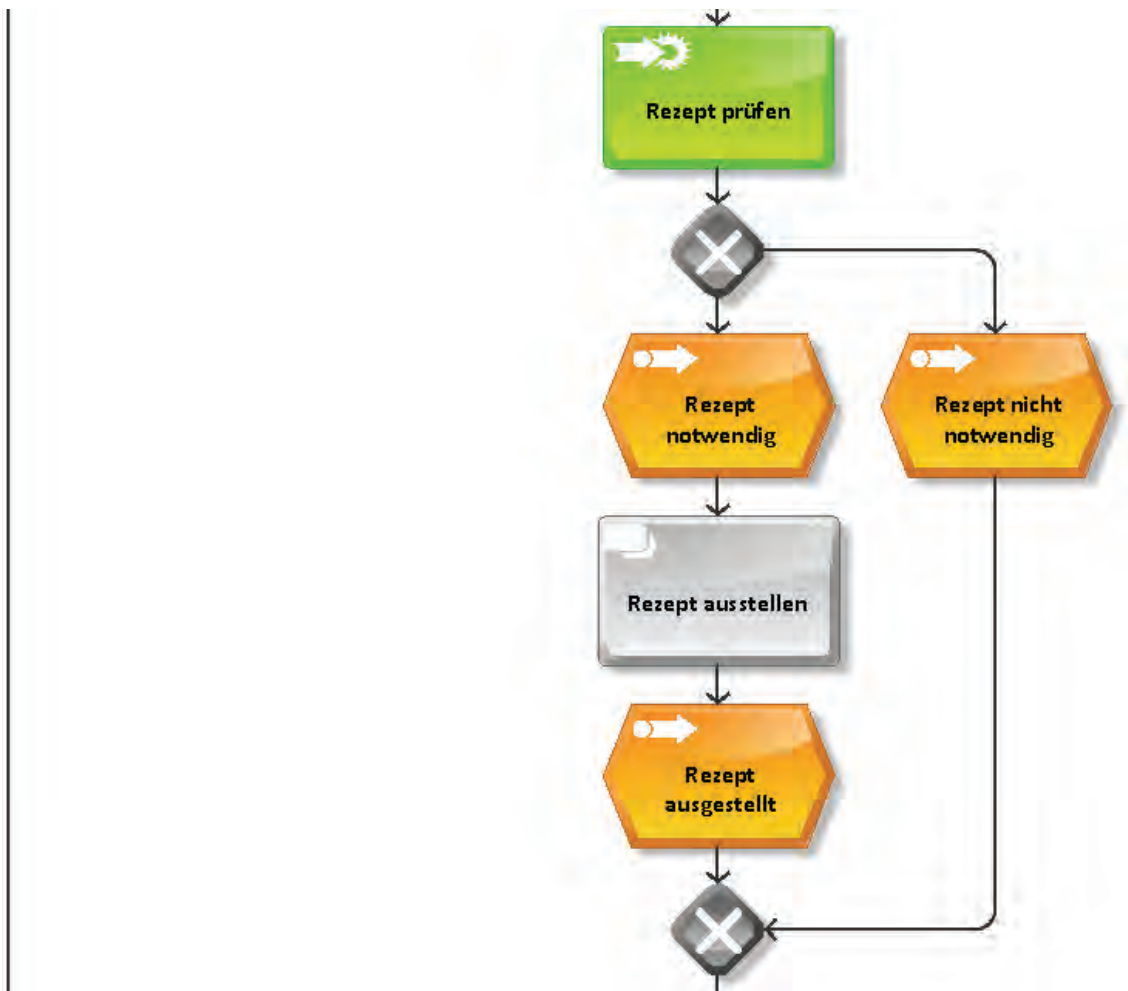


Abbildung 27: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 6 von 7)

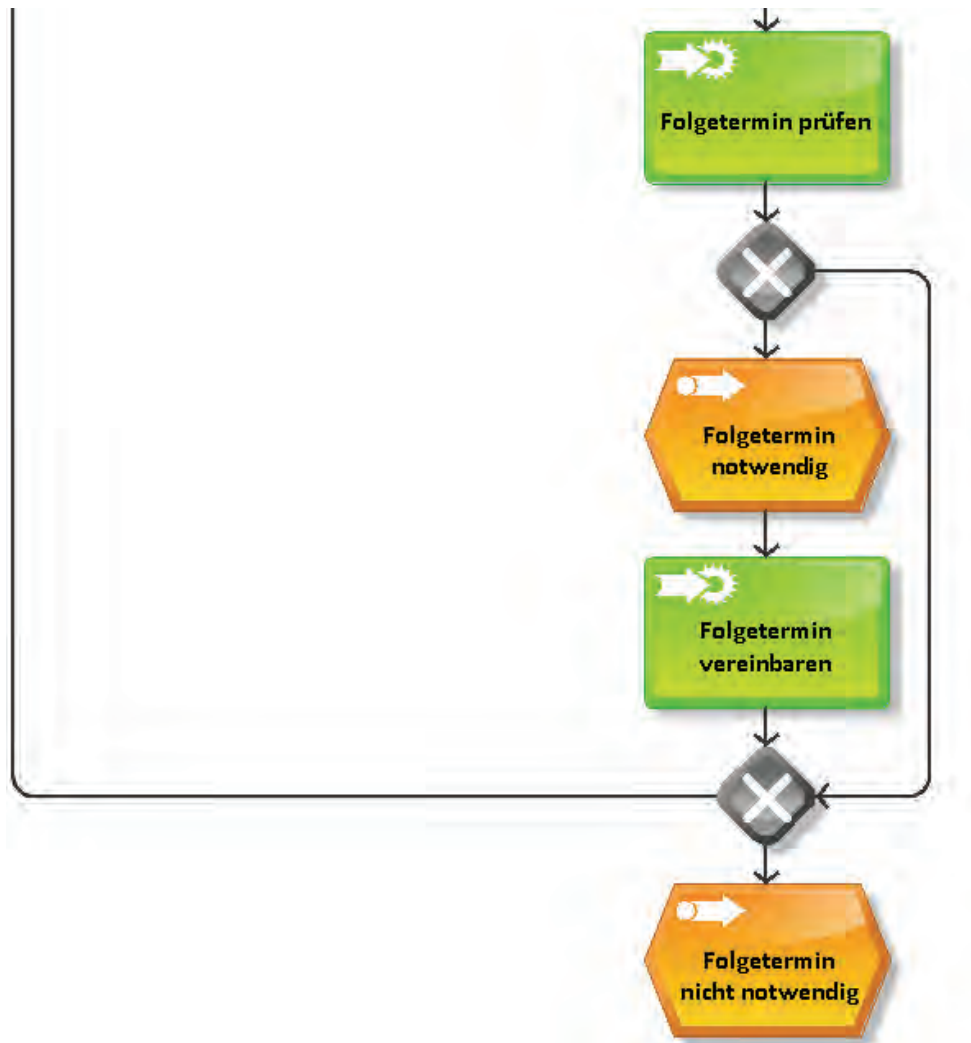


Abbildung 28: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 7 von 7)

4.2 Umsetzung des Szenarios

Zur Umsetzung des Szenarios wird die AristaFlow BPM Suite in der Version 1.0.1 – r57 verwendet. Die Implementierung in AristaFlow startet mit der Anmeldung eines Patienten in der Augenarztpraxis. Die Terminvereinbarung ist nicht Gegenstand der Prozessvorlagenerstellung. Der Datenfluss wird nur mit unbedingt notwendigen Datenelementen erstellt. Auf eine persistente Ablage der Daten, z.B. in einer Datenbank, wird verzichtet, da die Umsetzung lediglich eine Simulation des Prozesses erlauben soll. Für die Erstellung des Organisationsmodells in AristaFlow liegt das in Abbildung 29 abgebildete Organigramm der Augenarztpraxis zugrunde. Ein Chefarzt übernimmt die Leitung der Augenarztpraxis. Diverse Ärzte, die Orthoptisten und einige Helferinnen sind dem Chefarzt unterstellt. Zu den Helferinnen zählen sowohl die Rennerin als auch die Verteilerin. Für die Simulation des Szenarios werden beispielhaft zwei Ärzte, eine Orthoptistin, vier Helferinnen sowie eine Rennerin und eine Verteilerin eingerichtet.

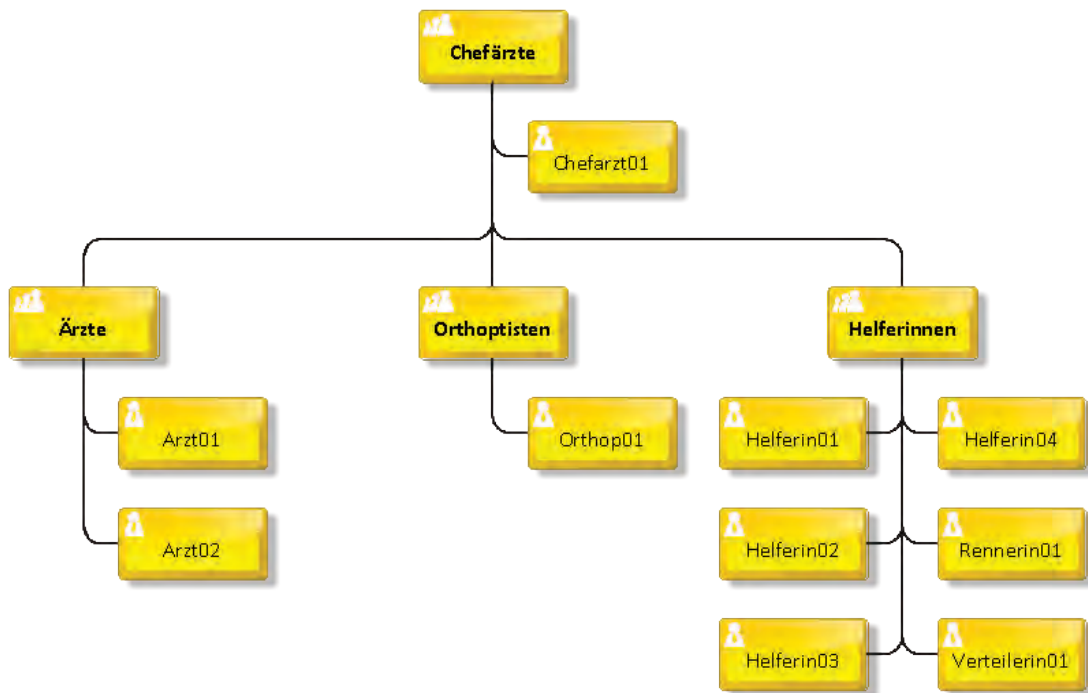


Abbildung 29: Organigramm für die Augenarztpraxis

Die Anlage des Organisationsmodells erfolgt mit dem AristaFlow Org Model Editor. In Abbildung 30 ist exemplarisch die Umsetzung der Organisationsposition (OrgPosition) der „Helferinnen“ mit den zugehörigen Vertretern (Agents) dargestellt.

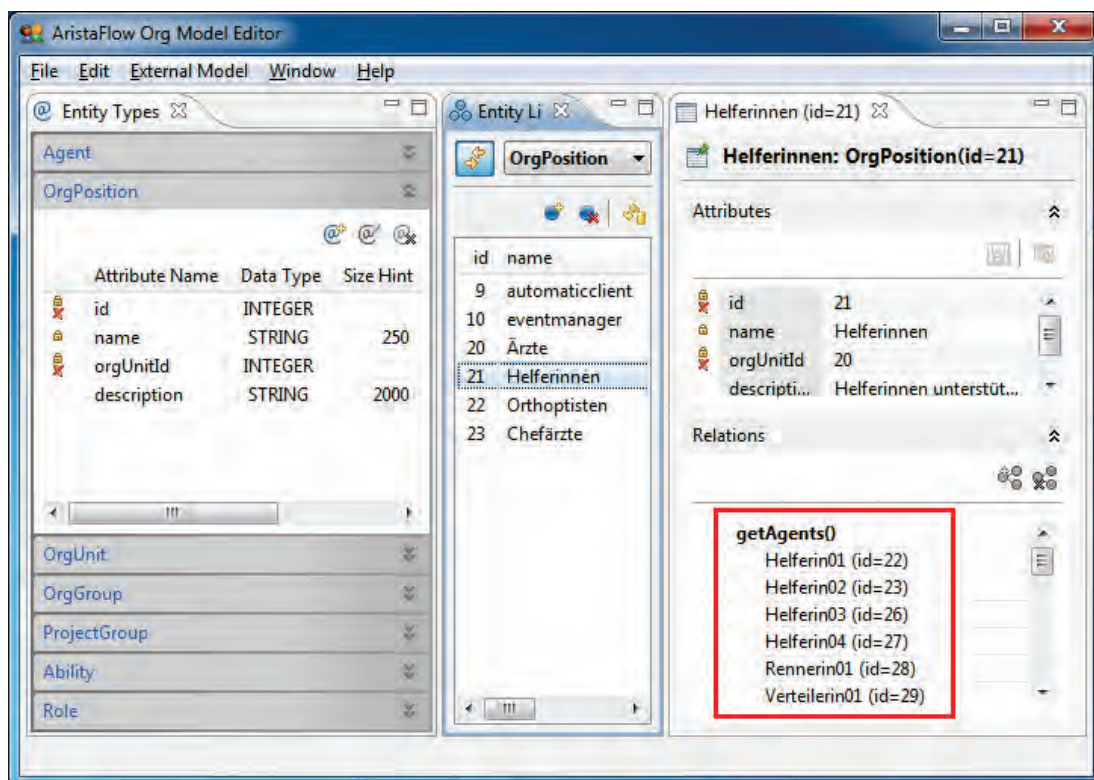


Abbildung 30: Umsetzung der Helferinnen als Organisationsposition im Org Model Editor

Abbildung 31 zeigt schließlich eine Helferin mit Benutzernamen „Helferin01“ mit ihrer Zuordnung (getOrgPositions) zur Organisationsposition „Helferinnen“. Außerdem sind alle angelegten Benutzer für die Augenarztpraxis in der Abbildung 31 ersichtlich. Die beiden Benutzer „automaticclient“ und „eventmanager“ werden von AristaFlow für die systemseitige automatische bzw. ereignisgesteuerte Prozessausführung benötigt.

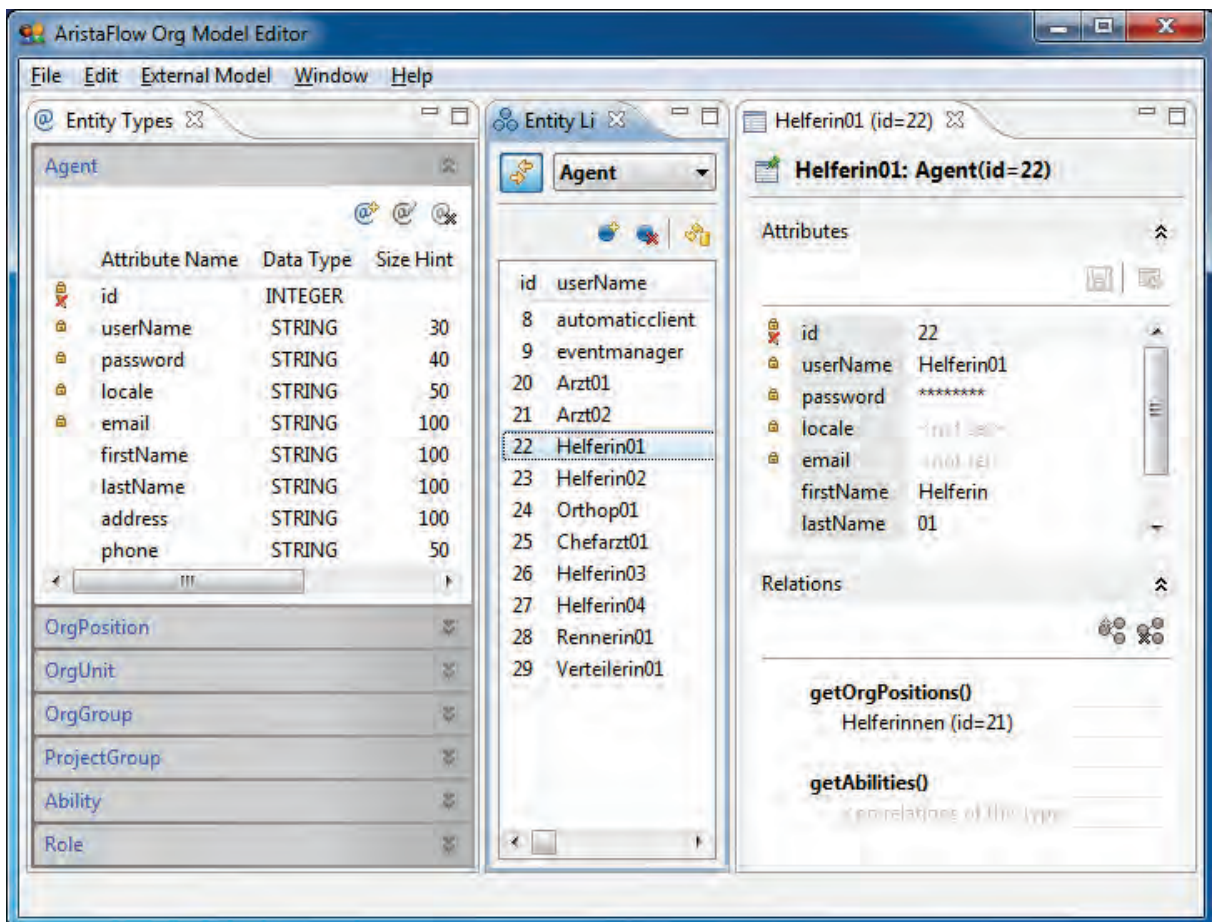


Abbildung 31: Umsetzung einer Helferin im AristaFlow Org Model Editor

Für die testweise Ausführung der Prozessvorlage ist das Organisationsmodell noch nicht notwendig, aber für die Produktivsetzung muss die Zuweisung der einzelnen Aktivitäten zum Organisationsmodell erfolgen.

Im nächsten Schritt der Umsetzung wird die Prozessvorlage inklusive dem Datenfluss mit dem Process Template Editor der AristaFlow BPM Suite modelliert und jeder Prozessschritt dem Organisationsmodell zugeordnet. Die erstellte Prozessvorlage für die „Patientendurchsteuerung“ ist in Abbildung 32 bis Abbildung 35 dargestellt. Nach erfolgreichem Test mit dem AristaFlow Test Client wird die Prozessvorlage für die produktive Nutzung freigegeben, so dass diese Vorlage im Standard Client der AristaFlow BPM Suite ausgeführt werden kann.

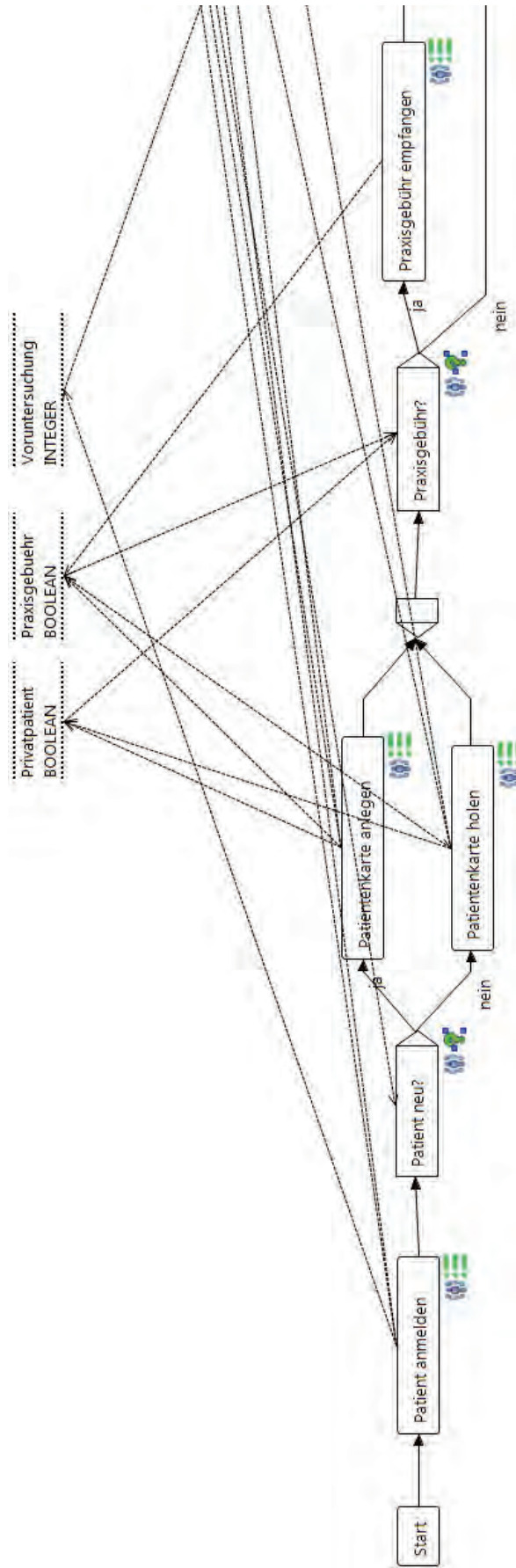


Abbildung 32: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 1 von 4)

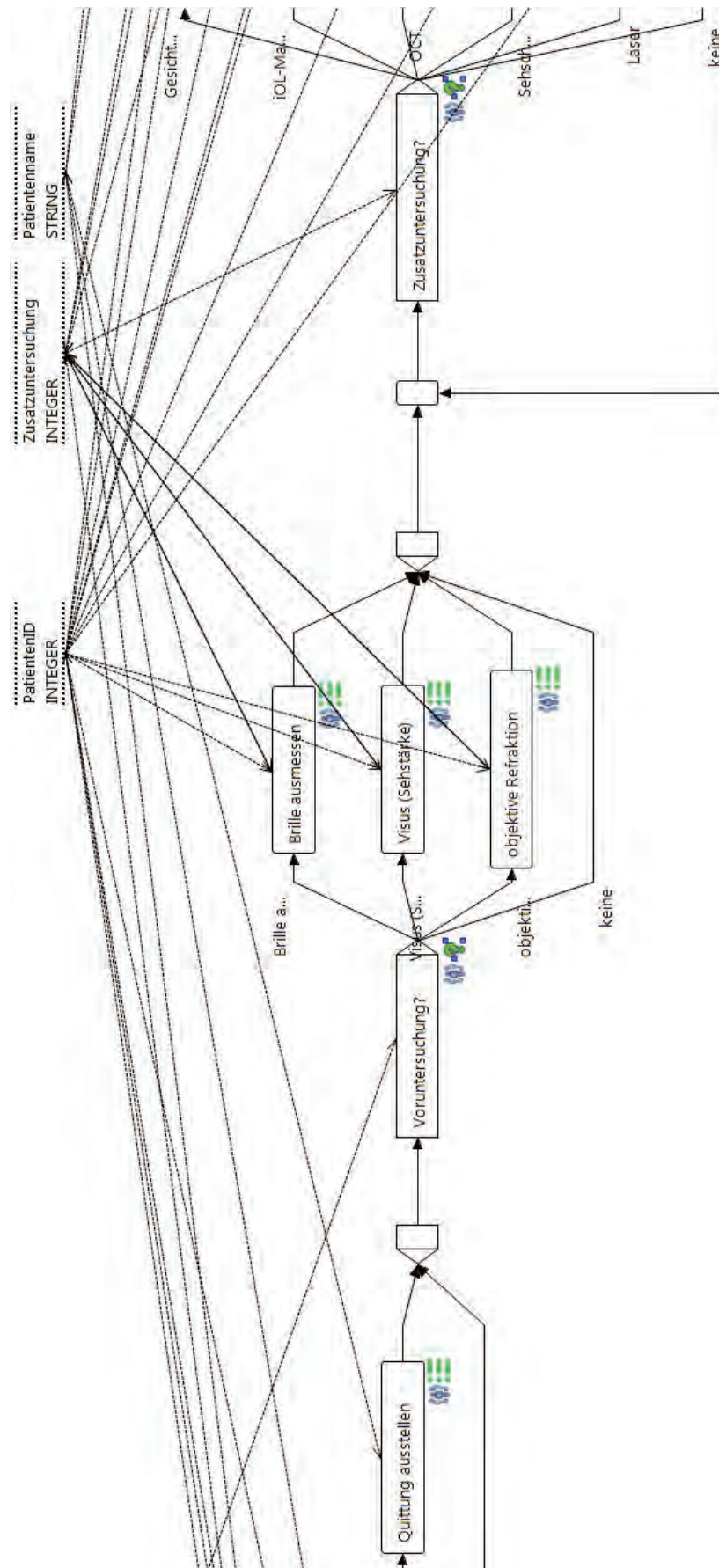


Abbildung 33: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 2 von 4)

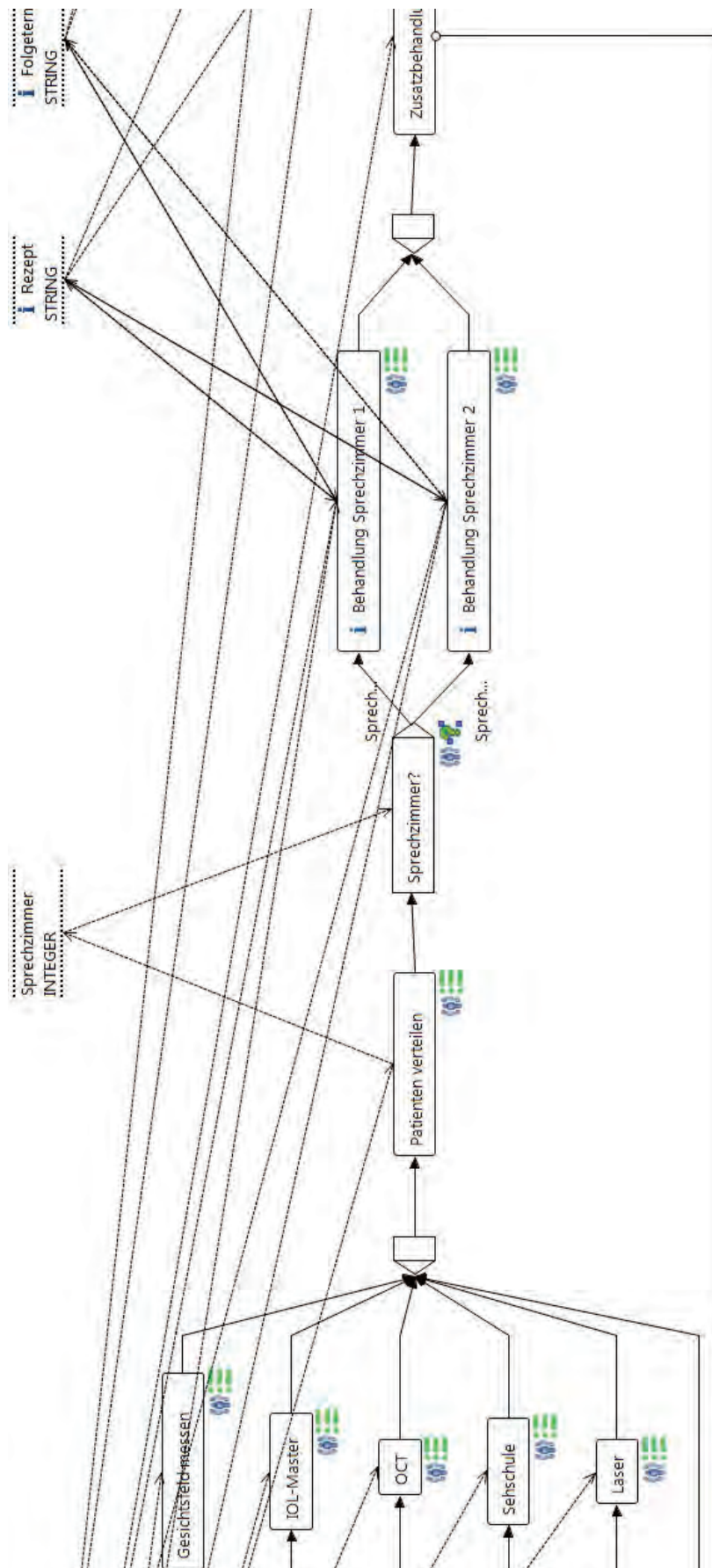


Abbildung 34: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 3 von 4)

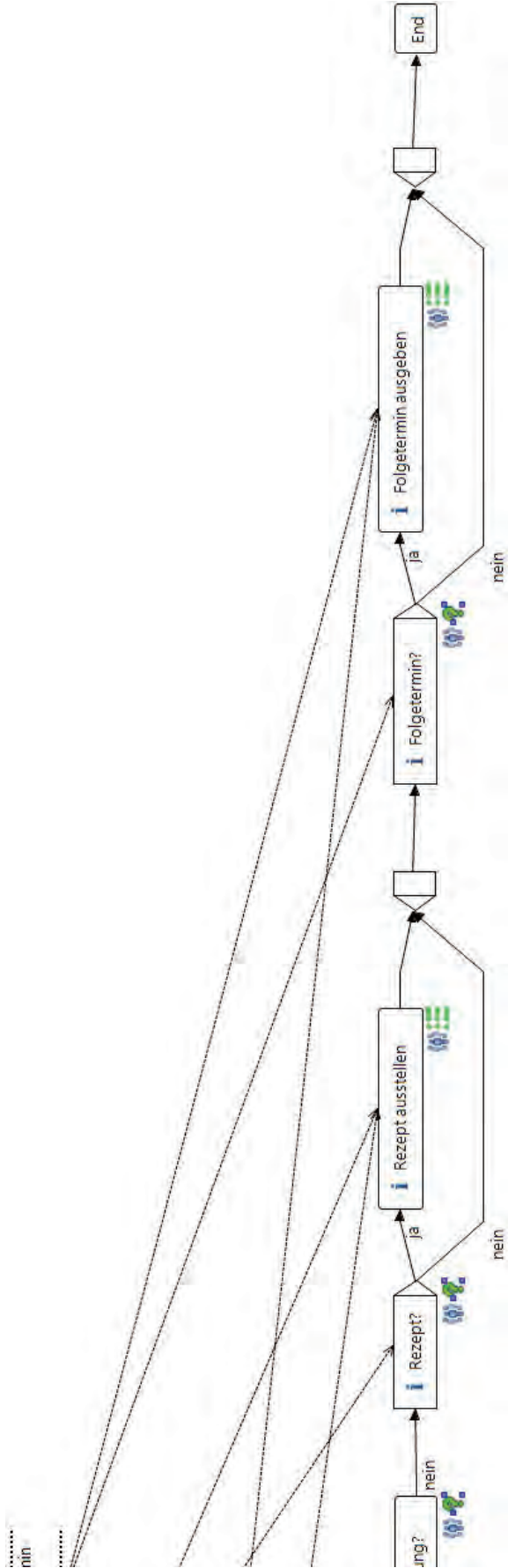


Abbildung 35: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 4 von 4)

Die Berechtigung zum Start einer produktiven Prozessvorlage wird im Vorlagenstatus („Template Status“) festgelegt. Im vorliegenden Szenario kann der Prozess von einer Helferin und dem Supervisor gestartet werden (Abbildung 36). Der Supervisor stellt den Administrator in AristaFlow dar.

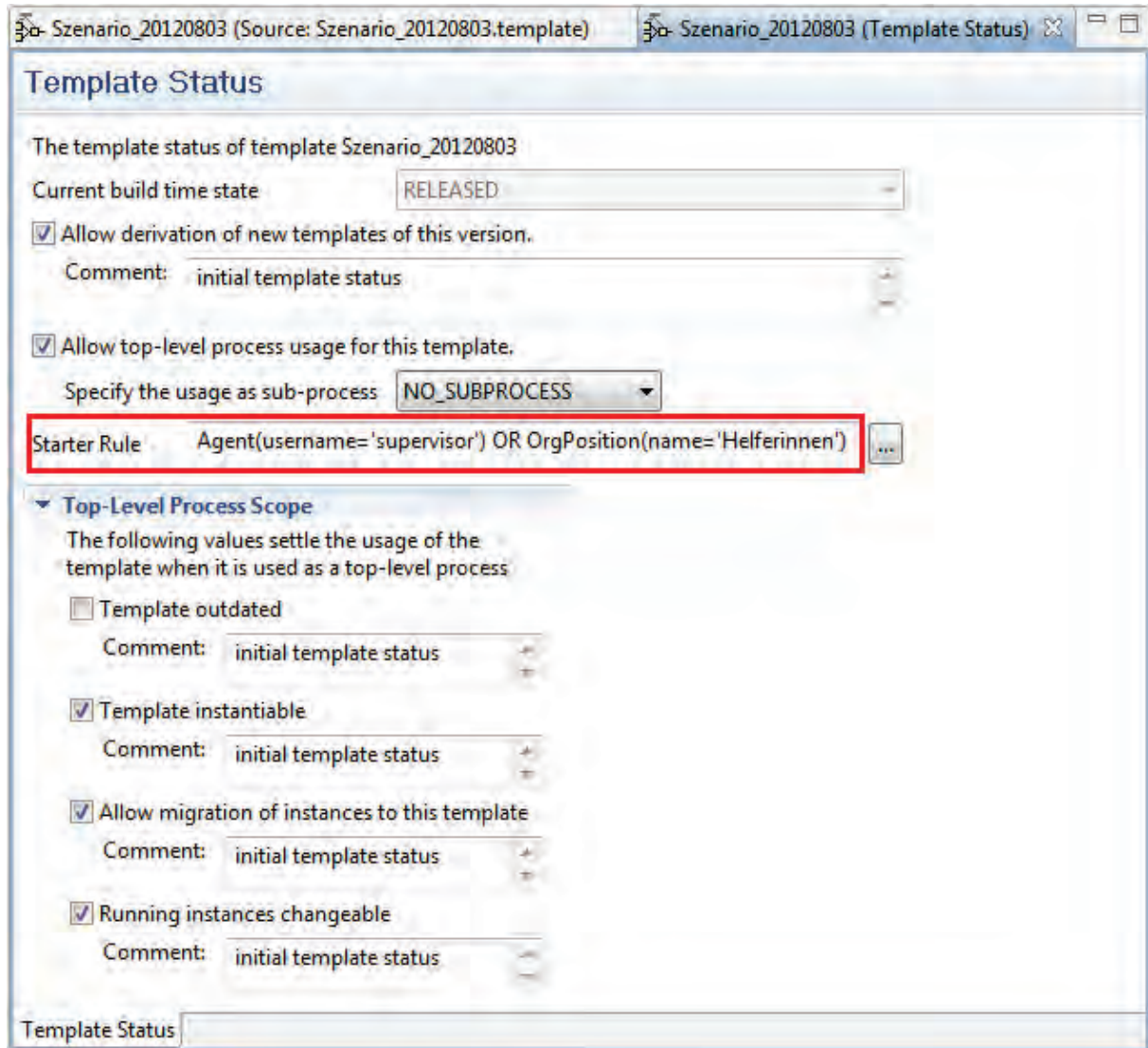


Abbildung 36: Vorlagenstatus der freigegebenen Prozessvorlage

4.3 Evaluierung der Möglichkeiten von AristaFlow

Der Aspekt der Prozessflexibilität zur Entwurfszeit wurde bereits in Abschnitt 3.2 bei der Modellierung mit AristaFlow erläutert.

Um die Möglichkeiten von AristaFlow hinsichtlich der Prozessflexibilität zur Ausführungszeit zu untersuchen, werden anhand des Beispielszenarios zunächst ein einfacher Prozessdurchlauf ausgeführt (Unterabschnitt 4.3.1) und anschließend zwei verschiedene Ausnahmesituationen erzeugt, die eine Ad-hoc-Anpassung notwendig machen (Unterabschnitte 4.3.2 und 4.3.3). Die Prozessprotokollierung folgt in Unterabschnitt 4.3.4.

Eine Prozessschemaevolution auf eine neue Prozessvorlage ist in der vorliegenden Version der AristaFlow BPM Suite nicht direkt über die Benutzeroberfläche ausführbar, so dass diese Funktionalität nicht berücksichtigt werden kann.

4.3.1 Prozessdurchlauf „Patientendurchsteuerung“

Der Prozess für die „Patientendurchsteuerung“ wird von einer Helferin gestartet, wenn ein Patient an der Anmeldung erscheint. Dazu meldet sie sich am AristaFlow-Klient an (hier: Helferin01), wählt die Prozessvorlage Szenario_20120803 aus und startet eine neue Prozessinstanz (Abbildung 37a). Der instanziierte Prozess ist in Abbildung 37b zu sehen.

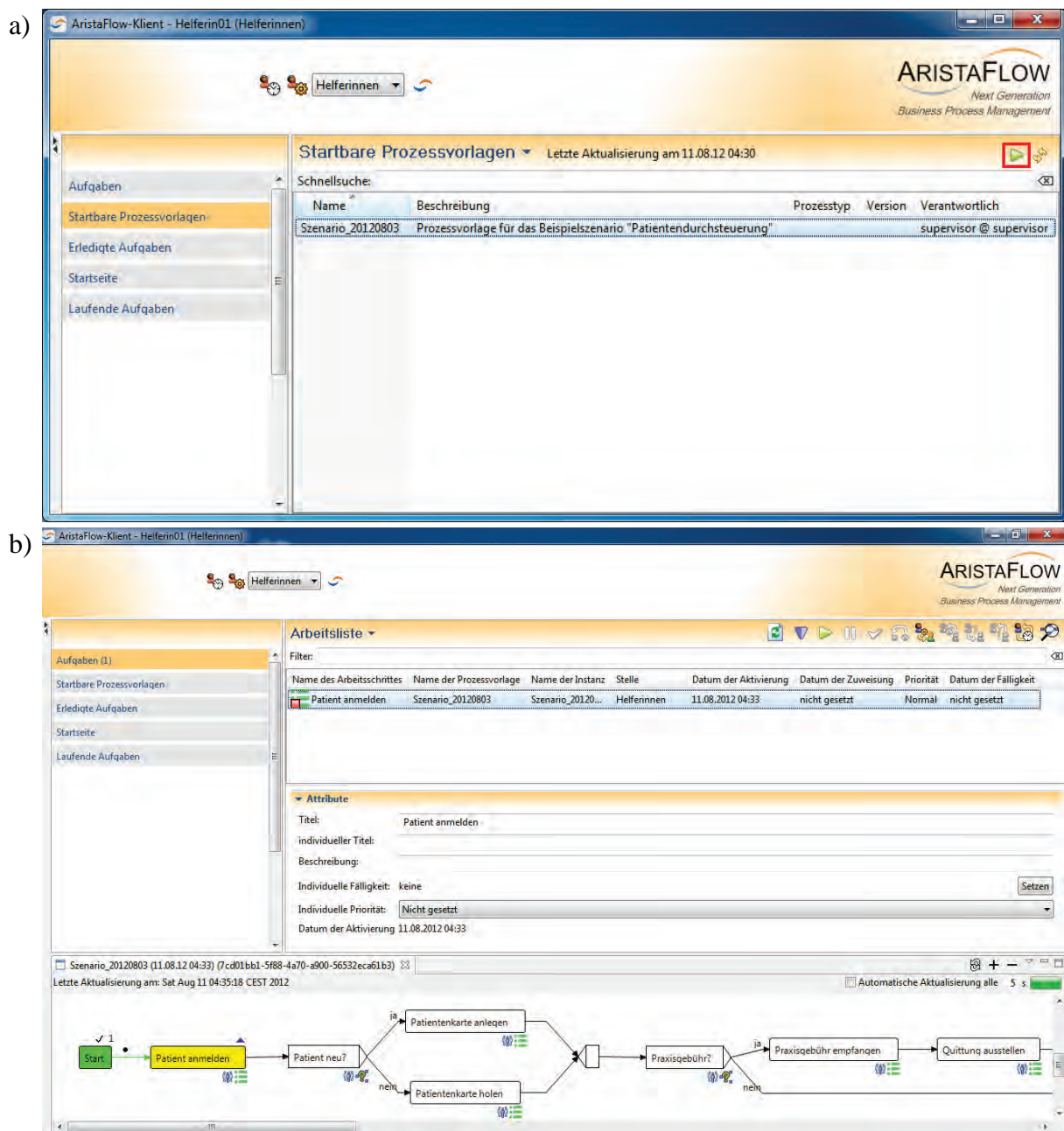


Abbildung 37: Auswahl und Instanziierung einer Prozessvorlage im Klient

Nachdem die Helferin01 den Patienten angenommen und die Vor- sowie Zusatzuntersuchung eingetragen hat (Abbildung 38a), kann der Prozess bis zum Prozessschritt „Voruntersuchung?“ von der Helferin01 ausgeführt werden. Die Verteilung auf die Voruntersuchungsräume nimmt dann die Rennerin vor. Daher steht bei der Helferin01 zu diesem Zeitpunkt keine Aufgabe mehr in ihrer Arbeitsliste an (Abbildung 38b).

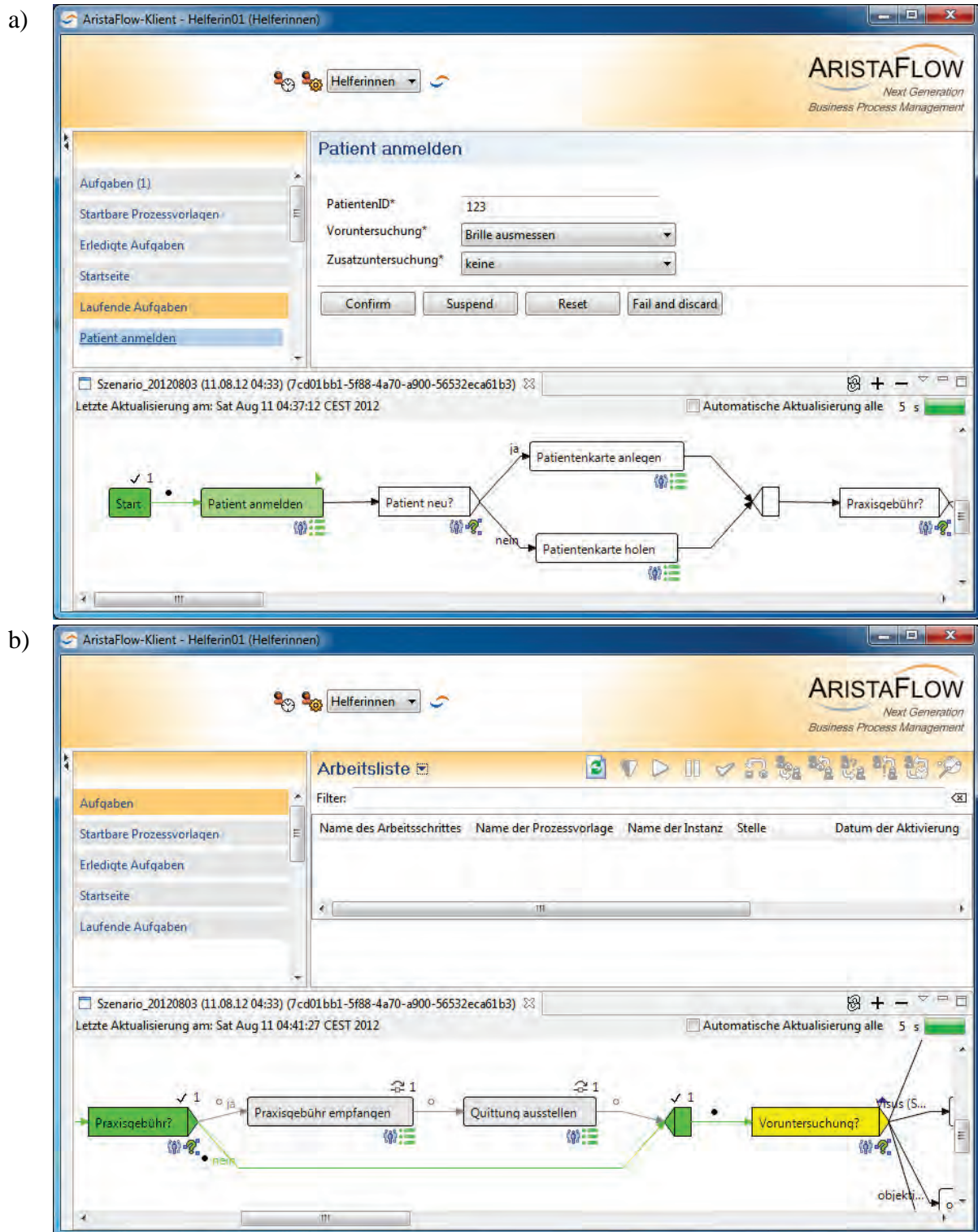


Abbildung 38: Bearbeitung der Arbeitsliste von Helferin01

Die in AristaFlow angemeldete Rennerin01 hat nun eine Aufgabe in ihrer Arbeitsliste erhalten (Abbildung 39a), da sie für diese Tätigkeit im Organisationsmodell vorgesehen ist. Gleiches gilt im weiteren Prozessverlauf bei Aktivierung der Aktivität „Patienten verteilen“ für die Verteilerin01, die die Patienten den beiden Ärzten zuweisen muss (Abbildung 39b).

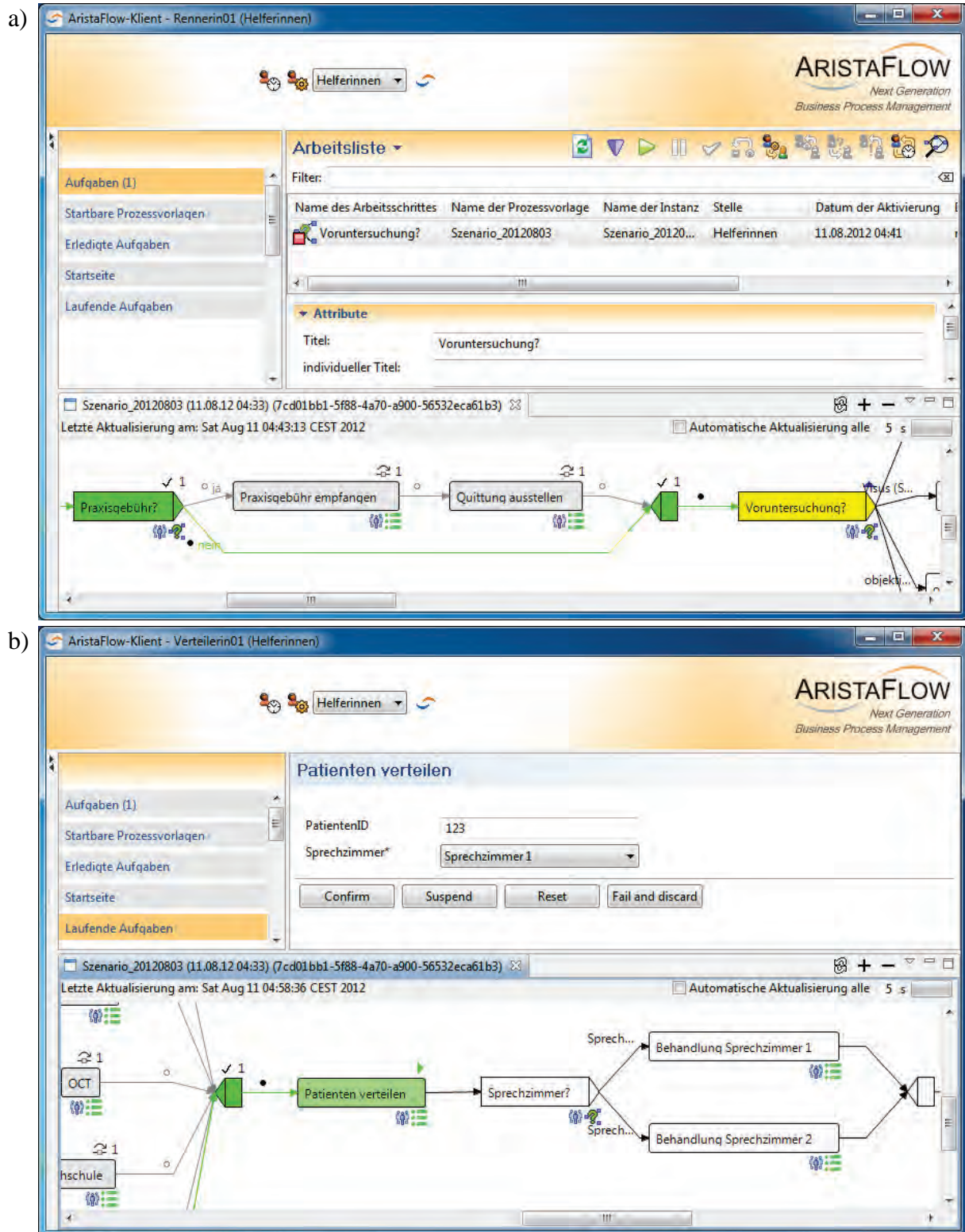


Abbildung 39: Wechsel der Zuständigkeiten im weiteren Prozessverlauf

Im Sprechzimmer 1 erfolgt schließlich die Behandlung des Patienten mit der ID 123 durch den Arzt01. Dieser kann noch einen Folgetermin vereinbaren, Medikamente verschreiben und eine weitere Zusatzuntersuchung anordnen (Abbildung 40a). Nach Ausstellung des Rezeptes terminiert die Prozessinstanz und die „Patientendurchsteuerung“ ist für diesen Patienten abgeschlossen (Abbildung 40b).

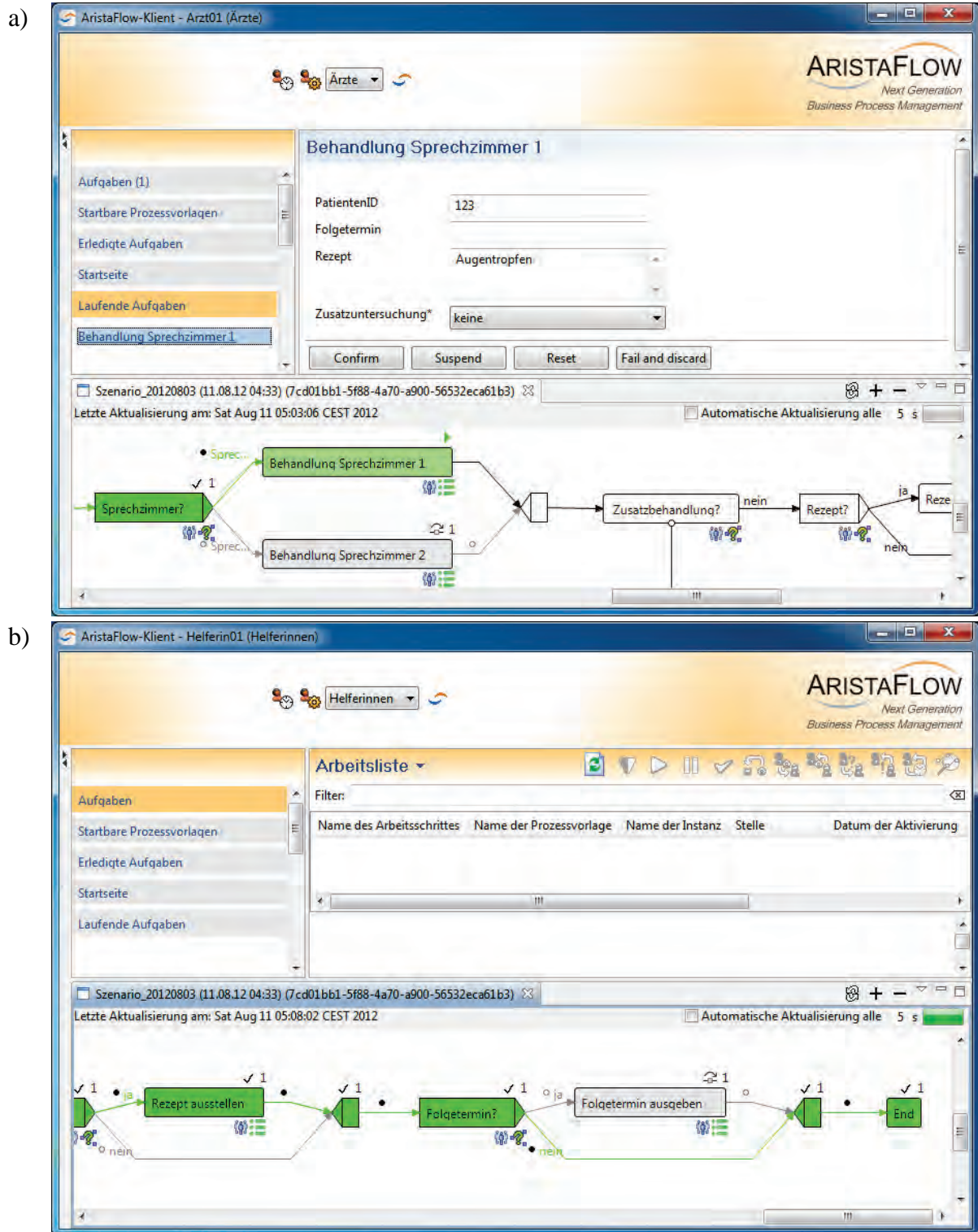


Abbildung 40: Behandlung durch den Arzt und Terminierung der Prozessinstanz

4.3.2 Erstes Ausnahmeszenario

Im ersten Ausnahmeszenario soll der Patient mit der ID 124 zur Voruntersuchung vorgelassen werden, jedoch steht die Rennerin gerade nicht zur Verfügung. Daher wird die laufende Prozessinstanz ad hoc angepasst, so dass auch die Helferin01 den Prozessschritt „Voruntersuchung?“ für den zuvor genannten Patienten ausführen kann.

Dazu wird die aktive Instanz im AristaFlow Monitoring von der Helferin01 ausgewählt und anschließend modifiziert. Abbildung 41 zeigt die Prozessinstanz für die „PatientenID“ 124.

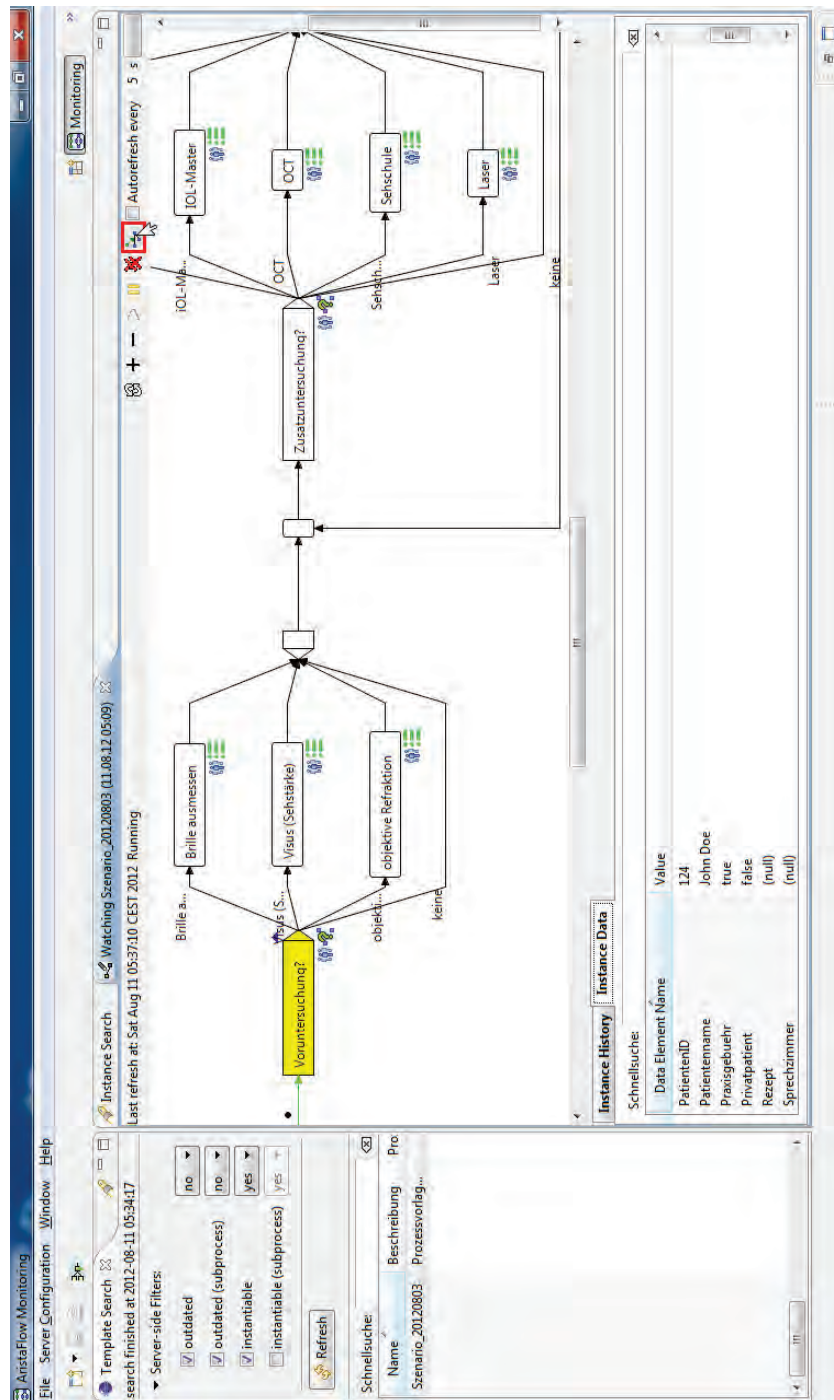


Abbildung 41: Auswahl einer laufenden Prozessinstanz im AristaFlow Monitoring

Nun wird für den Knoten „Voruntersuchung?“ die Regel für die Mitarbeiterzuordnung (Staff Assignment Rule) um die Helferin01 erweitert. Dies ist in Abbildung 42a und b dargestellt. Nach erfolgreicher Prüfung und Übernahme der Ad-hoc-Anpassung auf die laufende Instanz (siehe Abbildung 43) kann die Helferin01 die Zuweisung auf den Voruntersuchungsraum für die „PatientenID“ 124 aus ihrer Arbeitsliste heraus selbst vornehmen (siehe Abbildung 44).

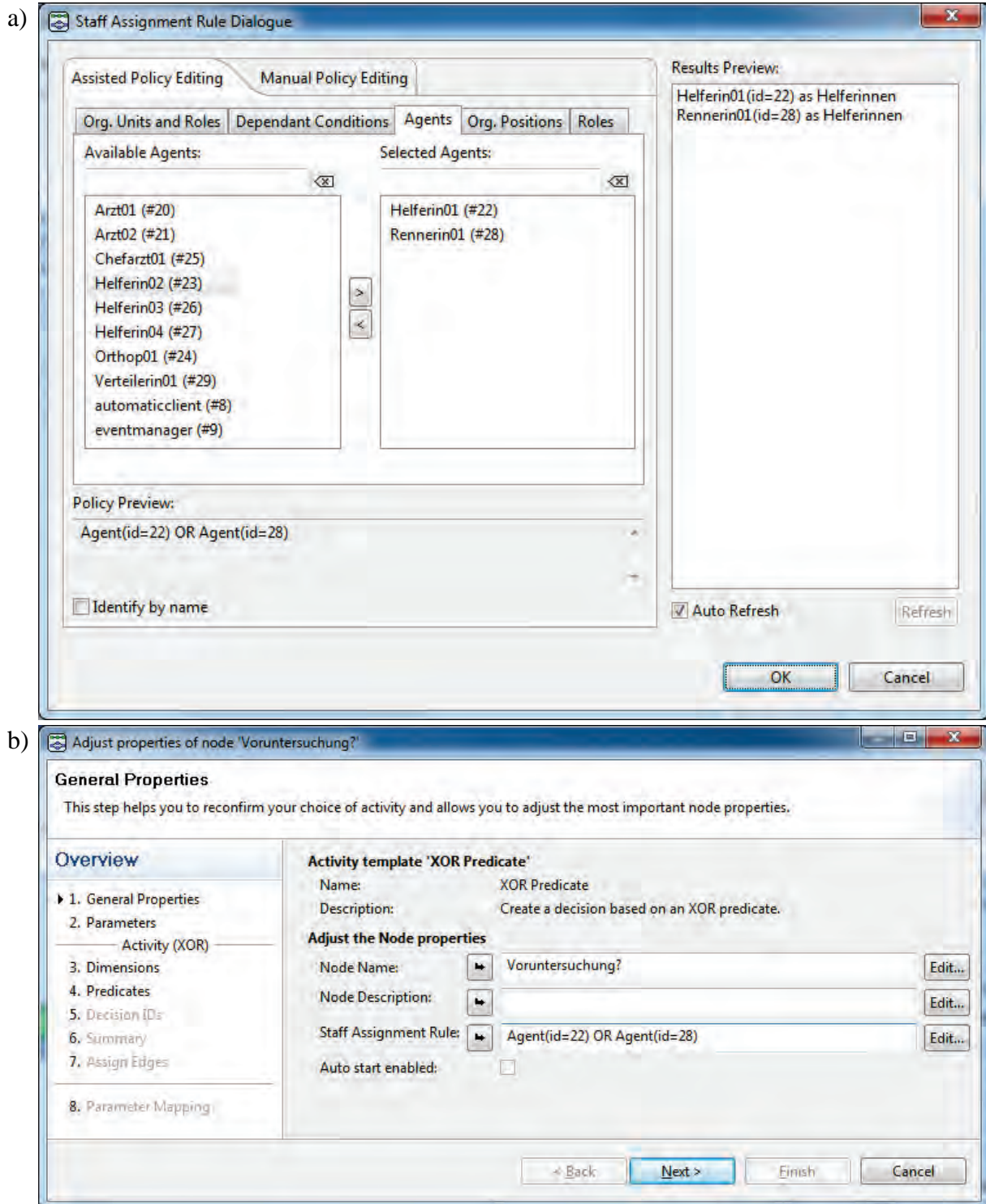


Abbildung 42: Ad-hoc-Anpassung der Regel für die Mitarbeiterzuordnung

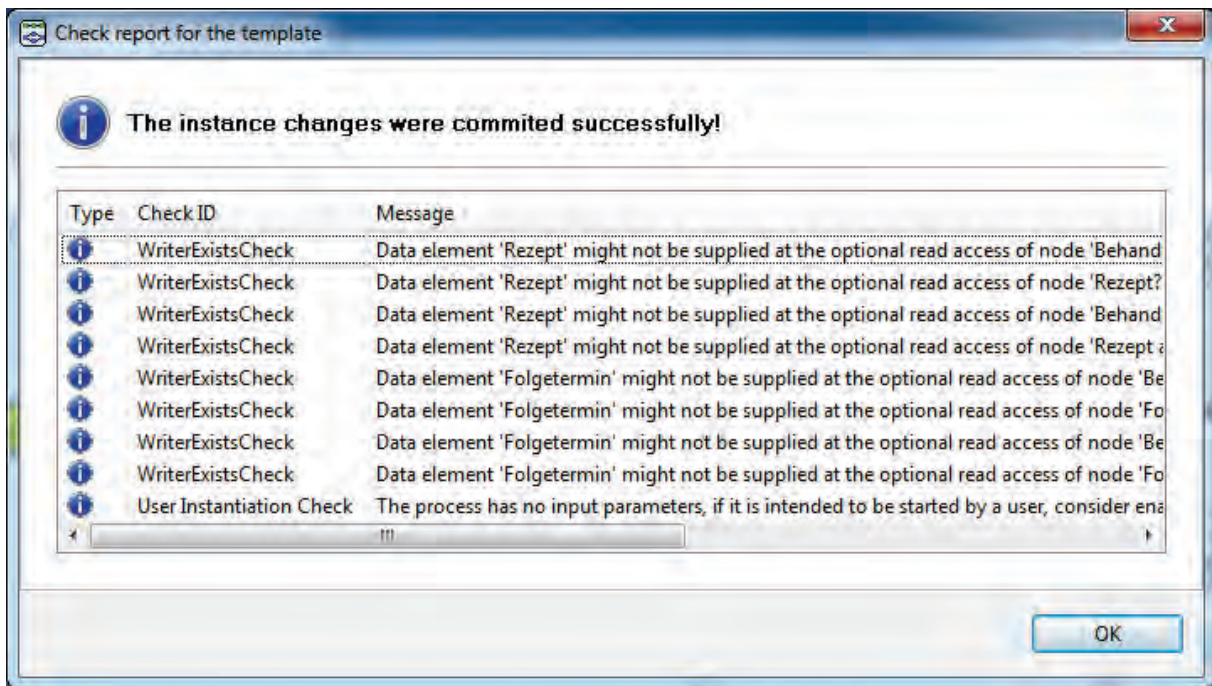


Abbildung 43: Ergebnisbericht bei Übernahme der Ad-hoc-Änderung

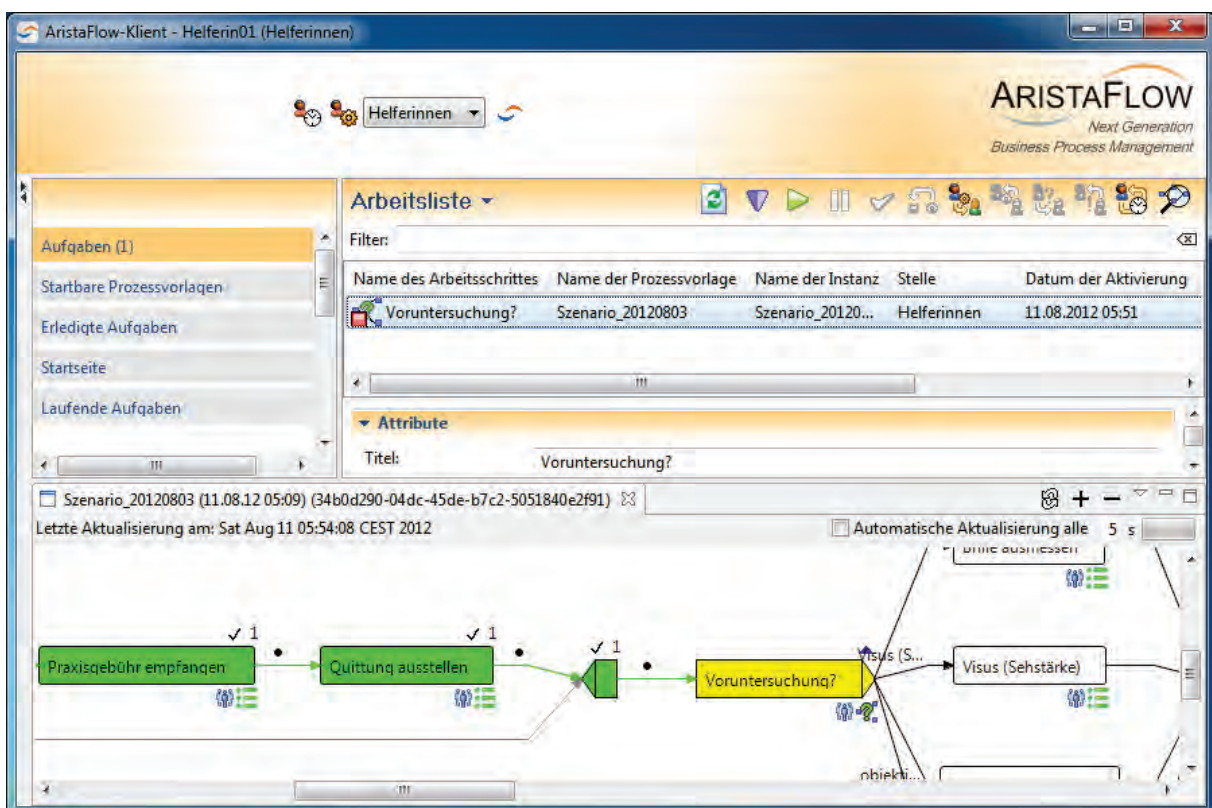


Abbildung 44: Arbeitsliste der Helferin01 nach Übernahme der Ad-hoc-Anpassung

4.3.3 Zweites Ausnahmeszenario

Im zweiten Ausnahmeszenario soll für den Patienten mit der ID 112 vor der Ausgabe des Rezeptes geprüft werden, ob die verschriebenen Medikamente evtl. in der Praxis verfügbar sind. Wenn dies der Fall ist, dann erfolgt eine direkte Ausgabe der Medikamente und das Rezept entfällt.

Diese Anpassung wird von der Helferin01 als Ad-hoc-Änderung auf die dazugehörige Prozessinstanz angewandt, die sich aktuell auf der Aktivität „Rezept ausstellen“ befindet (siehe Abbildung 45). Diese Aktivität ist lediglich aktiviert und wird noch nicht ausgeführt, so dass weitere Prozessschritte eingefügt werden können.

Die Auswahl, der zu ändernden Prozessinstanz, erfolgt wie im ersten Ausnahmeszenario im AristaFlow Monitoring durch die Helferin01, die in diesem Fall eine komplexere Anpassung an der Instanz durchzuführen hat. So werden weitere Aktivitäten für die Verfügbarkeitsprüfung und Ausgabe der Medikamente sowie eine zusätzliche XOR-Verzweigung für den Fall, dass die Prüfung positiv ausfällt, eingefügt. Das Ergebnis der Ad-hoc-Anpassung im AristaFlow Monitoring ist in Abbildung 46 dargestellt.

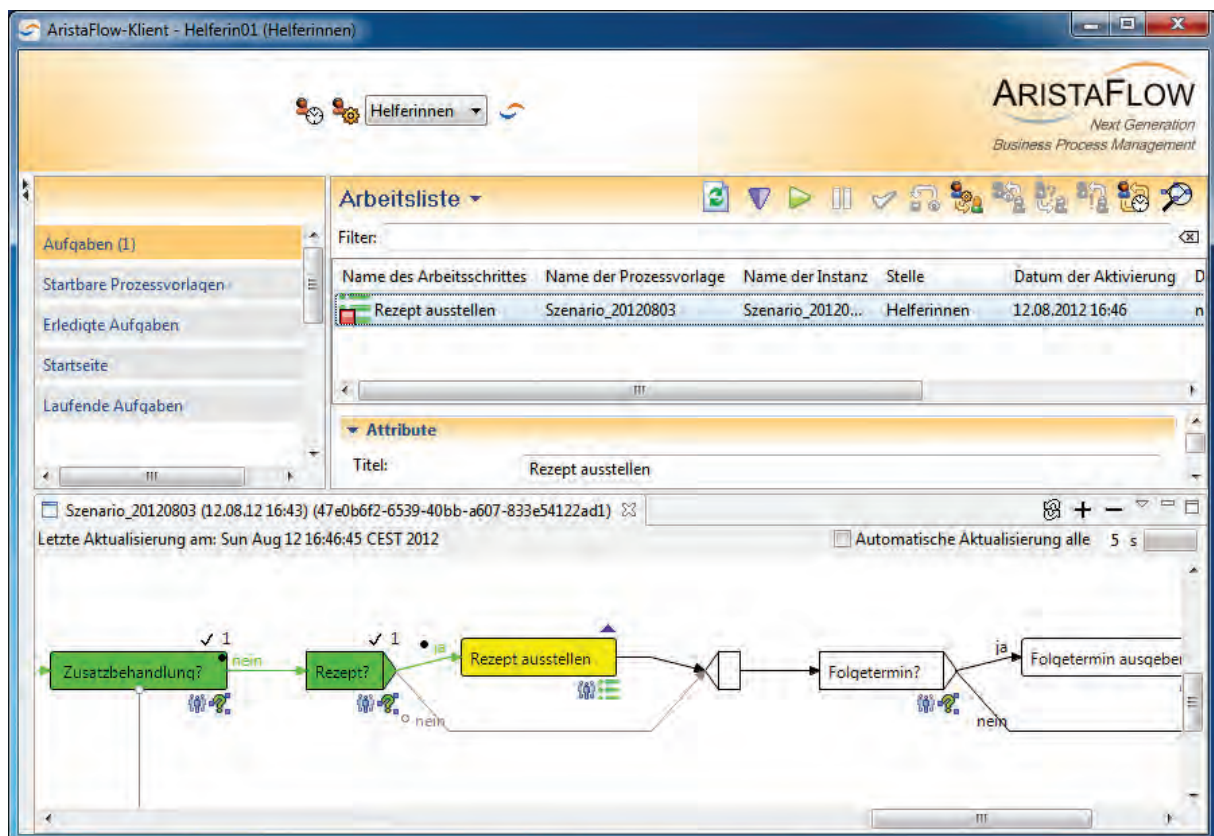


Abbildung 45: Prozessinstanz vor der Ad-hoc-Anpassung

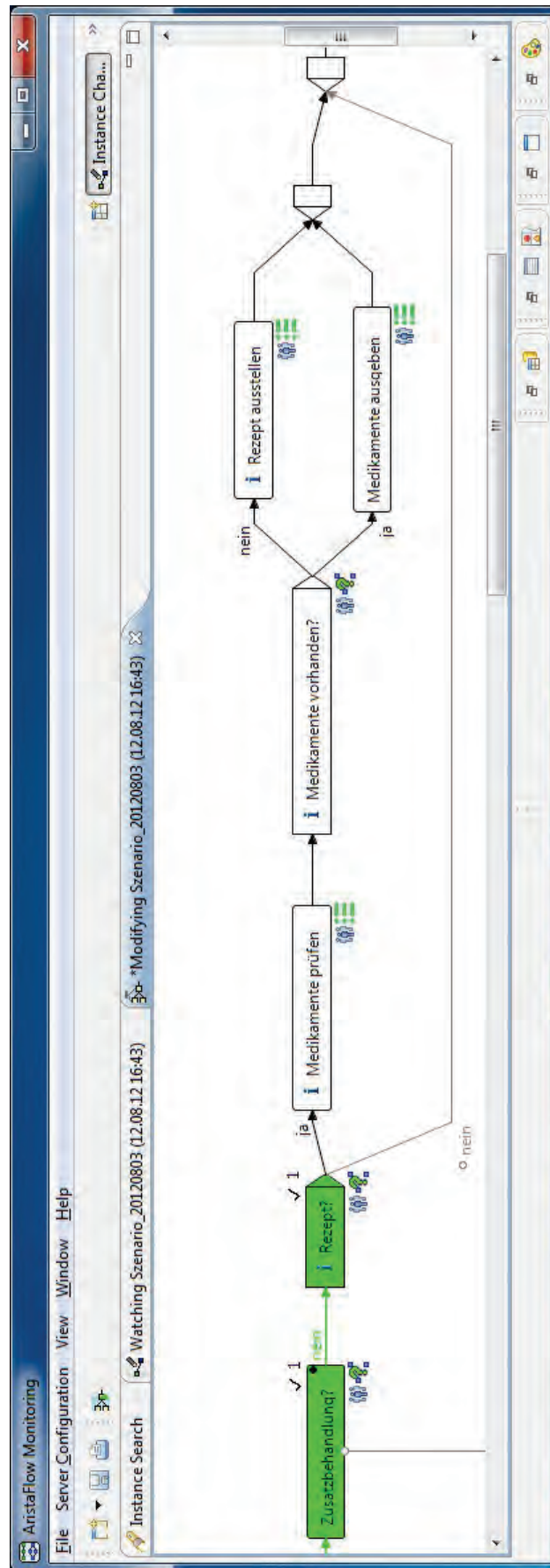


Abbildung 46: Durchgeführte Ad-hoc-Anpassung im AristaFlow Monitoring

Abschließend werden die Änderungen wieder auf die laufende Prozessinstanz propagiert, so dass der geänderte Ablauf im Klient weiter bearbeitet werden kann. Die Abbildung 47a und b zeigen den weiteren Prozessablauf, sofern die verschriebenen Augentropfen verfügbar sind.

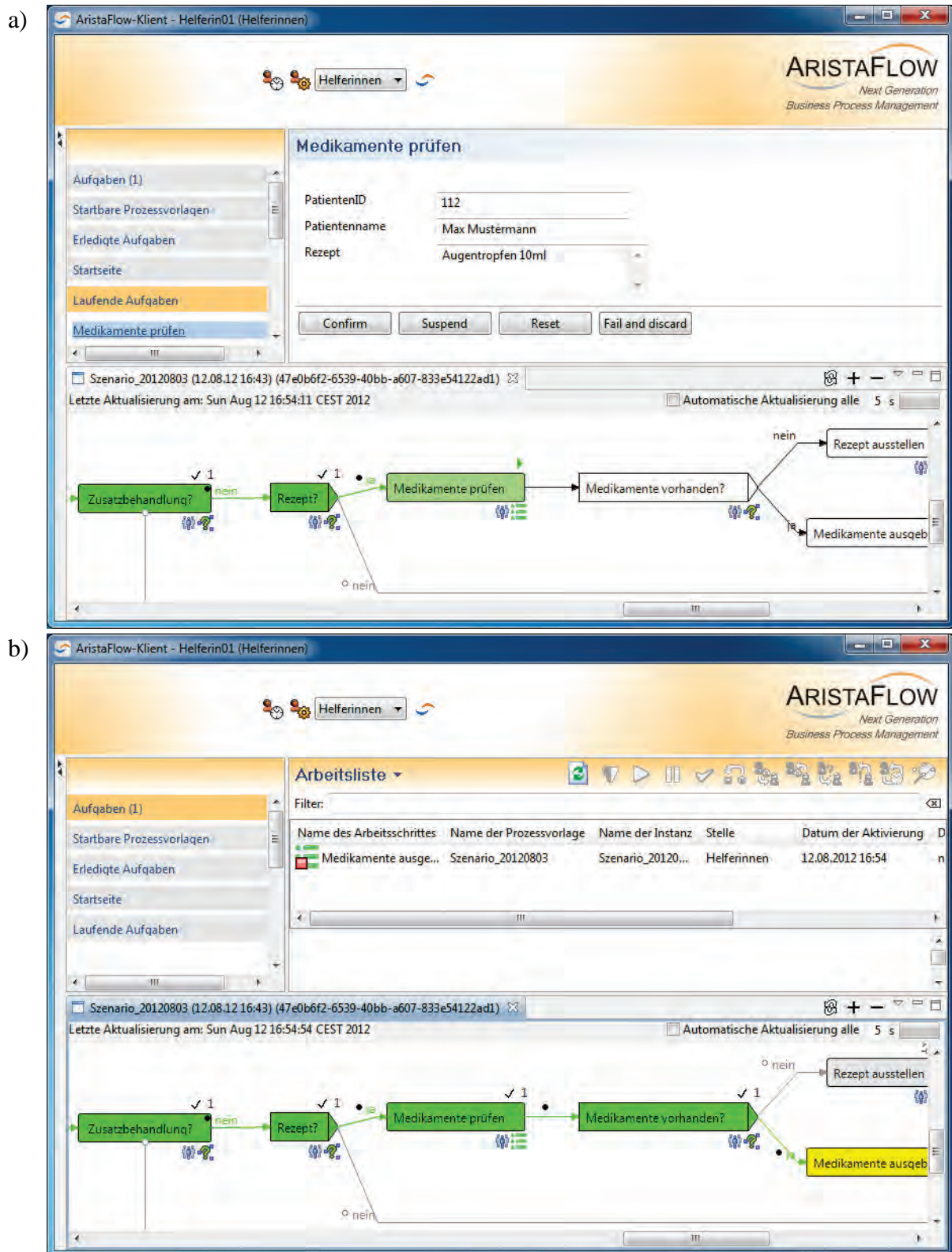


Abbildung 47: Möglicher Prozessablauf nach der Ad-hoc-Anpassung

4.3.4 Prozessprotokollierung

Die evtl. durchgeführten Ad-hoc-Änderungen und Prozesszustände für jede prozessierte Instanz können zu jeder Zeit im AristaFlow Monitoring nachvollzogen werden. Der tatsächliche Prozessverlauf wird dabei grafisch und die Instanz-Historie („Instance History“) tabellarisch dargestellt. Weiterhin können die Werte der Prozessinstanz („Instance Data“) eingesehen werden.

Abbildung 48 zeigt auszugsweise das Protokoll für das zuletzt ausgeführte Ausnahmeszenario aus Unterabschnitt 4.3.3. So ist unter anderem die Ad-hoc-Anpassung durch die Helferin01 im Prozessablauf und als Statusänderung (Zeile „INSTANCE_CHANGED“) protokolliert worden.

The screenshot displays the AristaFlow Monitoring application. The top part shows a process flow diagram with nodes: 'Rezept?', 'Medikamente prüfen', 'Medikamente vorhanden?', 'Rezept ausstellen', and 'Medikamente ausgeben'. The bottom part shows the 'Instance History' table with the following data:

timestamp	stateChange	nodeName	nodeID	iteration	agentID	agentOrgPositionID
2012-08-12 16:54:45.831	NODE_ACTIVATED		39	0		
2012-08-12 16:54:45.747	NODE_FINISHED	Medikamente vorhanden?	36	0	automaticclient (8)	automaticclient (9)
2012-08-12 16:54:45.648	NODE_STARTED	Medikamente vorhanden?	36	0	automaticclient (8)	automaticclient (9)
2012-08-12 16:54:44.597	NODE_ACTIVATED		36	0		
2012-08-12 16:54:44.515	NODE_FINISHED	Medikamente prüfen	38	0	Helferin01 (22)	Helferinnen (21)
2012-08-12 16:54:06.468	NODE_STARTED	Medikamente prüfen	38	0	Helferin01 (22)	Helferinnen (21)
2012-08-12 16:53:10.922	NODE_ACTIVATED		38	0		
2012-08-12 16:53:10.828	INSTANCE_RESUMED				Helferin01 (22)	Helferinnen (21)
2012-08-12 16:52:53.192	INSTANCE_CHANGED				Helferin01 (22)	Helferinnen (21)
2012-08-12 16:47:32.291	INSTANCE_SUSPENDED				Helferin01 (22)	Helferinnen (21)
2012-08-12 16:46:16.688	NODE_ACTIVATED		28	0		
2012-08-12 16:46:16.609	NODE_FINISHED	Rezept?	26	0	automaticclient (8)	automaticclient (9)
2012-08-12 16:46:16.538	NODE_STARTED	Rezept?	26	0	automaticclient (8)	automaticclient (9)
2012-08-12 16:46:15.597	NODE_ACTIVATED		26	0		

Abbildung 48: Protokoll zu einer Prozessinstanz in AristaFlow

5 Zusammenfassung und Ausblick

Die vorliegende Arbeit hat gezeigt, dass die Begriffe BPM und WFM fließend ineinander gehen. Die Differenzierung liegt lediglich in der Analyse und Optimierung von Prozessen. Die darauf aufbauenden Systeme (PMS bzw. WFMS) stellen durch die aktive Koordination von betrieblichen Aufgaben, durch die Automatisierung von Prozessen, durch verbesserte Nachvollziehbarkeit einzelner Prozessausführungen sowie durch leichtere Einhaltung von Compliance-Vorgaben gegenüber manuell ausgeführten Abläufen einen wichtigen Beitrag für die Unterstützung, Kontrolle und Steuerung von Unternehmensprozesse dar. Jedoch hängt der Erfolg eines PMS auch mit der Möglichkeit zusammen, einfach und effizient Prozessabläufe zu gestalten sowie bei Bedarf bereits laufende Prozesse flexibel an neue Gegebenheiten ohne Beeinträchtigung der Korrektheit und Konsistenz für den weiteren Ablauf anpassen zu können. Die für ein PMS notwendigen technischen Voraussetzungen bezüglich Prozessflexibilität zur Entwurfs- und Ausführungszeit sowie hinsichtlich der Prozessschemaevolution wurden dargestellt und mit Hilfe von AristaFlow an einem Beispielszenario aus der Krankenhausdomäne untersucht.

Die Stärken von ADEPT2 liegen insbesondere in der serviceorientierten Architektur, die es ermöglicht, ein eigenes PMS mit den in ADEPT2 verfügbaren Komponenten zu entwickeln, das alle Flexibilitätsaspekte unter einer Oberfläche vereint. Die AristaFlow BPM Suite hat mit ihrer zugrundeliegenden ADEPT2-Technologie gezeigt, wie die zuvor genannten Flexibilitätsaspekte teilweise in einem PMS verwirklicht wurden. Es ist mit Sicherheit nur eine Frage der Zeit, bis auch die Prozessschemaevolution in einer zukünftigen Version der AristaFlow-Software direkt integriert sein wird. Bereits am Markt etablierte PMS wie z.B. von Oracle oder IBM haben sich dem Thema Prozessmigration ebenfalls angenommen und zeigen integrierte Ansätze in ihrer Software auf Basis von BPEL und BPMN.

Wenn diese technologischen Ansätze dem Anwender bzw. Prozessmodellierer in leicht bedienbarer Art und Weise zur Verfügung gestellt werden, so wird es in Zukunft leichter sein, auch Unternehmen oder Branchen mit hohen Ansprüchen an die Prozessflexibilität für den Einsatz eines PMS überzeugen zu können.

Abkürzungsverzeichnis

ADEPT	Application Development based on Encapsulated pre-modeled Process Templates
API	Application-Programming-Interface
ARIS	Architektur integrierter Informationssysteme
BPEL	Business-Process-Execution-Language
BPM	Business-Process-Management
BPMN	Business-Process-Model and Notation
BPMS	Business-Process-Management-System
BT	Buildtime
EPK	Ereignisgesteuerte Prozesskette
IOL	Intraokularlinse
OCT	Optical-Coherence-Tomography (dt. optische Kohärenztomografie)
PMS	Prozessmanagementsystem
RT	Runtime
WFM	Workflow-Management
WFMS	Workflow-Management-System

Abbildungsverzeichnis

Abbildung 1: BPM-Lebenszyklus zum Vergleich von WFM und BPM nach [AHW03]	6
Abbildung 2: Trennung von Prozesslogik und Anwendungscode in Anlehnung an [Rind04]..	7
Abbildung 3: Flexibilitätsaspekte.....	8
Abbildung 4: Modellierungsfehler – unterbrochener Kontrollfluss.....	9
Abbildung 5: Modellierungsfehler – AND mit XOR abgeschlossen.....	9
Abbildung 6: Datenflüsse zwischen Anwendungsfunktionen	10
Abbildung 7: Ad-hoc-Anpassung einer laufenden Prozessinstanz	11
Abbildung 8: Modifikation von Prozessschema S zu S'	13
Abbildung 9: Beispiele von unverzerrten Prozessinstanzen	13
Abbildung 10: Beispiele von verzerrten Prozessinstanzen	14
Abbildung 11: Beispiel einer verzerrten Prozessinstanz mit Überlappung.....	15
Abbildung 12: Grobarchitektur von ADEPT2 nach [Reic08].....	18
Abbildung 13: ADEPT-Prozess-Meta-Modell aus [Dada09]	19
Abbildung 14: Initiale Prozessvorlage im AristaFlow Process Template Editor.....	20
Abbildung 15: Kontextabhängige Änderungsoperationen in AristaFlow	21
Abbildung 16: Fehlermeldung eines inkonsistenten Datenflusses.....	22
Abbildung 17: Beispiel eines Prozessmodells für eine spätere Ad-hoc-Anpassung.....	23
Abbildung 18: Versuch einer Ad-hoc-Anpassung an zwei Prozessinstanzen.....	24
Abbildung 19: Prozessinstanz (a) nach Einfügen des zusätzlichen Prozessschrittes.....	25
Abbildung 20: Grafische Oberfläche des Oracle Jdevelopers	27
Abbildung 21: Teilfenster für aufgetretene Fehler und Meldungen	27
Abbildung 22: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 1 von 7).....	30
Abbildung 23: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 2 von 7).....	31
Abbildung 24: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 3 von 7).....	32
Abbildung 25: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 4 von 7).....	32
Abbildung 26: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 5 von 7).....	33
Abbildung 27: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 6 von 7).....	33
Abbildung 28: EPK des Prozessablaufs „Patientendurchsteuerung“ (Teil 7 von 7).....	34
Abbildung 29: Organigramm für die Augenarztpraxis	35
Abbildung 30: Umsetzung der Helferinnen als Organisationsposition im Org Model Editor.	35
Abbildung 31: Umsetzung einer Helferin im AristaFlow Org Model Editor	36
Abbildung 32: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 1 von 4).....	37
Abbildung 33: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 2 von 4).....	38
Abbildung 34: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 3 von 4).....	39
Abbildung 35: AristaFlow-Prozessvorlage „Patientendurchsteuerung“ (Teil 4 von 4).....	40
Abbildung 36: Vorlagenstatus der freigegebenen Prozessvorlage.....	41
Abbildung 37: Auswahl und Instanziierung einer Prozessvorlage im Klient	42
Abbildung 38: Bearbeitung der Arbeitsliste von Helferin01	43
Abbildung 39: Wechsel der Zuständigkeiten im weiteren Prozessverlauf	44

Abbildung 40: Behandlung durch den Arzt und Terminierung der Prozessinstanz.....	45
Abbildung 41: Auswahl einer laufenden Prozessinstanz im AristaFlow Monitoring.....	46
Abbildung 42: Ad-hoc-Anpassung der Regel für die Mitarbeiterzuordnung	47
Abbildung 43: Ergebnisbericht bei Übernahme der Ad-hoc-Änderung	48
Abbildung 44: Arbeitsliste der Helferin01 nach Übernahme der Ad-hoc-Anpassung	48
Abbildung 45: Prozessinstanz vor der Ad-hoc-Anpassung.....	49
Abbildung 46: Durchgeführte Ad-hoc-Anpassung im AristaFlow Monitoring.....	50
Abbildung 47: Möglicher Prozessablauf nach der Ad-hoc-Anpassung	51
Abbildung 48: Protokoll zu einer Prozessinstanz in AristaFlow	52

Tabellenverzeichnis

Tabelle 1: Legende der Symbole.....	11
Tabelle 2: Software-Komponenten der AristaFlow BPM Suite.....	17
Tabelle 3: Vergleich der AristaFlow BPM Suite mit der Oracle BPM Suite	28

Literaturverzeichnis

- [AHW03] van der Aalst, W.M.P.; ter Hofstede, A.H.M.; Weske, M.: Business Process Management: A Survey. Business Process Management, Volume 2678 of Lecture Notes in Computer Science, S. 1-12, Springer-Verlag, Berlin, 2003.
- [Aris12] AristaFlow – AristaFlow BPM Suite. http://www.aristaflow.com/AristaFlow_BPM-Suite.html, Abruf am 2012-06-03.
- [Bach10] Bachmeier, A.: Entwurf und Implementierung eines Prozesses aus der Verwaltung am Beispiel einer Reisekostenabrechnung. Universität Ulm, Bachelorarbeit, 2010.
- [Cim12] CIM Aachen. http://www.cim-aachen.de/operations_prozess.php, Abruf am 2012-06-10.
- [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.
- [Dada09] Dadam, P.; Reichert, M.; Rinderle-Ma, S.; Göser, K.; Kreher, U.; Jurisch, M.: Von ADEPT zur AristaFlow BPM Suite – Eine Vision wird Realität: "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen. Entwicklungsmethoden für Informationssysteme und deren Anwendung (EMISA) Forum, 29(1), S. 9-28, 2009.
- [Dada10] Dadam, P.: Business Process Management. Vorlesung Wintersemester 2010/11 an der Universität Ulm, Institut für Datenbanken und Informationssysteme, 2010.
- [DaRe09] Dadam P.; Manfred 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, 23(2), S. 81-98, 2009.
- [DiSc05] Dietrich, E.; Schulze, A.: Statistische Verfahren zur Maschinen- und Prozessqualifikation. Carl Hanser Verlag, München Wien, 5., aktualisierte Auflage 2005.
- [DRK00] Dadam, P.; Reichert, M.; Kuhn, K.: Clinical Workflows – The Killer Application for Process-oriented Information Systems? Proceedings International Conference on Business Information Systems, BIS 2000, 4th International Conference, Poznan, Poland, 2000, S. 36-59, 2000.
- [DRR04] Dadam, P.; Reichert, M.; Rinderle, S.: Von funktionsorientierten zu prozessorientierten Informationssystemen – Herausforderungen und Lösungsansätze. In: Keynote (P. Dadam), Tagungsband 17. Deutsche Oracle-Anwender-Konferenz, Mannheim, November 2004
- [DRR11] Dadam, P.; Reichert, M.; Rinderle-Ma, S.: Prozessmanagementsysteme. Informatik Spektrum, 34(4), S. 364-376, 2011.
- [Gab12] Gabler Verlag (Herausgeber), Gabler Wirtschaftslexikon, Stichwort: Robustheit, online im Internet: <http://wirtschaftslexikon.gabler.de/Archiv/57696/robustheit-v7.html>, Abruf am 2012-06-10.

- [HaNe09] Hansen, H.R.; Neumann, G.: Wirtschaftsinformatik 1 – Grundlagen und Anwendungen. UTP, Stuttgart, 10. Auflage, 2009.
- [Jen02] Jen, E.: Stable or robust? What's the difference? Santa Fe Institute, 2002.
- [Klep08] Kleppmann, W.: Taschenbuch Versuchsplanung. Produkte und Prozesse optimieren. Carl Hanser Verlag, München Wien, 5., überarbeitete Auflage 2008.
- [KRRD09] Kreher, U.; Reichert, M.; Rinderle-Ma, S.; Dadam, P.: Speichereffiziente Repräsentation instanzspezifischer Änderungen in Prozess-Management-Systemen. Technical Report. Universität Ulm, Ulm, 2009.
- [LRR04] Lauer, M.; Rinderle, S.; Reichert M.: Repräsentation von Schema- und Instanzobjekten in adaptiven Prozess-Management-Systemen. Proceedings Workshop Geschäftsprozessorientierte Architekturen (Informatik '04), Lecture Notes in Informatics (LNI) P-51, Germany, September 2004, S. 555-560, 2004.
- [Mert09] Mertens, P.: Integrierte Informationsverarbeitung 1: Operative Systeme in der Industrie. Gabler, Wiesbaden, 17. Auflage, 2009.
- [Nahl05] Nahler, M.: Semantische Konflikte in adaptiven Prozess-Management-Systemen. Universität Ulm, Diplomarbeit, 2005.
- [Orac12] Oracle: Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management, 11g Release 1 (11.1.1.6.2), E15176-09, 2012.
- [ReDa98] Reichert, M.; Dadam, P.: ADEPTflex – Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems, Kluwer Academic Publishers, 10(2), S. 93-129, 1998.
- [Reic00] Reichert, M.: Dynamische Ablaufänderungen in Workflow-Management-Systemen. Universität Ulm, Dissertation, 2000.
- [Reic08] Reichert, M.; Dadam, P.; Jurisch, M.; Kreher, U.; Göser, K.; Lauer, M.: Architectural Design of Flexible Process Management Technology. In: Ulmer Informatik Berichte Nr. 2008-02, Universität Ulm. Fakultät für Ingenieurwissenschaften und Informatik, Ulm, 2008.
- [Reut11] Reuter, C.: Modellierung und dynamische Adaption klinischer Pfade auf Basis semantischer Prozessfragmente. Fraunhofer Institut Software und Systemtechnik, Dortmund, Dissertation, 2011.
- [Rind04] Rinderle, S.B.: Schema Evolution in Process Management Systems. Universität Ulm, Dissertation, 2004.
- [Rind09] Rinderle-Ma, S.: Data Flow Correctness in Adaptive Workflow Systems. Entwicklungsmethoden für Informationssysteme und deren Anwendung (EMISA) Forum, 29(2), S. 25-35, 2009.
- [RRD04] Rinderle, S.; Reichert, M.; Dadam, P.: Flexible Support of Team Processes by Adaptive Workflow Systems. Distributed and Parallel Databases, 16(1), S. 91-116, 2004.
- [RRD04a] Rinderle, S.; Reichert, M.; Dadam, P.: Correctness Criteria for Dynamic Changes in Workflow Systems – A Survey. Data and Knowledge Engineering, 50(1), S. 9-34, 2004.
- [Stro09] Strohbücker, B.: Prozess-Stabilität durch Neuordnung von Tätigkeiten. 12. QM-Forum des Verbandes der Universitätsklinika Deutschlands e.V. (VUD), 27.06.2009.
- [WaPa07] Wagner, K.W.; Patzak, G.: Performance Excellence – Der Praxisleitfaden zum effektiven Prozessmanagement. Carl Hanser Verlag, München, 2007.

- [WiWa11] Wieland, A.; Wallenburg, C.M.: Supply-Chain-Management in stürmischen Zeiten. Universitätsverlag der TU Berlin, 2011.
- [Zhou06] Zhou, H.: Effiziente Überprüfung semantischer Korrektheit in adaptiven Prozess-Management-Systemen. Universität Ulm, Diplomarbeit, 2006.

Eidesstattliche Erklärung

Name des Studierenden: Theis, Michael

Matrikelnummer: 7651724

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt habe. Andere als die angegebenen Quellen und Hilfsmittel habe ich nicht benutzt, wörtliche oder sinngemäße Zitate wurden als solche gekennzeichnet.

Schiffweiler, den 13.08.2012