

Instanz-Migration bei der Evolution von Prozess-Choreographien

Diplomarbeit an der Universität Ulm
Fakultät für Informatik
Abteilung Datenbanken und Informationssysteme



vorgelegt von:

Carolin Hitzer

März 2007

Gutachter: *Prof. Dr. Peter Dadam*
Dr. Stefanie Rinderle

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
1 Einleitung	1
1.1 Problemstellung	2
1.2 Aufbau der Arbeit	4
2 Verwandte Arbeiten	6
2.1 Definitionen einer Choreographie	6
2.2 Top-Down vs. Bottom-Up Entstehung	8
2.3 Workflow-Evolution	10
2.4 Evolution von Choreographien	12
3 Grundlagen	15
3.1 Privater und öffentlicher Prozess	16
3.2 Orchestrierung und Choreographie	18
3.3 Choreographie mit Schleifen	20
3.4 Instanzen in einer Prozess-Choreographie	21
4 Prozess-Choreographie Ansätze	22
4.1 Ansatz 1 - Verteilter Workflow	22
4.2 Ansatz 2 - Inhouse Choreographie	23
4.3 Ansatz 3 - Choreographie mit globalem öffentlichen Prozess	24
4.3.1 Subtraktive Änderungsoperation	25
4.3.2 Additive Änderungsoperation	26
4.4 Ansatz 4 - Choreographie mit verteilten öffentlichen Prozessen	28
4.4.1 Kaskadierende subtraktive Änderungsoperation	29
4.4.2 Ablaufänderung	30
5 Korrektheit bei dynamischen Änderungen	32
5.1 Korrektheitskriterium für den zentralen Fall	33
5.2 Korrektheitskriterium für den verteilten Fall in einer Prozess-Choreographie	33
5.3 Vorgehen bei der Durchführung von Änderungen	36

6 Lokale Entscheidungsverfahren	39
6.1 Markierungen	41
6.2 Mittiger Entscheider	43
6.3 Verkettete Entscheidung	45
7 Globale Entscheidungsverfahren	48
7.1 Vermeidung von Inkonsistenzen und Deadlocks	48
7.2 Zentraler Änderungsmanager	50
7.2.1 Ablauf bei Ansatz 2	51
7.2.2 Ablauf bei Ansatz 3	52
7.2.3 Ablauf bei Ansatz 4	52
7.3 Zentraler Koordinator	53
7.4 Verkettete Abstimmung	55
8 Migrationsverfahren	58
8.1 Anpassung der Zustände	59
8.2 Zentraler Änderungsmanager	60
8.3 Zentraler Koordinator	63
8.4 Verkettete Durchführung	64
8.5 Optimistisches Verfahren	65
8.6 Versionierung	67
9 Weiterführende Aspekte	69
9.1 Verteilte Schnappschussgewinnung	69
9.2 Verteilter wechselseitiger Ausschluss	71
9.3 Verteilte Transaktionen	73
9.3.1 Nebenläufigkeitskontrolle	74
9.3.2 Eigenschaften einer Transaktion	76
10 Zusammenfassung und Ausblick	78
Literaturverzeichnis	80

Abbildungsverzeichnis

1.1	Beispiel Szenario	3
2.1	Top-Down-Entstehung	8
2.2	Bottom-Up-Entstehung	9
2.3	Workflow-Schema (mit Änderungen entnommen aus [RRD02])	11
2.4	Workflow-Instanzen (mit Änderungen entnommen aus [RRD02])	12
2.5	Schemaevolution bei DYCHOR	13
3.1	Kooperationsaufbau	15
3.2	Privater Prozess	16
3.3	Privater Prozess mit den öffentlichen Aktivitäten	16
3.4	Öffentlicher Prozess	17
3.5	Choreographie-Beispiel mit Interaktionen	19
3.6	Schleife ohne kooperative Aktivitäten	20
3.7	Schleife mit kooperativen Aktivitäten	21
4.1	Verteilter Workflow	23
4.2	Inhouse Choreographie	24
4.3	Choreographie mit globalem öffentlichen Prozess	25
4.4	Löschen der Aktivität h	26
4.5	Einfügen der Aktivität x	27
4.6	Choreographie mit verteilten öffentlichen Prozessen	28
4.7	Löschen der Aktivität e	29
4.8	Verschieben der Aktivität j	31
5.1	Private Instanz erkennt Konflikt	35
5.2	Überblick Entscheidungs- und Migrationsverfahren	38
6.1	Zustandsbild aus Sicht des roten Partners	40
6.2	Zustandsbild mit Markierungen aus Sicht des roten Partners (Ansatz 3)	42
6.3	Zustandsbild mit Markierungen aus Sicht des roten Partners (Ansatz 4)	44
6.4	Zustandsbild mit Markierungen aus Sicht des blauen Partners (Ansatz 3)	45
7.1	Entscheidungsfindung mit zentralem Koordinator (Ansatz 3)	54

7.2	Verkettete Abstimmung (Ansatz 4)	57
8.1	Mögliche Kombinationen der Verfahren	58
8.2	Migration mit Zustandsanpassung	60
8.3	Verschiedene Versionen nach der Migration	62
8.4	Verzögerte Migration	65
8.5	Konflikt - ein Partner migriert nicht	66
9.1	Verteilter Schnappschuss bei einer Choreographie-Instanz	70
9.2	Verteilter wechselseitiger Ausschluss bei einer Änderung	71
9.3	Verteilter wechselseitiger Ausschluss bei zwei Änderungen	72
9.4	Verteilte Transaktion mit Sperren	75

1 Einleitung

In vielen Unternehmen sind bereits Workflow Management Systeme (WfMS) im Einsatz, um Geschäftsprozesse zu automatisieren und zu kontrollieren. Diese helfen bei der Durchführung unternehmensweiter Abläufe, indem sie den Benutzer ablaufbezogen unterstützen, die laufenden Prozesse koordinieren und den realen Verlauf dokumentieren [RRD02]. Viele Unternehmen konnten dadurch ihre internen Arbeitsabläufe optimieren und Kosten senken. Dennoch sehen sich viele Unternehmen im Kontext der Globalisierung weiter einem großen Wettbewerbsdruck ausgesetzt. Dieser Wettbewerb führt dazu, dass eventuell zusätzliche intensive Umstrukturierungen der Prozesse notwendig sind, um Produkte und Dienstleistungen noch effizienter und kostengünstiger zu machen [CTD04].

Eine Möglichkeit besteht darin, die enge Zusammenarbeit mit Geschäftspartnern (Business-to-Business, B2B) besser elektronisch zu unterstützen und zu automatisieren. In einem so entstehenden Organisations-Netzwerk können verschiedene Unternehmen ihre Dienstleistungen zusammenlegen und ein aufgewertetes komplexeres Produkt anbieten [LASS00]. Eine große Herausforderung besteht bei der Zusammenarbeit verschiedener Organisationen vor allem darin, dass die teilnehmenden Unternehmen weiter als autonome Einheiten gesehen werden müssen. Zusätzlich müssen die in den Unternehmen bereits bestehenden heterogenen Systeme in die neue Kooperation prozessorientiert integriert werden. Es findet also eine Verbindung der einzelnen Geschäftsprozesse und ein Informationsfluss zwischen verschiedenen Organisationen statt und dieses Szenario wird in einer so genannten Prozess-Choreographie beschrieben.

Ein Beispiel ist das Outsourcing von Teilen eines Prozesses, die ein Unternehmen nicht selber durchführen kann oder will. So kann man sich einen Versandhandel vorstellen, der die Bestellungen durch einen externen Paketdienst ausliefern lässt. Die Modellierung solcher kooperativen Prozesse findet meist unter Benutzung eines elektronischen Geschäftsvertrages statt, in dem die Partner festlegen, was die Zusammenarbeit genau beinhalten soll. Die Zusammenarbeit verschiedener Organisationen kann zeitlich begrenzt sein, z.B. bis ein Projekt abgeschlossen ist. In diesem Zusammenhang wird auch von einer virtuellen Organisation gesprochen, die aus zeitlich und räumlich getrennten realen Organisationen besteht [CTD04]. Ein weiteres Beispiel und eine spezielle Form von Choreographien stellt ein E-Marketplace dar. Hier treffen sich Benutzer und softwaregesteuerte Agenten um zu verhandeln und Geschäfte zu tätigen, wobei die Verhandlungsfunktion mit elektronischen Mitteln ausgeführt wird (E-Commerce), z.B. wie bei Auktionen im Internet [BR05]. Web Services werden dann auch häufig im Kontext der

Choreographien aufgeführt, um die Kooperation zwischen Organisationen zu implementieren. Das Web entwickelt sich zur Hauptplattform um Daten und Services sowohl für Benutzer als auch für Anwendungen zugänglich zu machen und die verfügbaren Standards bieten sich an für den Aufbau einer serviceorientierten Architektur mit dem Ziel der automatischen Auswahl und der prozessorientierten Komposition von Diensten [RS04].

1.1 Problemstellung

Im Bereich zentraler Workflows ist seit längerem bekannt, dass Adaptivität eine wichtige Eigenschaft moderner WfMS darstellt, da es sich herausgestellt hat, dass prozessorientierte Anwendungen sehr viel häufiger angepasst werden müssen als funktionsorientierte Informationssysteme [RRD02]. Entscheidend für die Adaptivität ist die konsistente und effektive Handhabung der Evolution von Workflows, d.h. der Änderungen bei existierender Workflows während der Laufzeit [CCPP98].

Die Forschung im Bereich des Workflow Managements beschäftigt sich seit ein paar Jahren mit dem Thema der Choreographien. Es wurden Modelle für verschiedene Formen der Kooperation zwischen Organisationen entwickelt, welche unterschiedliche Eigenschaften haben und je nach Gegebenheiten und Zielsetzung variieren. Das Thema Adaptivität bei Choreographien, d.h. Änderungen die mehrere Kooperationspartner betreffen können, wurde aber bisher vernachlässigt. Wenn überhaupt werden nur lokale Änderungen erlaubt, die den gegebenen Kooperationsvertrag nicht beeinflussen dürfen. Eine Prozess-Choreographie basiert aber oft auf lang laufenden Unternehmensprozessen, welche häufige Interaktionen zwischen den Kooperationspartnern erfordern. Es ist schwer von Anfang an auszumachen, ob die entsprechende Realisierung der angestrebten Kooperation voll entspricht, oder ob nachträglich noch Anpassungen vorgenommen werden müssen. Dynamische Anpassungen sollten also auch bei einer Prozess-Choreographie möglich sein.

Bei der Evolution von Workflows sind zwei Aspekte zu berücksichtigen. Als erstes die statische Evolution, bei der das Schema, also die Beschreibung des Workflows, verändert wird. Dabei muss die neue Version des Schemas allerdings auch wieder syntaktisch korrekt sein. Als zweites die dynamische Evolution, wobei laufende Instanzen an ein neues Schema angepasst werden, oder Änderungen, welche nur diese eine Instanz betreffen (so genannte Ad-Hoc-Änderungen), vorgenommen werden. Man spricht hierbei auch von der Migration der Workflow-Instanz. Bei der Überprüfung, ob eine Instanz migrieren kann, müssen deren aktuellen Zustände berücksichtigt werden und unerwünschte Effekte müssen ausgeschlossen werden können.

Bei der Evolution von Prozess-Choreographien vergrößert sich die Problematik dadurch, dass die Änderungen mehrere Partner betreffen können, die an der Zusammenarbeit beteiligt sind. Aufgrund dessen, dass die Partner weiter autonome Einheiten sind, kann man sie nicht zwingen zu migrieren. Außerdem kommt erschwerend hinzu, dass die Autonomie den Partnern ermöglicht, nicht alle internen Abläufe öffentlich zu machen, wodurch die Auswirkungen einer Änderung nicht global absehbar sind.

Im Projekt DYCHOR [RWR06] wurde ein erster Ansatz zur Evolution von Choreographien auf Schemaebene vorgestellt und damit ein Grundstein zur statischen Korrektheit gelegt. Daneben ist allerdings auch die dynamische Korrektheit der Prozess-Choreographie bei Änderungen unabdingbar, d.h. auch die Zustände laufender Choreographie-Instanzen müssen bei Modifikationen der Choreographie jederzeit korrekt sein. Ansonsten kann es wie im zentralen Fall zu Verklemmungen oder unerwünschten Systemzuständen kommen.

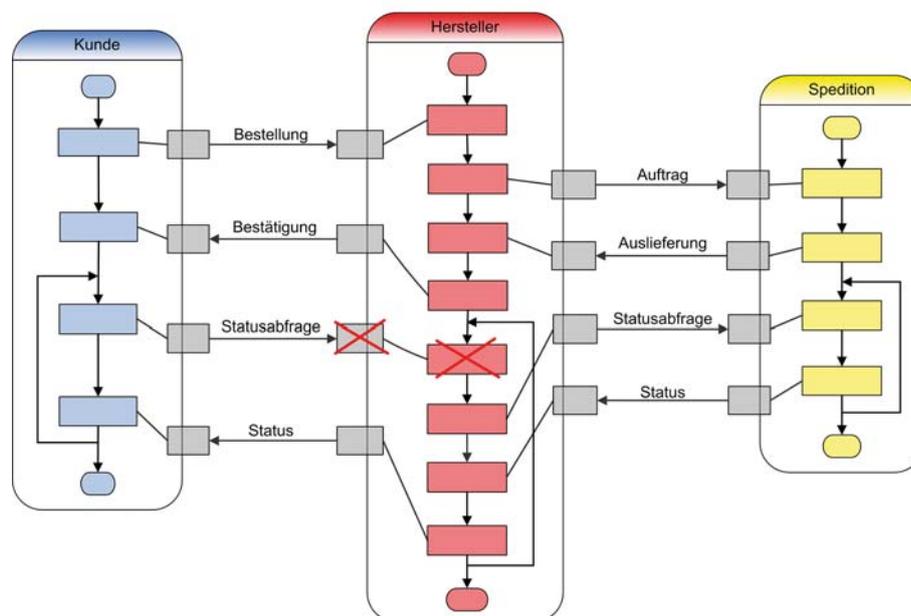


Abbildung 1.1: Beispiel Szenario

In Abbildung 1.1 ist ein Beispiel Szenario für eine Prozess-Choreographie dargestellt. In dieses Szenario sind drei Kooperationspartner involviert: ein Kunde, ein Hersteller und eine Spedition. Jeder Partner hat einen privaten Workflow, dessen Struktur den anderen nicht bekannt ist. Die Partner tauschen Nachrichten über ihre öffentlichen Aktivitäten aus, welche eine Art Schnittstelle zwischen deren privatem Workflow und dem Kooperationspartner darstellen. Der Kunde in Abbildung 1.1 kann z.B. eine Bestellung an den Hersteller senden und sich dann nach deren Status erkundigen. Der Hersteller möchte nun aber keine Statusabfragen mehr entgegennehmen

und den Kunden in Zukunft ohne Aufforderung über den Status informieren. Diese Änderung an seinem privaten Workflow hat auch Auswirkungen auf die öffentlichen Aktivitäten. Da die Änderungsanfrage des Kunden nun nicht mehr vom Hersteller entgegengenommen wird, müssen auch der private Workflow und die öffentlichen Aktivitäten des Kunden angepasst werden. Wie sich strukturelle Änderungen auf die Partner auswirken und welche Modifikationen dadurch bei den Partnern nötig werden, um die Korrektheit der Choreographie sicherzustellen, wurde von Rinderle, Wombacher, Reichert [RWR06] untersucht. Diese Korrektheit auf Schemaebene reicht aber nicht aus, wenn die Änderungen auch auf laufende Instanzen übertragen werden sollen. Eine Instanz kann nur dann auf ein geändertes Schema migrieren, falls ihr bisheriger Ablauf die Änderung zulässt, d.h. der aktuelle Zustand der Instanz muss berücksichtigt werden. Der Hersteller in Abbildung 1.1 kennt weder die internen Abläufe des Kunden, noch dessen aktuellen Zustand. Da die gewünschte Änderung aber auch den Kunden betrifft, müssen sowohl der aktuelle Zustand des Herstellers, als auch der des Kunden für die Überprüfung berücksichtigt werden.

Der aktuelle globale Zustand einer Choreographie-Instanz liegt bei einer Prozess-Choreographie über alle Partner hinweg nicht vor. Aufbauend auf dieser Situation, besteht die Problemstellung dieser Arbeit darin, Konzepte zur Prüfung der Instanzzustände in einem verteilten Choreographie-Szenario zu entwickeln, damit darüber entschieden werden kann, ob eine Migration möglich ist oder nicht. Für diese Entscheidung wird, wie im zentralen Fall, ein Korrektheitskriterium benötigt, welches für den verteilten Fall in einer Choreographie erweitert werden muss. Dafür soll eine geeignete Darstellung für die Abläufe und Abhängigkeiten in einer Prozess-Choreographie gefunden werden. Auf dieser Grundlage kann dann aufgezeigt werden, welche Informationen für eine Entscheidung zur Verfügung stehen. Außerdem sollen Strategien diskutiert werden, wie das Korrektheitskriterium bei der Änderungsdurchführung zugesichert werden kann. Da die Migration mehrere Partner betrifft, müssen Verfahren entwickelt werden, um die Migration über die verteilten Partner koordinieren zu können.

1.2 Aufbau der Arbeit

In der aktuellen Literatur zur Thematik der Prozess-Choreographien wird deutlich, dass von unterschiedlichen Voraussetzungen ausgegangen wird, wie eng und dynamisch die Zusammenarbeit und die Verbindung der Prozesse verschiedener Organisationen gestaltet werden sollte. Des Weiteren werden bei den bestehenden Choreographie-Modellen unterschiedliche Begriffe für die gleichen Sachverhalte verwendet oder die Definitionen eines Begriffs stimmen nicht überein. Um dies zu verdeutlichen, werden in Kapitel 2 verwandte Arbeiten gegenüber gestellt und die unterschiedlichen Auffassungen aufgezeigt. Es wird in diesem Kapitel auch auf die

Entstehung einer Choreographie eingegangen, da die Struktur einer gegebenen Choreographie für die Thematik der Arbeit von Bedeutung ist und diese häufig von ihrer Entstehung abhängt.

In Kapitel 3 werden anhand von Beispielen Begriffe und Sachverhalte eingeführt, welche die Grundlage für die folgenden Kapitel darstellen. Um die Problematik auf der Ebene der Prozesslogik aufarbeiten zu können, wird eine Darstellung vorgestellt, anhand derer die Struktur einer Prozess-Choreographie abgebildet werden kann.

Um eine Aussage treffen zu können, wann und wie man laufende Instanzen bei der Evolution von Prozess-Choreographien migrieren kann, ist es wichtig den Aufbau eines bestehenden Choreographie-Szenarios zu analysieren. Da es aber nicht nur ein typisches Szenario gibt, werden in Kapitel 4 verschiedene mögliche Ansätze diskutiert und die Problematik bei Änderungsoperationen aufgezeigt.

Auf der Grundlage dieser Ansätze kann dann untersucht werden, welche Informationen über die Partner vorhanden sind oder benötigt werden. In Kapitel 5 wird eine Möglichkeit für ein Korrektheitskriterium für Choreographie-Instanzen auf Basis des lokalen Korrektheitskriteriums hergeleitet.

In Kapitel 6 wird ein lokales Entscheidungsverfahren entwickelt, durch welches nur aufgrund der Zustände eines Partners eventuell bereits eine Vorentscheidung über die Migration der anderen involvierten Partner getroffen werden kann.

Für die erfolgreiche Evolution der gesamten Choreographie muss sichergestellt werden, dass alle Kooperationspartner korrekt migrieren können. Deshalb werden in Kapitel 7 globale Entscheidungsverfahren diskutiert, bei denen die Partner in die Entscheidungsfindung miteinbezogen werden.

Wie die Durchführung der Änderungen für alle Partner koordiniert werden kann, wird in Kapitel 8 anhand von verschiedenen Migrationsverfahren in Zusammenhang mit den Ansätzen aus Kapitel 4 analysiert.

In Kapitel 9 findet eine Überprüfung statt, wie man die Strategien verwandter Themengebiete auf die in den vorherigen Kapiteln erörterten Verfahren anwenden könnte.

2 Verwandte Arbeiten

Modelle und Ansätze für Choreographien oder für einer Choreographie ähnliche Szenarien sind in aktuellen Arbeiten mehrere zu finden (z.B. [GALH01, LASS00, OY05a, CTD04, ZLY05, DBD06, Wom05]). Alle Modelle haben gemeinsam, dass sie versuchen eine Kooperation zwischen autonomen Partnern zu modellieren. Wobei nur soviel an Information von den internen Abläufen, dem so genannten privaten Workflow, preisgegeben wird, wie für die Kommunikation mit den Partnern benötigt wird. Es gibt aber entscheidende Unterschiede in Bezug auf Aufbau und Komplexität der Modelle, die vor allem durch deren unterschiedlichen Background, wie z.B. die klassischen WfMS oder den Web Service Technologien, geprägt werden [MBB⁺03]. Bei den verwendeten Begriffen und deren Definitionen gibt es ebenso keinen allgemeingültigen Standard, wodurch es zusätzlich erschwert wird die Modelle einzuordnen. Die Verbindung der Geschäftsprozesse einzelner Organisationen wird in der Literatur bezeichnet als Inter-Organizational Workflow [Aal99, CTD04, MH05, BKKR03], Cross-Organizational Workflow [RPG05, VG03], Collaborative Business Processes [ZLY05], Service Oriented Business Collaboration [OY05b] oder Business-to-Business Interactions [MBB⁺03].

Der Begriff der Choreographie wird außerdem von verschiedenen Autoren unterschiedlich verwendet, worauf unter 2.1 genauer eingegangen wird. Im Abschnitt 2.2 werden die zwei typischen Entstehungen einer Choreographie vorgestellt. Die Frage nach der Adaptivität dieser Modelle wurde noch überhaupt nicht oder nur mit sehr eingeschränkten Möglichkeiten diskutiert. Deshalb wird unter 2.3 zuerst die Evolution bei zentralen Workflows vorgestellt, da die Erkenntnisse und Lösungen im zentralen Fall als Grundlage für die Evolution von Choreographien dienen sollen.

2.1 Definitionen einer Choreographie

In diesem Abschnitt wird anhand von drei unterschiedlichen Definitionen einer Choreographie gezeigt, dass sich die Ansichten, was unter einer Choreographie zu verstehen ist, deutlich unterscheiden.

Nach van der Aalst u.a. [ADO⁺05]: Der Begriff Choreographie wird im Kontext einer serviceorientierten Architektur (SOA) vorgestellt. Bei einer SOA liegt der Schwerpunkt auf der Beschreibung von Diensten. Dienste bieten eine bestimmte Funktionalität über eine plattform-

unabhängige Schnittstelle an. Alle Partner stellen Dienste bereit und nehmen welche in Anspruch und diese Dienste kommunizieren miteinander, was koordiniert werden muss. Für diese Koordination gibt es drei sich überlappende Konzepte: die Choreographie, die Orchestrierung und die Komposition von Diensten. Unter der Choreographie versteht man, dass die Dienste autonom sind und sich die Koordination des übergreifenden Prozesses teilen. Wohingegen die Orchestrierung die Existenz eines einzelnen Koordinators impliziert. Bei der Komposition von Diensten wird betont, dass größere Dienste aus kleinen zusammengesetzt sind. Um den Unterschied zwischen Choreographie und Orchestrierung zu verdeutlichen, verwendet van der Aalst einen Vergleich mit der Regelung des Straßenverkehrs [ADO⁺05]. Bei der Choreographie handelt es sich um eine Regelung des "Verkehrs" wie bei einem Kreisverkehr und bei der Orchestrierung wird zur Regelung eine Ampel eingesetzt. Nach dieser Definition handelt es sich also um eine Choreographie, wenn alle Partner an der Koordination der Interaktionen beteiligt sind und es keinen Koordinator gibt der diese Aufgabe übernimmt.

Nach Dumas u.a. [DBD06, DZD06]: Der Begriff Choreographie tritt hier auch in Verbindung mit der Interaktion von Diensten in einem serviceorientierten System auf. Jedoch gibt die Choreographie nun eine globale Perspektive auf die Interaktion im Sinne eines ideellen Beobachters, welche alle Interaktionen der involvierten Partner umfasst. Im Gegensatz dazu, stellt die Orchestrierung die Perspektive eines einzelnen Teilnehmers dar und beinhaltet somit nur die Interaktionen in die dieser direkt involviert ist. Nach dieser Definition handelt es sich nur um eine Choreographie, wenn eine globale Sicht auf das Szenario mit allen Partnern und deren Interaktionen gegeben ist.

Nach Reichert/Stoll [RS04]: Die Komposition, d.h. das Zusammenfügen von Diensten zu einem neuen Geschäftsprozess, wird als Überbegriff für Konversation, Choreographie und Orchestrierung aufgeführt. Eine Konversation ist die konkrete Nachrichtenabfolge zwischen Partnern. Wobei sich die Choreographie mit der Beschreibung der zulässigen Nachrichtenabfolge zwischen einem oder mehreren Partnern beschäftigt. Sie beschreibt die Kommunikation aus einer öffentlichen Sicht ohne Angaben über die privaten Arbeitsschritte der Partner. Die Orchestrierung beschreibt die Geschäftsprozesslogik eines Partners, also dessen privaten Workflow und die Kommunikation mit dessen Partnern. Bei dieser Definition wird also bereits durch die unterschiedlichen Beschreibungen von Choreographie und Orchestrierung die Autonomie der Partner ermöglicht.

Der Begriff der Choreographie wird, je nachdem auf welche Definitionsvariante sich der Autor stützt, in aktuellen Arbeiten vor allem dann vermieden, wenn betont werden soll, dass es sich nur um eine temporäre und dynamische Zusammenarbeit handelt [CTD04] oder es keine globale Sicht gibt [ZLY05]. Wenn eine Choreographie auf der Ebene der Prozesslogik betrachtet wird, werden die Dienste in Form von Workflow Aktivitäten, das sind Teilschritte eines Workflows, dargestellt und es wird ein Workflow benützt um die Interaktionen zwischen Workflows

zu modellieren. Dieser Aspekt wird speziell in Kapitel 3 für die Darstellung einer Prozess-Choreographie aufgegriffen.

2.2 Top-Down vs. Bottom-Up Entstehung

Es gibt zwei verschiedene Möglichkeiten der Entstehung eines Choreographie-Szenarios. Da die Struktur einer Choreographie auch von deren Entstehung abhängt, werden die zwei verschiedenen Vorgehensweisen kurz vorgestellt.

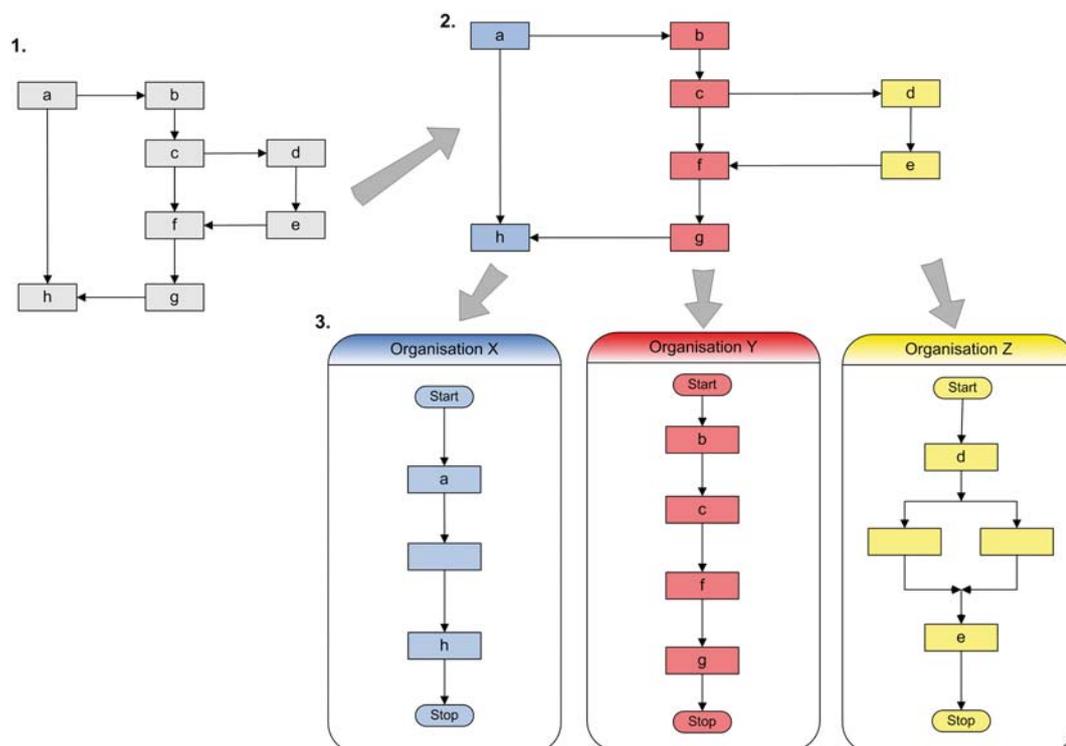


Abbildung 2.1: Top-Down-Entstehung

Die Top-Down Entstehung, welche auch als Public-to-Private Ansatz bezeichnet wird [AW01], basiert auf dem Vererbungsprinzip. Die Entstehung vollzieht sich in drei Schritten. Zuerst einigen sich alle involvierten Organisationen auf einen gemeinsamen öffentlichen Workflow wie in Abbildung 2.1 unter 1. dargestellt, welcher als Vertrag zwischen den Partnern gesehen werden kann. Als zweites wird jede Aktivität, d.h. die Teilschritte a-h, auf eine Organisation abgebildet (siehe Abbildung 2.1 unter 2.). Somit ist jede Organisation für einen Teil des öffentlichen Workflows verantwortlich. Als dritter Schritt erstellt dann jede Organisation autonom ihren privaten Workflow (siehe Abbildung 2.1 unter 3.), wobei der private Workflow eine Unterklasse

nach definierten Vererbungsprinzipien der auf die Organisation abgebildeten Aktivitäten darstellen muss. Dadurch, dass jede Organisation eine Kopie des öffentlichen Workflows hat, ist die Autonomie eines Partners aber etwas eingeschränkt, da durch die Anwendung der Vererbungsprinzipien der private Workflow der anderen erahnbar ist. Wichtig bei dieser Entstehung ist vor allem, dass alle Partner vorab bekannt sein müssen und es einen globalen öffentlichen Workflow gibt [GALH01, AW01], den alle Partner kennen. Dieser öffentliche Workflow muss jedoch nicht immer von allen Partnern zusammen ausgehandelt werden, beispielsweise könnte es auch eine zentrale Einheit geben, die den öffentlichen Workflow vorgibt [BR05].

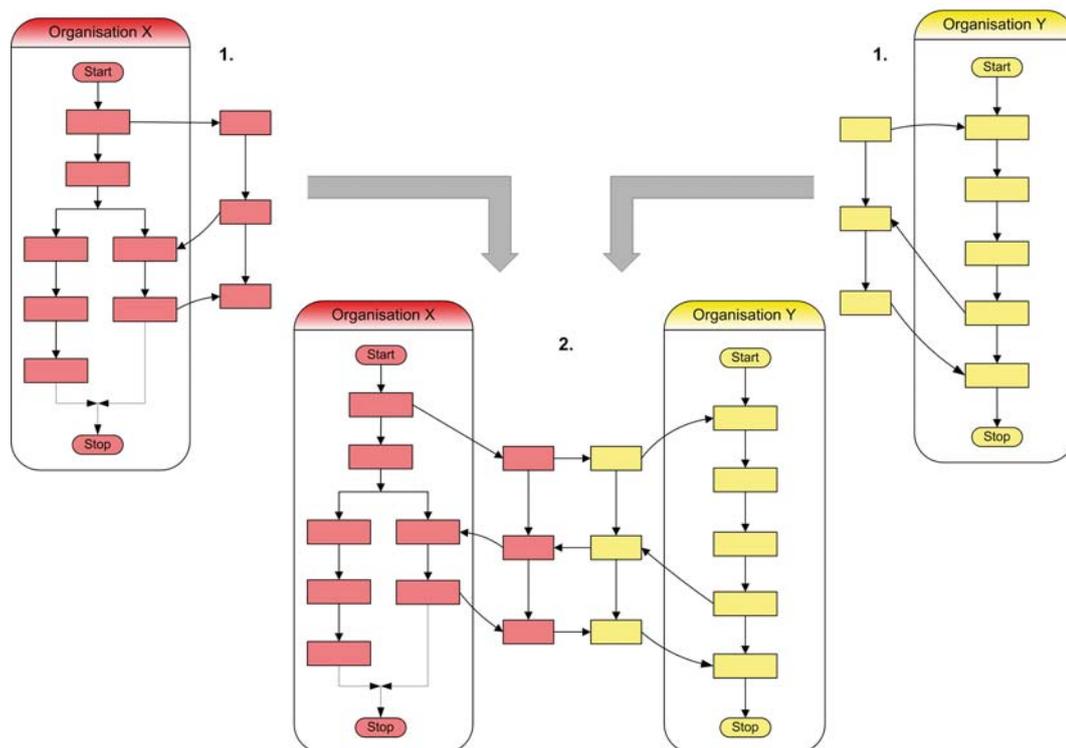


Abbildung 2.2: Bottom-Up-Entstehung

Bei der Bottom-Up Entstehung bilden bereits existierende Workflows verschiedener Organisationen die Ausgangsbasis für die Choreographie. Jeder Partner veröffentlicht die für seinen Workflow benötigten und die von ihm zur Verfügung gestellten Aktivitäten (siehe Abbildung 2.2 unter 1.), z.B. in einer Registry, meistens in Form einer semantischen Beschreibung [CTD04, WFMN04]. Oft werden in der Beschreibung auch Rollen vergeben [ACD⁺05] wie z.B. Verkäufer oder Käufer, die das Auffinden von passenden Partnern erleichtern soll. Eine Rolle ist ein Platzhalter, der dann in einer Konversation durch einen konkreten Dienst ersetzt wird. Wie in Abbildung 2.2 unter 2. dargestellt, müssen die öffentlichen Aktivitäten noch ge-

koppelt werden. Es werden Kooperationsregeln [Tat02] zwischen den öffentlichen Aktivitäten zweier Partner definiert, wodurch auch diese Entstehung vertraglich geregelt ist [GALH01]. Die Verträge sind entweder statisch festgelegt, oder sie werden durch intelligente Agenten bzw. sonstige geeignete Mechanismen ausgehandelt. Im Unterschied zur Top-Down Entstehung besteht dieser Vertrag aber immer nur zwischen zwei Partnern. Eine multilaterale Zusammenarbeit wird also auf Basis bilateraler Kooperation gegründet [Wom05, ZLY05]. Die Vermittlung von passenden Partnern wird auch als Matchmaking bezeichnet und stellt eine komplexe Aufgabe dar. Zur Automatisierung des Matchmaking werden aktuell meist in Verbindung mit (Web) Services einige Vorschläge gemacht [WFMN04, DBRS04, CM05]. Geschäftsbeziehungen sind so konzipiert, dass beide Parteien sowohl Forderungen an den Partner stellen, als auch Zugeständnisse an den jeweils Anderen machen. Wichtig bei dieser Entstehung ist, dass die Basis nicht ein globaler Workflow bildet, sondern die verschiedenen bereits existierenden Workflows der Partner. Dadurch kann es zu einer dynamischen Partnerbildung kommen, d.h. es können bei einer bestehenden Kooperation Partner hinzukommen oder austreten. Auch bestimmt jede Organisation ihr Sichtbarkeitslevel selbst und entscheidet daher nach ihren eigenen Bedürfnissen, wie viele Informationen über interne Abläufe nach außen gegeben wird [ZLY05]. Außerdem ist somit einer Organisation eventuell nicht bekannt, mit welchen anderen Organisationen ihr direkter Verhandlungspartner noch in Verbindung steht.

2.3 Workflow-Evolution

In einem Workflow Management System unterscheidet man bei der Evolution von Workflows zwischen dem statischen und dem dynamischen Fall [CCPP98]. Im statischen Fall werden nur die erforderlichen Strukturtransformationen betrachtet, nach deren Ausführung wieder ein korrektes Schema resultieren sollte. Ein Schema dient zur Modellierung von Workflows und legt unter anderem fest, welche Aktivitäten in welcher Reihenfolge zur Ausführung kommen. Durch das Workflow-Schema sind also bereits zur Modellierungszeit alle zur Laufzeit möglichen Ausführungsvarianten bestimmt. Beim dynamischen Fall werden nun auch bereits laufende Workflow-Instanzen, die basierend auf dem entsprechenden Schema erzeugt wurden, betrachtet. Die auf Schemaebene definierten Änderungen werden dann auf eine laufende Workflow-Instanz propagiert. Man spricht dann auch von der Migration der Workflow-Instanz auf das geänderte Schema. Alternativ gibt es noch die Möglichkeit so genannter Ad-hoc-Modifikationen, d.h. Änderungen bei einer laufenden Instanz direkt vorzunehmen, wobei dann deren Instanzstruktur nicht mehr der entsprechenden Schemavorlage entspricht.

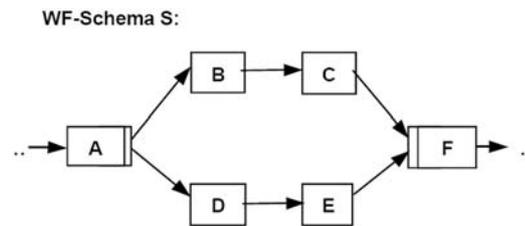


Abbildung 2.3: Workflow-Schema (mit Änderungen entnommen aus [RRD02])

Diese Arbeit stützt sich auf die Ergebnisse, die aus dem ADEPT¹-Projekt hervorgehen, welches Änderungen zur Laufzeit vollständig unterstützt [LRR05]. Abbildung 2.3 zeigt die Spezifikation eines Workflows durch ein WF-Schema S mit seinen Arbeitsschritten, den so genannten Aktivitäten. Die zugehörigen Kontrollkanten verbinden die Aktivitäten und legen den Kontrollfluss fest, welcher die Ausführungsreihenfolge vorgibt. Zusätzlich zu den in der Abbildung verwendeten parallelen Verzweigungen (AND-Split, AND-Joint), können noch alternative Verzweigungen (XOR-Split, XOR-Joint) und Schleifen (STARTLOOP, ENDLOOP) modelliert werden. Ausgehend vom Schema S in Abbildung 2.3 wurden die Instanzen I1 und I2 in Abbildung 2.4 erzeugt und gestartet. Anhand der Zustandsinformationen lässt sich ermitteln, wie weit die Ausführung einer Workflow-Instanz bereits fortgeschritten ist. Der Zustand einer einzelnen Aktivität ist initial NOT_ACTIVATED und geht bei ihrer Aktivierung in ACTIVATED über. Wird eine Aktivität ausgewählt und gestartet, geht sie in den Zustand RUNNING über. Bei Beendigung erhält die Aktivität den Zustand COMPLETED. Steht dagegen fest, dass sie in der Folge nicht mehr ausführbar ist, wird ihr der Zustand SKIPPED zugewiesen. Den Kanten wird initial der Zustand NOT_SIGNALED zugeordnet und werden im Verlauf der Ausführung mit TRUE_SIGNALED oder FALSE_SIGNALED markiert.

Der ADEPT-Ansatz unterstützt sowohl Ad-hoc-Änderungen einer einzelnen Instanz [RD98], als auch die Änderungen auf Schemaebene [RRD02], welche beide für langlaufende Prozesse benötigt werden. Anhand der Struktur und des Zustands der Workflow-Instanz wird überprüft, ob die gewünschte Änderung zulässig ist oder nicht. Eine Änderung bei einer WF-Instanz darf nur dann vorgenommen werden, wenn sie wieder in einem syntaktisch korrekten Workflow, der sich in einem "legalen" Zustand befindet, resultiert. Es darf somit nicht als Folge zu unerwünschten Effekten wie Inkonsistenzen oder Deadlocks kommen. Um dies zu gewährleisten, wurde ein allgemeingültiges Korrektheitskriterium entwickelt, um festzustellen ob eine Workflow-Instanz mit einem geänderten Schema verträglich ist. Angenommen es wird ein korrektes Workflow-Schema S in ein ebenfalls korrektes Schema S' transformiert. Dann dürfen die Änderungen nur dann auf eine Instanz I von S angewendet werden, wenn die WF-Instanz

¹ADEPT steht für Application Development based on Encapsulated Pre-Modeled Process Templates.

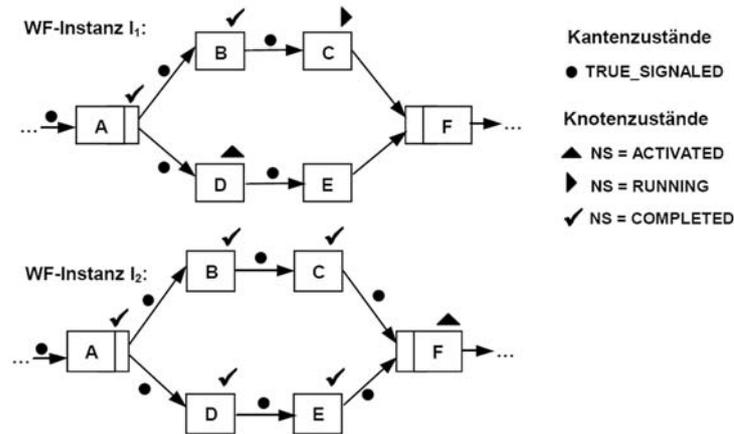


Abbildung 2.4: Workflow-Instanzen (mit Änderungen entnommen aus [RRD02])

I, entsprechend ihrem bisherigen Verlauf, auch auf S' hätte ausgeführt werden können. Es darf also infolge einer Migration der Instanz I auf das geänderte Schema S' nicht zu einer Verfälschung der Vergangenheit kommen [RRD02]. Angenommen beim Schema S in Abbildung 2.3 wird zwischen den Aktivitäten D und E eine zusätzliche Aktivität X eingeführt, und daraus resultiert das Schema S' . Dann würde das für die Instanzen aus Abbildung 2.4 bedeuten, I1 kann auf das geänderte Schema migrieren, aber I2 nicht, da hier bereits E direkt nach D ausgeführt wurde und diese Abfolge unter S' nicht mehr möglich ist. Da für die Verträglichkeitsprüfungen in der Regel nicht die kompletten Ablaufhistorien notwendig sind, wurden Korrektheitskriterien erarbeitet, welche auf den Zustandsinformationen basieren. Diese Kriterien dienen als Grundlage für das Korrektheitskriterium in dieser Arbeit und werden in Kapitel 5.1 vorgestellt.

2.4 Evolution von Choreographien

Die Entwicklung der Organisationen geht in die Richtung zu mehr Flexibilität. Flexible Organisationen sind in der Lage rasch neue Prozesse aufzusetzen oder die existierenden aufgrund entsprechender Änderungen am Markt oder bei der Produktion, schnell anzupassen. Im Bereich der Choreographien wurden Änderungen, die auch die Interaktionen zwischen Partnern betreffen, bisher vernachlässigt. Es wurden zwar Modelle entwickelt, bei denen das dynamische Ein- und Austreten von Partnern in einer bereits laufenden Kooperation möglich ist, aber die Anpassung einer bestehenden Zusammenarbeit ist, wenn überhaupt, nur unter sehr restriktiven Bedingungen möglich. Es sind dann nur Änderungen des privaten Workflows eines Partners möglich, die entweder die Unterklasse-Bedingung zum öffentlichen Workflow aufrechterhalten [AW01] oder den bestehen Kooperationsvertrag nicht verletzen [CTD04]. Demnach sind Änderungen nicht vorgesehen, die die Kooperation, also die öffentlichen Aktivitäten, betref-

fen. Falls solche Änderungen nötig wären, müsste man die bestehende Kooperation beenden und mit einem neuen Vertrag eine neue Zusammenarbeit aufsetzen. Vor allem für langlaufende Prozess-Choreographien stellt diese Einschränkung ein Problem dar, da sie entweder ohne die nötigen Änderungen zu Ende laufen müssen oder sie werden abgebrochen und bereits ausgeführte Arbeiten werden verworfen.

Im Projekt DYCHOR [RWR06] wurde ein erster Ansatz zur Evolution von Prozess-Choreographien auf Schemaebene vorgestellt. Änderungen die den öffentlichen Prozess und somit auch die Partner betreffen können, dürfen nicht unkontrolliert erfolgen, da es sonst zu Inkonsistenzen oder Fehlern bei der Zusammenarbeit kommen kann. Es werden Methoden benötigt, um die Änderungen eines privaten Workflows, falls es nötig ist, auch auf die öffentlichen Aktivitäten und die privaten Workflows der Partner zu propagieren.

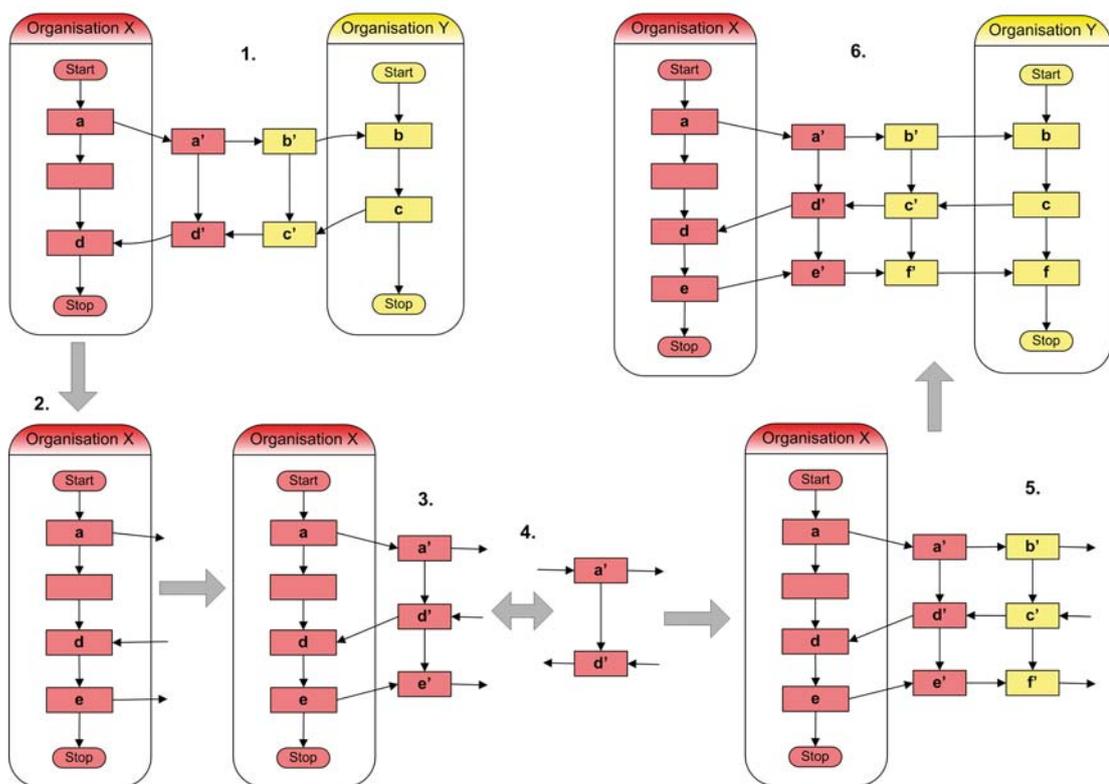


Abbildung 2.5: Schemaevolution bei DYCHOR

Die Vorgehensweise soll vereinfacht an einem Beispiel, welches in Abbildung 2.5 dargestellt ist, kurz vorgestellt werden. Das formale Modell wird in [RWR06] ausführlich beschrieben. Die Ausgangsbasis für das Beispiel ist eine Choreographie mit zwei Partnern X und Y (siehe Abbildung 2.5 unter 1.), sowie deren private Workflows und die zugehörigen öffentlichen

Aktivitäten a'-d'. Organisation X führt eine Änderung an ihrem privaten Workflow durch das Einfügen der Aktivität e aus (siehe Abbildung 2.5 unter 2.). Daraufhin werden auch die öffentlichen Aktivitäten von X erneuert und die nun benötigte Aktivität e' hinzugefügt (siehe Abbildung 2.5 unter 3.). Als 4. Schritt werden die "alten" öffentlichen Aktivitäten mit den neu erstellten verglichen und festgestellt, ob es zu Änderungen bei den öffentlichen Aktivitäten von X gekommen ist. Falls ja, muss geprüft werden, ob auch Anpassungen bei den öffentlichen Aktivitäten von Y nötig sind. Dies geschieht indem die Konsistenz zwischen den öffentlichen Aktivitäten von X und Y berechnet wird und bei Inkonsistenzen die öffentlichen Aktivitäten von Y durch anfügen von f' entsprechend geändert werden (siehe Abbildung 2.5 unter 5.). Als nächstes muss noch der private Workflow von Organisation Y angepasst werden (Abbildung 2.5 unter 6.). Für die notwendigen Änderungen können aber nur Vorschläge gemacht werden, da Anpassungen am privaten Workflow eines Partners aufgrund deren Autonomie nicht automatisch erfolgen können. Im Projekt DYCHOR werden die privaten Prozesse mit BPEL² beschrieben, einer Beschreibungssprache mit der man das Verhalten von Geschäftsprozessen basierend auf Diensten, welche als Web Services zur Verfügung stehen, spezifizieren kann. Der Fokus von BPEL liegt auf der Orchestrierung bestehender Dienste. Für die Beschreibung der öffentlichen Prozesse wird die Darstellung eines annotierten endlichen Zustandsautomaten verwendet. Dieser Formalismus wurde gewählt, damit z.B. die Aussage der Konsistenz durch den Schnitt zweier Automaten einfach getroffen werden kann.

²BPEL steht für Business Process Execution Language

3 Grundlagen

Wie sich in Kapitel 2 herausgestellt hat, gibt es keine allgemeingültige Definition wie eine Choreographie aufgebaut ist, oder was sie genau beinhaltet. Deshalb werden in diesem Kapitel die Grundlagen und Begrifflichkeiten besprochen, welche für das Verständnis der weiteren Arbeit benötigt werden.

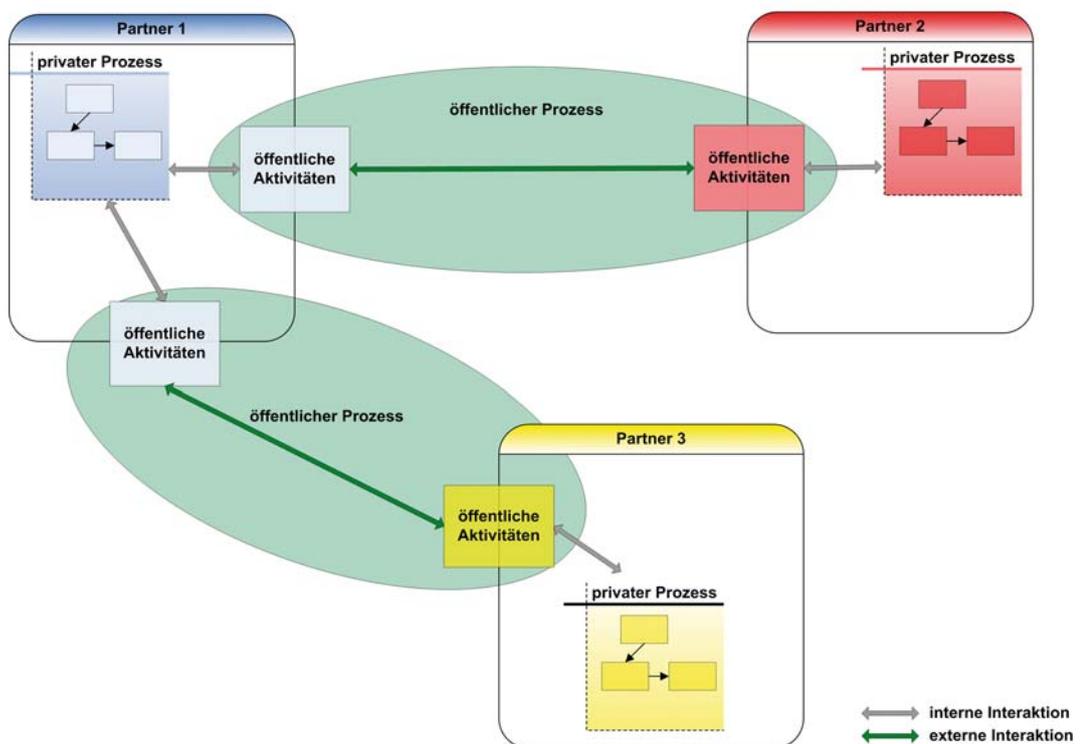


Abbildung 3.1: Kooperationsaufbau

In einer Choreographie kommt es zu Interaktionen zwischen den beteiligten Workflows. Die verschiedenen Partner kommunizieren über ihre öffentlichen Prozesse durch den Austausch von Nachrichten [RWR06] (siehe externe Interaktionen in Abbildung 3.1). Diese Kommunikation kann z.B. über eine Web Service Middleware realisiert werden [MBB⁺03]. Die Web Services sind aber nicht Schwerpunkt dieser Arbeit, da es um die Prozesslogik der Choreographien geht und diese wie bei den WfMS getrennt von der Implementierung der Anwendungsfunktion

betrachtet werden sollte. Eine ausführliche Diskussion der möglichen Technologien findet in [MBB⁺03] statt.

Die öffentlichen Prozesse dienen dazu, den privaten Workflow der Partner zu verbergen, um deren Autonomie zu wahren. Die Trennung zwischen privaten und öffentlichen Prozessen hilft aber auch die Heterogenität zwischen den von den Organisationen verwendeten Technologien zu überbrücken.

3.1 Privater und öffentlicher Prozess

Die Unterschiede zwischen einem privaten und einem öffentlichen Prozess werden anhand eines konkreten Beispiels erläutert.

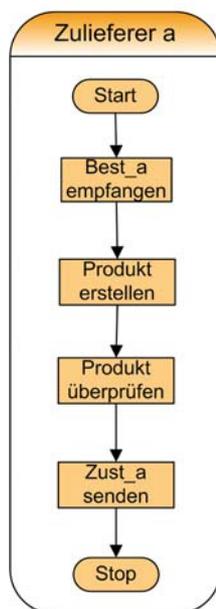


Abbildung 3.2: Privater Prozess

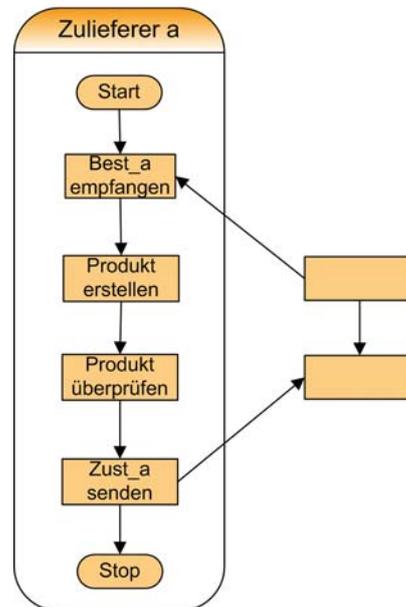


Abbildung 3.3: Privater Prozess mit den öffentlichen Aktivitäten

Die Abbildung 3.2 zeigt den privaten Workflow von Zulieferer a, dessen privater Prozess enthält 4 Aktivitäten. Zwei dieser Aktivitäten Best_a empfangen und Zust_a senden sind so genannte kooperative Aktivitäten, d.h. bei diesen Aktivitäten kommt es zu einer Interaktion mit anderen Partnern. Es gibt zwei verschiedene Typen der kooperativen Aktivitäten und zwar die sendende Aktivität und die empfangende Aktivität, oder bei der Modellierung durch Dienste die anbietende und die konsumierende Aktivität. Um die interne Struktur von Zulieferer a zu verbergen werden von den kooperativen Aktivitäten die öffentlichen Aktivitäten abgeleitet.

Die Abbildung 3.3 zeigt die öffentlichen Aktivitäten von Zulieferer a, welche aus den zwei kooperativen Aktivitäten des privaten Prozesses abgeleitet werden und mit diesen verbunden sind. Es werden für jeden Partner, mit dem direkt kommuniziert wird, die benötigten öffentlichen Aktivitäten abgeleitet. Diese öffentlichen Aktivitäten werden manchmal auch als “public view“ auf den privaten Workflow bezeichnet. Der Begriff “view“ ist aber nicht ganz passend, denn es handelt sich nicht nur um das Ausblenden von Informationen, die nicht gesehen werden sollen, wie z.B. im Bereich der Datenbanken, sondern durch die öffentlichen Aktivitäten wird die Interaktion mit dem Partner ermöglicht. Sie werden, wie in Abbildung 3.4 dargestellt, mit den öffentlichen Aktivitäten des Partners verbunden und repräsentieren den Kooperationsvertrag zwischen den Organisationen Zulieferer a und Hersteller. Diese öffentlichen Aktivitäten mit ihren Kanten werden auch als Workflow einer virtuellen Organisation bezeichnet, die nur besteht, solange die Kooperation aufrechterhalten wird. Zu bevorzugen ist aber die Bezeichnung öffentlicher Prozess (zwischen Zulieferer a und Hersteller), um eventuellen Missverständnissen vorzubeugen, da es sich um eine Kooperation und nicht um eine neue Organisation bzw. Partner handelt.

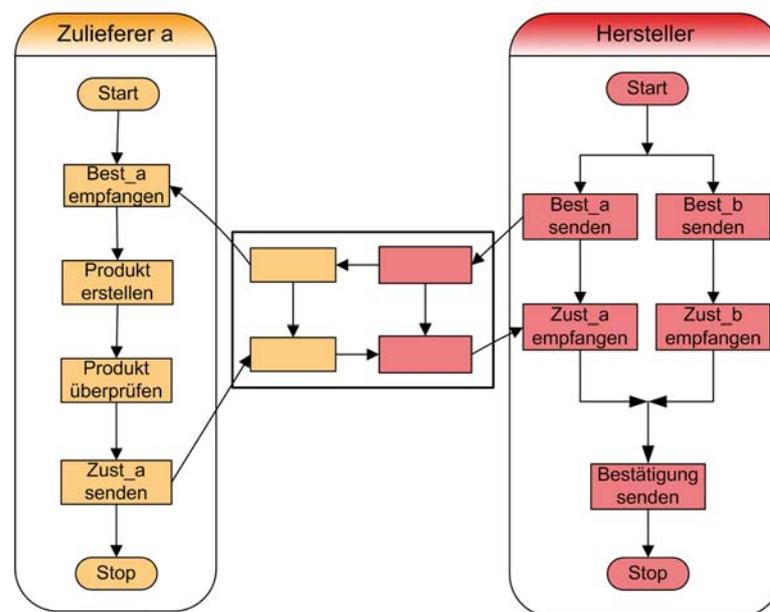


Abbildung 3.4: Öffentlicher Prozess

Von einer sendenden kooperativen Aktivität gehen immer zwei Kanten aus, eine als Verbindung zur entsprechenden öffentlichen Aktivität und die andere zu der als nächstes auszuführende Aktivität des privaten Workflows. Es handelt sich dabei logisch immer um einen AND-Split, also werden immer beide ausgehenden Kanten auf TRUE_SIGNED gesetzt nach dem Beenden

der Aktivität. Nach dem gleichen Prinzip stellt eine empfangende kooperative Aktivität immer einen AND_Joint dar und beide Kanten müssen TRUE_Signaled haben, damit die Aktivität aktiviert werden kann. Ebenso verhält es sich bei den Aktivitäten des öffentlichen Prozesses, von denen zwei Kanten ausgehen bzw. bei denen zwei Kanten enden. Bei dieser Darstellung müssen demnach immer alle kooperativen und öffentlichen Aktivitäten zur Ausführung kommen. Die Abhandlung alternativer Interaktionen zwischen Partnern, d.h. die kooperativen Aktivitäten sind in einem OR-Zweig modelliert und kommen deshalb möglicherweise nicht zur Ausführung, ist in dieser Arbeit nicht vorgesehen, da im Zusammenhang mit Choreographien eine alternative Kooperation in der einschlägigen Literatur bislang nicht diskutiert wurde.

3.2 Orchestrierung und Choreographie

Nachdem, wie in Kapitel 2 bereits diskutiert wurde, nicht klar definiert ist, was mit den Begriffen Choreographie und Orchestrierung genau beschrieben wird, sollen diese anhand des Beispiels in Abbildung 3.5 erläutert werden. Dargestellt sind vier in eine Kooperation involvierte Partner (Kunde, Hersteller, Zulieferer a und Zulieferer b). Kunde, Zulieferer a und Zulieferer b stehen alle in bilateraler Beziehung mit dem Hersteller. Der Hersteller hat demnach für jeden Partner die entsprechenden öffentlichen Aktivitäten bereitgestellt und mit denen seiner Partner verbunden. Dadurch entstehen drei öffentliche Prozesse. Als Orchestrierung bezeichnen wir den privaten Workflow der Partner. Dazu gehören alle privaten Aktivitäten, das sind die Aktivitäten zwischen Start und Stop, und die Kontrollflusskanten zwischen diesen Aktivitäten. Mit der Choreographie ist das gesamte Szenario mit allen Partnern (Kunde, Hersteller, Zulieferer a und Zulieferer b) gemeint und es werden in deren Beschreibung die öffentlichen Prozesse zwischen den Partnern definiert. Der wesentliche Fokus einer Choreographie-Beschreibung liegt demnach auf der Spezifizierung der Berührungspunkte zwischen Kooperationspartnern. Hierbei wird nicht der Anspruch erhoben, eine ausführbare Beschreibung einer solchen Choreographie zu erstellen, sondern es soll diese vielmehr als eine Schablone fungieren.

In [CM05] wird davon ausgegangen, dass jede Choreographie einen öffentlichen Prozess repräsentiert, aber um einen Überblick zu bekommen wer und in welcher Form an einer Kooperation beteiligt ist, erscheint es sinnvoller mit nur einer Choreographie alle involvierten Partner und damit alle öffentlichen Prozesse zu erfassen. An dieser Stelle soll betont werden, dass auch die privaten Prozesse zur Choreographie gehören, nur werden diese durch die Orchestrierung beschrieben.

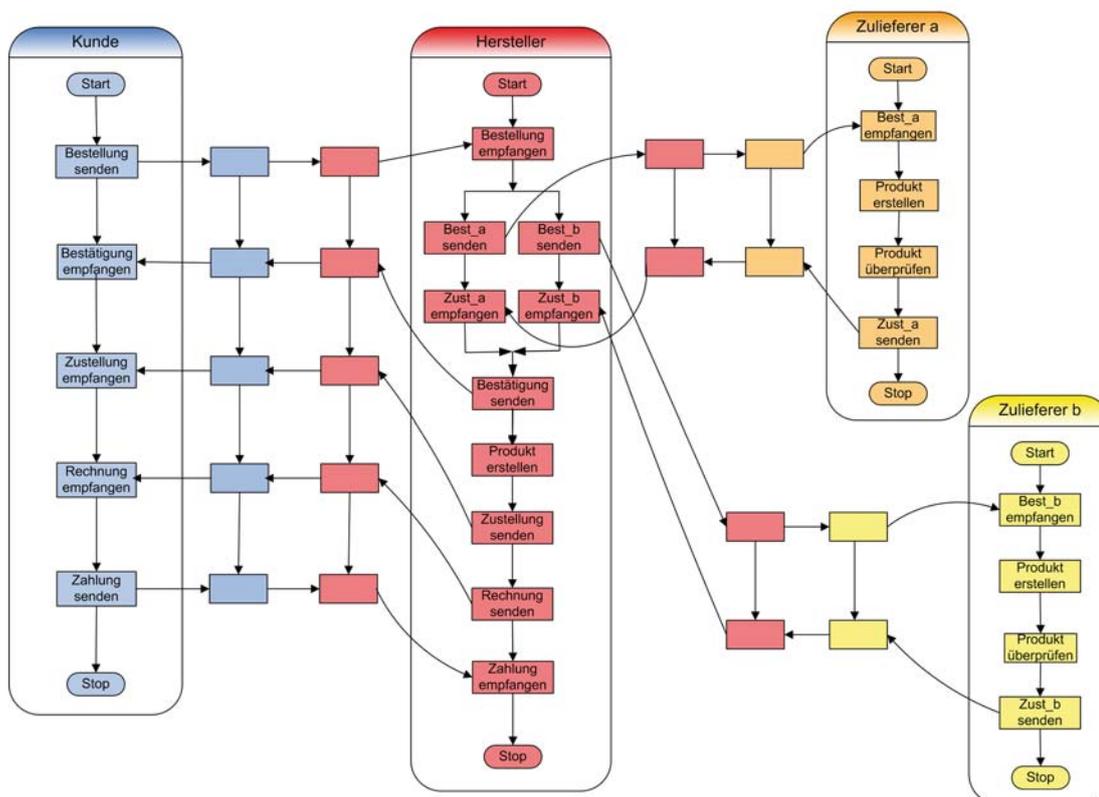


Abbildung 3.5: Choreographie-Beispiel mit Interaktionen

3.3 Choreographie mit Schleifen

Bei Schleifen muss berücksichtigt werden, ob sie auch kooperative Aktivitäten mit einschließen, oder nur innerhalb eines privaten Workflows zu Wiederholungen führen.

In Abbildung 3.6 enthält der private Workflow des gelben Partners eine Schleife. Für den roten Partner ist diese Schleife nicht sichtbar und beeinflusst ihn auch nicht, da keine kooperativen Aktivitäten in die Schleife involviert sind. Eine Schleife wird durch spezielle Aktivitäten und Kanten repräsentiert. Die Aktivität L_S ist die Start-Aktivität und die Aktivität L_E dementsprechend die Ende-Aktivität für die Schleife. In dem Beispiel in Abbildung 3.6 ist die Bedingung für einen erneuten Schleifendurchlauf auf FALSE gesetzt worden, weshalb die Rücksprungkante (Loop) den Zustand FALSE_SIGNED bekommen hat.

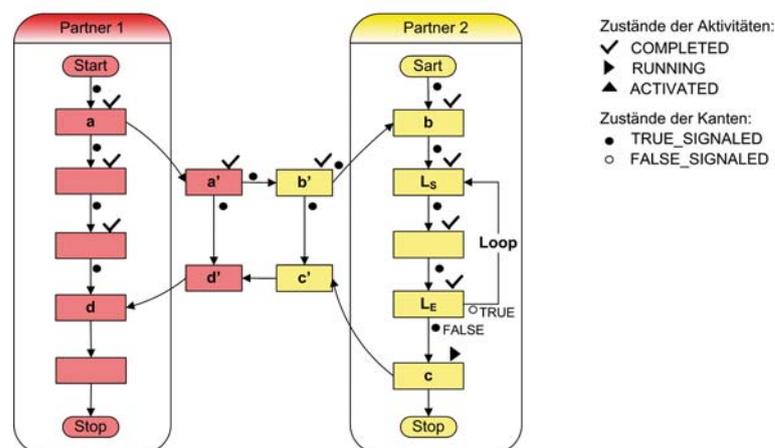


Abbildung 3.6: Schleife ohne kooperative Aktivitäten

Sobald kooperative Aktivitäten in die Schleife involviert sind, kann es zum mehrmaligen ausführen der kooperativen Aktivitäten und dadurch auch zum wiederholten Senden von Nachrichten über die öffentlichen Aktivitäten kommen. Deshalb muss eine entsprechende Schleife auch beim Partner modelliert sein. In Abbildung 3.7 hat Partner 1 eine Schleife, welche die kooperativen Aktivitäten a und d enthält und Partner 2 hat entsprechend eine Schleife mit den kooperativen Aktivitäten b und c. Das Durchlaufen der Schleifen ist prinzipiell gleich wie bei einer Schleife ohne kooperative Aktivitäten, allerdings bestimmt bei zwei kooperativen Schleifen nur eine Aktivität L_E , ob die Rücksprungbedingung auf TRUE oder FALSE gesetzt wird. Im Beispiel in Abbildung 3.7 ist dies Partner 1, der dann Partner 2 mittels den öffentlichen Aktivitäten $L1'$ und $L2'$ über die Entscheidung informiert, damit dieser bei TRUE auch eine weitere Iteration einleiten kann. Diese Synchronisation ist deshalb wichtig, weil beim Rücksprung die

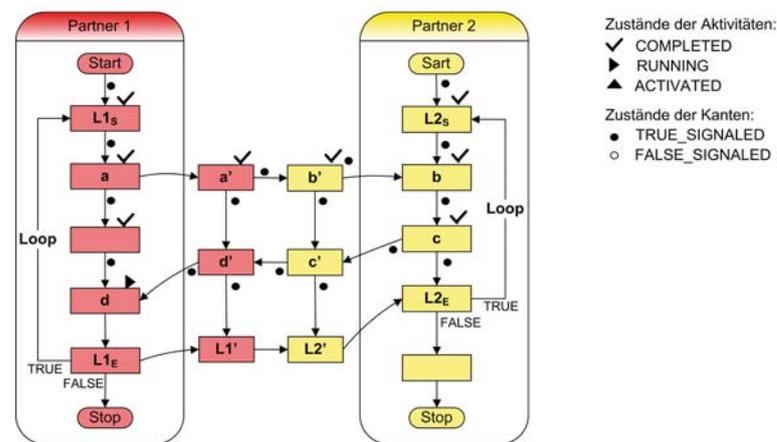


Abbildung 3.7: Schleife mit kooperativen Aktivitäten

Zustände der Aktivitäten und Kanten zurückgesetzt werden, um einen weiteren Durchlauf zu ermöglichen.

3.4 Instanzen in einer Prozess-Choreographie

Die Frage wie viele Instanzen eine bestehende Choreographie - logisch gesehen - hat ist nicht trivial zu beantworten. Erstmal hat jeder private Workflow eine Instanz, es gibt also so viele private Instanzen wie Partner in der Choreographie. Dazu kommt dann noch eine Choreographie-Instanz, welche das gesamte Choreographie-Szenario wie in Abbildung 3.5 auf Seite 19 gezeigt umfasst. Nun könnte man noch vermuten, dass es für jeden öffentlichen Prozess auch noch eine Instanz gibt. Bei der Choreographie in Abbildung 3.5 auf Seite 19 wären das dann nochmals drei Instanzen. Dies erscheint aber nicht sinnvoll, wenn davon ausgegangen wird, dass es sich bei einem öffentlichen Prozess nicht wirklich um einen ausführbaren Prozess handelt. Es handelt sich dabei mehr um eine Art öffentliche Schnittstelle und Beschreibung des Nachrichtenaustausches. Die öffentlichen Aktivitäten transferieren also Daten zu/von anderen Workflows, z.B. kann die Nachricht hier entsprechend dem benötigten Format ver-/entpackt werden.

4 Prozess-Choreographie Ansätze

In diesem Kapitel werden vier verschiedene Ansätze für ein Choreographie-Szenario vorgestellt, welche von unterschiedlichen Voraussetzungen ausgehen. Die Vorgehensweise bei einer Änderungsdurchführung hängt von den für einen Teilnehmer zur Verfügung stehenden Informationen über die anderen Partner ab und baut demzufolge auf den erarbeiteten Ansätzen auf.

Ansatz 1 orientiert sich an einem verteilten Workflow und soll dessen Unterschied zu einer Prozess-Choreographie verdeutlichen. Ansatz 2 beinhaltet die Kooperation verschiedener WfMS, stellt aber einen Sonderfall dar, weil in einer Inhouse-Choreographie die Partner nicht autonom auftreten müssen. Bei Ansatz 3 gibt es einen globalen öffentlichen Prozess, wodurch die privaten Workflows geheim gehalten werden. Während bei Ansatz 4 der öffentliche Prozess auch noch verteilt ist. Anhand der Ansätze 3 und 4 werden dann noch je zwei typische Änderungen in einer Prozess-Choreographie anhand von Beispielen vorgestellt.

4.1 Ansatz 1 - Verteilter Workflow

Ein erster Ausgangspunkt wäre, sich eine Prozess-Choreographie wie einen verteilten Workflow vorzustellen. Demzufolge würde es sich dann um einen Workflow handeln, der verteilt durch verschiedene Server ausgeführt wird. Die Abbildung 4.1 zeigt zur Verdeutlichung ein WfMS, welches einen "kooperierenden" Workflow auf drei Server verteilt. Ein Workflow wird normalerweise verteilt ausgeführt, um z.B. eine parallele Verarbeitung zu ermöglichen. Bei Verwendung dieses Ansatzes könnte man praktischerweise auf die Konzepte und Lösungen, die im Bereich der verteilten Workflows zur Adaptivität bereits erarbeitet wurden [BRD01c, BRD01b], zurückgreifen. Allerdings gibt es entscheidende Unterschiede zwischen einer Choreographie und einem verteilten Workflow, weshalb Ansatz 1 in dieser Arbeit auch nicht weiter verfolgt wird und hier nur zur Abgrenzung dienen soll.

Bei einem verteilten Workflow wird die Kontrolle über die Ausführung eines Workflows komplett übergeben. Es sind zwar auch parallele Verarbeitungen möglich, wobei mehrere Server gleichzeitig an der Ausführung einer Workflow-Instanz beteiligt sind, aber dann kommt es nicht noch zusätzlich zu einer "Kommunikation" zwischen den parallelen Abschnitten wie es in Abbildung 4.1 der Fall wäre. Da alle Aktivitäten in demselben WfMS verwaltet werden, wäre es auch nicht möglich heterogene Anwendungen in eine Choreographie einzubinden. Außerdem ließe sich bei diesem Ansatz auch die Autonomie der Partner nicht realisieren.

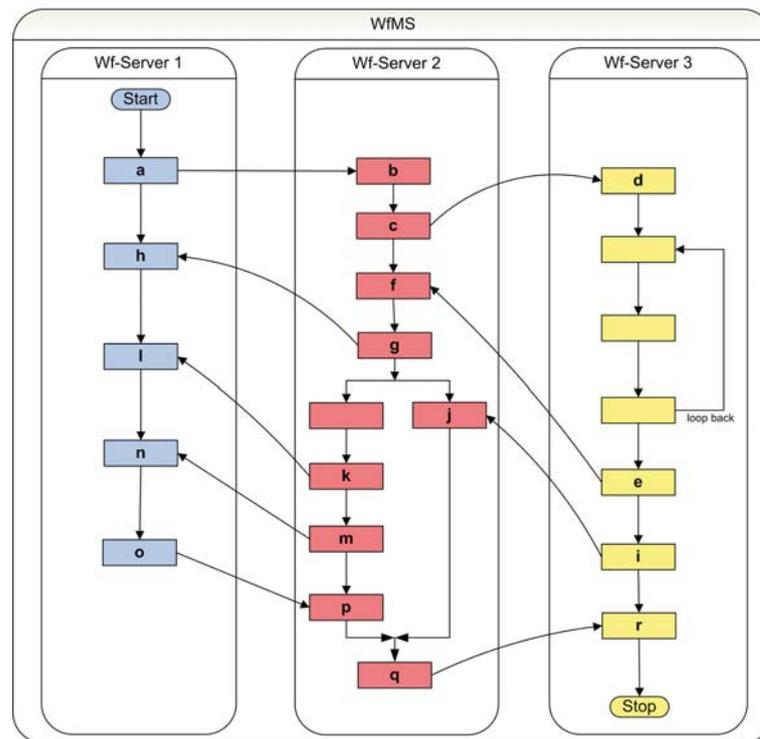


Abbildung 4.1: Verteilter Workflow

4.2 Ansatz 2 - Inhouse Choreographie

Obwohl die Autonomie der Partner in einer Prozess-Choreographie ein wichtiges Merkmal ist, kann es Szenarien geben, bei denen auch der private Workflow öffentlich ist. Vor allem bei größeren Unternehmen kann es vorkommen, dass Prozesse verschiedener WfMSs zusammenarbeiten müssen, z.B. weil jede Abteilung ihr eigenes WfMS hat, aber gemeinsam an einem Projekt arbeiten. Dadurch, dass die privaten Workflows der Partner hier bekannt sein dürfen, können auch die Auswirkungen von Änderungen auf die Partner erkannt werden.

Als zweiten Ansatz wird infolgedessen von einer Prozess-Choreographie ausgegangen, bei der jedem Partner die gesamte Struktur der Choreographie mit allen Aktivitäten vorliegt. Bei Ansatz 2 besteht deshalb die Möglichkeit, anhand eines globalen Zustandsbilds eine Entscheidung über die Verträglichkeit einer Instanz zu treffen. Trotzdem muss für die Entscheidung über eine Änderung bei einer laufenden Choreographie-Instanz eine Synchronisierung der aktuellen Zustände erfolgen. Der blaue Partner in Abbildung 4.2 weiß zu einem Zeitpunkt, bei dem er die Aktivität a bereits beendet hat, aber die Aktivität h noch nicht gestartet wurde, ob die Aktivität

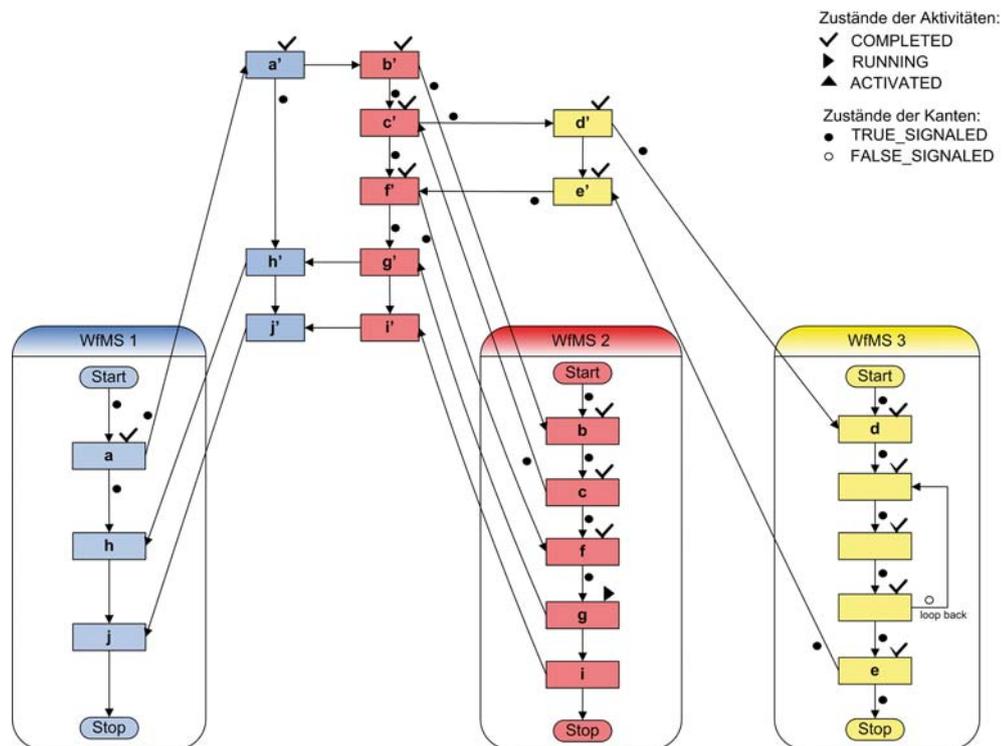


Abbildung 4.3: Choreographie mit globalem öffentlichen Prozess

In Abbildung 4.3 sind auch die Zustände einer Choreographie-Instanz angegeben. Die Zustände werden an die ADEPT-Notation angelehnt, die in Kapitel 2.3 vorgestellt wurde. Bei einer Prozess-Choreographie haben aber nicht nur die Aktivitäten der privaten Workflows Zustände, sondern auch die öffentlichen Aktivitäten. Wobei die öffentlichen Aktivitäten initial den Zustand NOT_ACTIVATED haben und dann in den Zustand COMPLETED übergehen, wenn sie ausgeführt worden sind. Die Zustände ACTIVATED und RUNNING werden im Gegensatz zu den privaten Aktivitäten nicht benötigt, da eine Nachricht entweder verschickt bzw. empfangen worden ist oder eben noch nicht.

4.3.1 Subtraktive Änderungsoperation

Anhand des folgenden Beispiels soll aufgezeigt werden, wie es zu Konflikten kommen kann, wenn ein Partner eine für seinen privaten Workflow und den globalen öffentlichen Prozess korrekte Änderung vornimmt, aber den privaten Workflow des Partners nicht berücksichtigt.

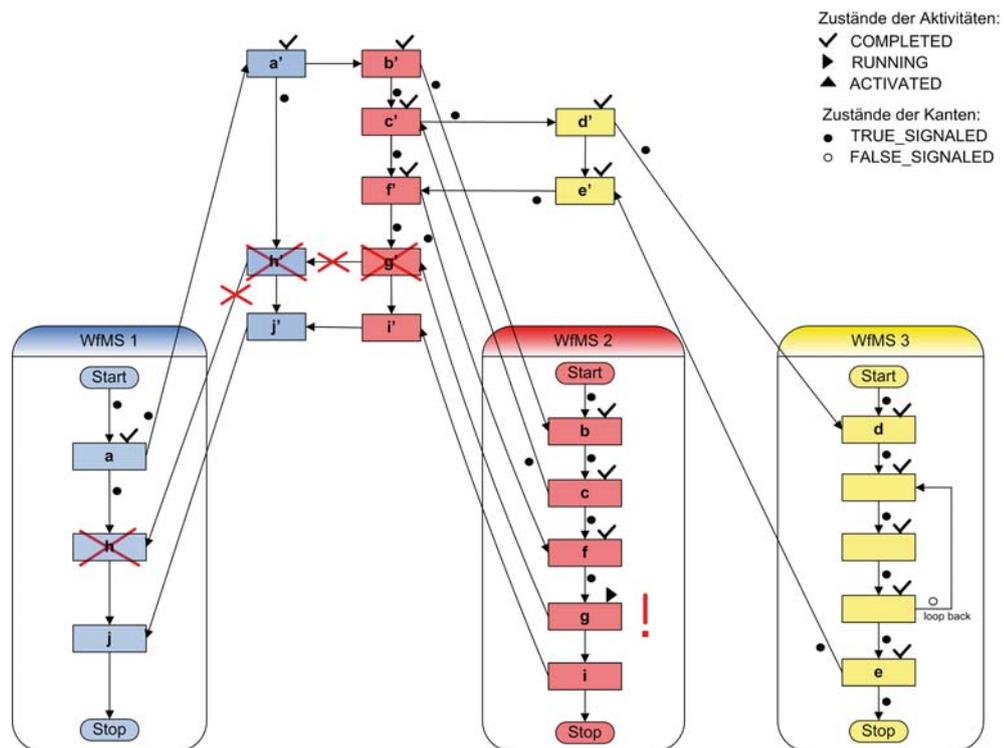


Abbildung 4.4: Löschen der Aktivität h

Ausgehend von der Choreographie-Instanz in Abbildung 4.3 möchte der blaue Partner seine kooperative Aktivität h löschen. Dadurch muss auch seine öffentliche Aktivität h' sowie die zugehörige Kante gelöscht werden, wie in Abbildung 4.4 dargestellt. Dies wiederum führt dazu, dass auch die entsprechende öffentliche Aktivität g' des roten Partners entfernt werden muss, um strukturell wieder eine korrekte Choreographie zu bekommen. Das Löschen von g' hat aber auch Auswirkungen auf den privaten Workflow des roten Partners und zwar auf die kooperative Aktivität g. Das Problem ist nun, dass der blaue Partner nicht weiß, wie weit der rote Partner in der Ausführung seines privaten Workflows vorangeschritten ist und ob die Auswirkungen seiner Löschoption der Aktivität h eventuell zu Problemen führen könnten. Tatsächlich befindet sich in Abbildung 4.4 die Aktivität g bereits im Zustand RUNNING und der rote Partner kann deshalb g nicht mehr löschen.

4.3.2 Additive Änderungsoperation

Für das folgende Beispiel dient als Ausgangssituation wieder die Choreographie-Instanz in Abbildung 4.3. Allerdings hat die Aktivität g jetzt den Zustand ACTIVATED und noch nicht

RUNNING. Der rote Partner fügt in Abbildung 4.5 die kooperative Aktivität x, zwischen den Aktivitäten f und g, in seinem privaten Workflow ein. Diese Operation ist aus Sicht von Rot korrekt, da die Aktivität g den Zustand ACTIVATED hat und noch nicht läuft. Auf die Korrektheitskriterien für Änderungsoperationen wird in Kapitel 5 noch eingegangen. Daraufhin wird auch der globale öffentliche Prozess angepasst und die öffentlichen Aktivitäten x' und y' eingebunden. Nun muss der blaue Partner noch seinen privaten Workflow anpassen und die kooperative Aktivität y zwischen den Aktivitäten a und h einfügen. An diesem Beispiel kann auch aufgezeigt werden, dass eine Änderungsdurchführung nicht nur beinhaltet wie Aktivitäten hinzugefügt bzw. entfernt werden. Anschließend müssen auch noch die Zustände der Aktivitäten angepasst werden. Die Aktivität g des roten Partners wird vom Zustand ACTIVATED auf den Zustand NOT_ACTIVATED zurückgesetzt. Dafür bekommt die neu eingefügte Aktivität y des roten Partners gleich den Zustand ACTIVATED zugewiesen. Auch müssen die Zustände der Kanten, wie in Abbildung 4.5 dargestellt, entsprechend angepasst werden.

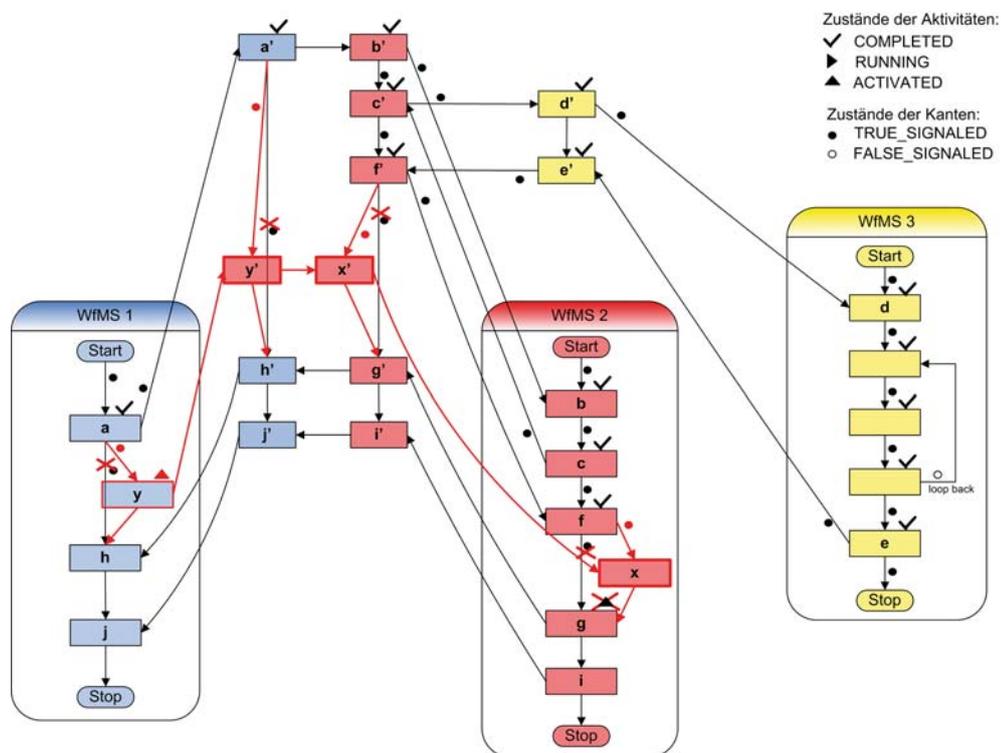


Abbildung 4.5: Einfügen der Aktivität x

4.4 Ansatz 4 - Choreographie mit verteilten öffentlichen Prozessen

Bei diesem vierten Ansatz gibt es mehrere öffentliche Prozesse, welche jeweils die bilaterale Kooperation zwischen zwei Partnern repräsentieren. Eine Prozess-Choreographie-Instanz nach Ansatz 4 ist in Abbildung 4.6 dargestellt. Eine multilaterale Zusammenarbeit entsteht demzufolge durch mehrere bilaterale öffentliche Prozesse. Ein solches Szenario basiert auf einer Bottom-Up-Entstehung. Es ermöglicht das dynamische Ein- und Ausgliedern neuer Partner während einer bereits bestehenden Kooperation. Die Auswirkungen lokaler Änderungen sind bei Ansatz 4 noch schwieriger abzuschätzen. Der Grund dafür ist, dass die öffentlichen Prozesse verteilt sind und dadurch weiß z.B. der blaue Partner in Abbildung 4.6 nichts von der Existenz des gelben Partners, aber beide kooperieren mit dem roten Partner innerhalb derselben Choreographie-Instanz.

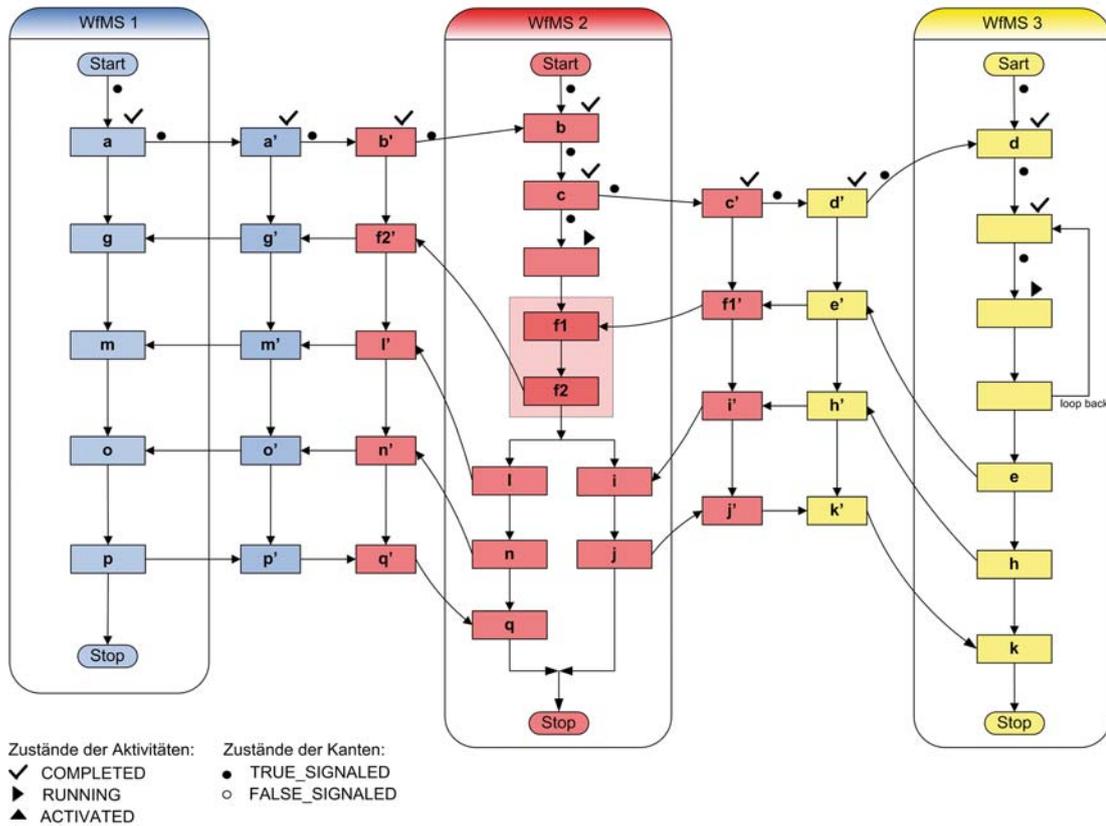


Abbildung 4.6: Choreographie mit verteilten öffentlichen Prozessen

Um eine möglichst einfache Darstellung zu ermöglichen, stehen die kooperativen Aktivitäten über ihre öffentlichen Aktivitäten entweder in einer sendenden oder einer empfangenden Funktion mit anderen Workflows in Verbindung. Es könnten aber auch mehrere logischen Schritte innerhalb einer Aktivität ablaufen, wie in Abbildung 4.6 bei der Aktivität f des roten Partners. Hier findet sowohl eine Kommunikation mit dem gelben Partner durch f1, als auch eine Kommunikation mit dem blauen Partner durch f2 statt. Durch die Aufteilung der Aktivität f soll verdeutlicht werden, dass innerhalb des privaten roten Workflows f1 nur in Verbindung mit f2 existiert und umgekehrt.

4.4.1 Kaskadierende subtraktive Änderungsoperation

Kaskadierende Änderungen breiten sich über mehrere Partner aus. Somit kann es passieren, dass eine Änderung Auswirkungen auf den privaten Workflow eines Partners hat, der zwar auch in die Choreographie involviert ist, aber nicht in direkter Beziehung zum Initiator der Änderung steht.

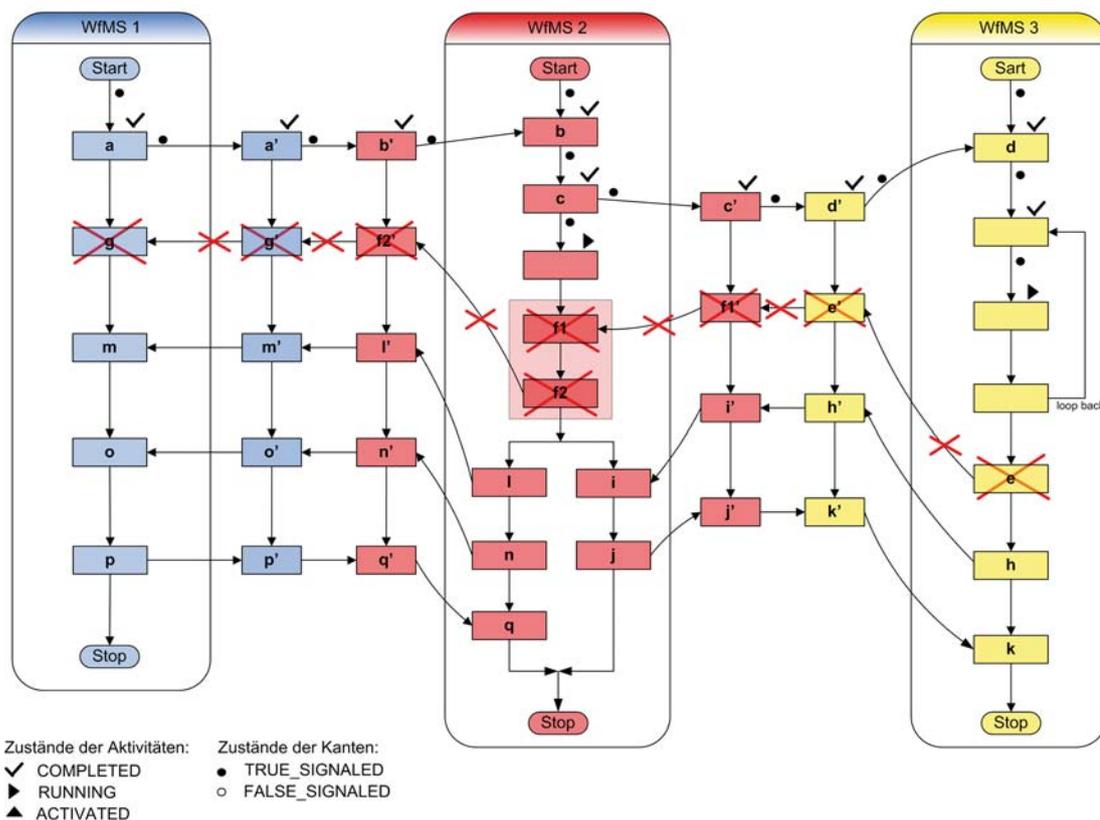


Abbildung 4.7: Löschen der Aktivität e

Ausgehend von der Choreographie-Instanz in Abbildung 4.6 wird die kooperative Aktivität e im privaten Workflow des gelben Partners gelöscht. Dies führt, wie in Abbildung 4.7 dargestellt, zum Entfernen der öffentlichen Aktivitäten e' und $f1'$ des öffentlichen Prozesses zwischen dem gelben und roten Partner. Da die private Aktivität $f1$ aber, wie in Kapitel 4.4 beschrieben, in Verbindung steht mit der Aktivität $f2$, muss diese ebenfalls gelöscht werden. Da $f2$ aber wiederum eine kooperative Aktivität ist, kommt es auch zu Auswirkungen im öffentlichen Prozess zwischen dem roten und blauen Partner. Infolgedessen sind auch noch die beiden öffentlichen Aktivitäten $f2'$ und g' , sowie die kooperative Aktivität g des privaten blauen Workflows betroffen. Es kann also durchaus vorkommen, dass eine Änderung im privaten Workflow eines Partners Auswirkungen auf den privaten Workflow eines Teilnehmers der Choreographie hat, welcher überhaupt nicht in bilateraler Beziehung zum Auslöser steht und es wie in Abbildung 4.7 zu einem kaskadierenden Löschen über mehrere Partner hinweg kommt.

4.4.2 Ablaufänderung

Bei diesem Beispiel wird lediglich eine einfache lokale Änderung im Ablauf vorgenommen, welche trotzdem globale Auswirkungen auf die Choreographie hat.

Ausgehend von Abbildung 4.6 verändert der rote Partner die Abfolge seiner privaten Aktivitäten, indem er die Aktivität j , wie in Abbildung 4.8 dargestellt, nach unten zieht. Diese Änderung seines privaten Workflows hat keine weiteren Änderungen bei den Partnern zur Folge, weder im öffentlichen Prozess mit dem roten noch mit dem grünen Partner. Und trotzdem hat die lokale Änderung auch Auswirkungen auf die anderen Partner, denn es werden dadurch neue globale Abhängigkeiten geschaffen. Der gelbe Partner muss nun warten bis der rote Partner seine Kommunikation mit dem blauen Partner abgeschlossen hat, bevor er seinen privaten Workflow beenden kann. Durch solche neuen Abhängigkeiten können durchaus Probleme entstehen, da sie die Ausführung eines Workflows verzögern können und damit z.B. Liefertermine nicht eingehalten werden können.

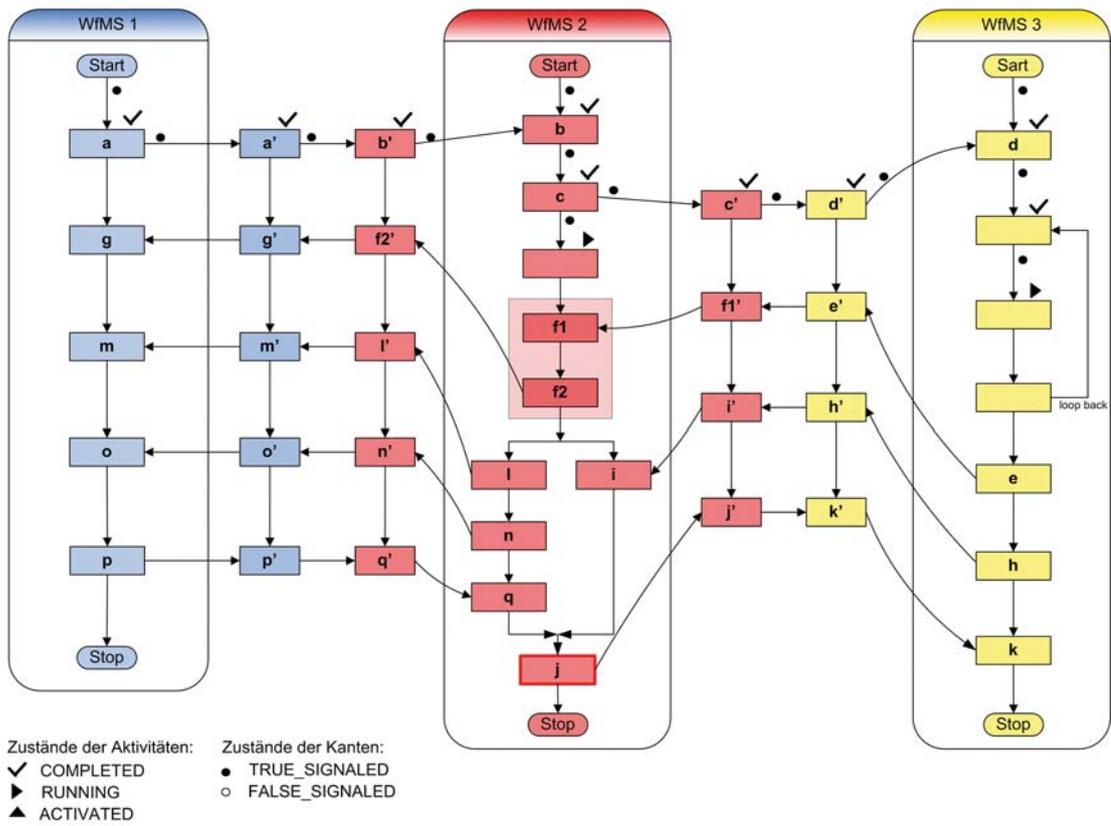


Abbildung 4.8: Verschieben der Aktivität j

5 Korrektheit bei dynamischen Änderungen

Prinzipiell laufen alle dynamischen Änderungen in zwei Schritten ab. Als erster Schritt wird anhand der Struktur und des Zustands der WF-Instanz überprüft, ob eine Änderung zulässig ist. Falls dem so ist, wird als zweiter Schritt der Ausführungsgraph der WF-Instanz entsprechend modifiziert. Für die Überprüfung der Zulässigkeit einer Änderung wird normalerweise der globale aktuelle Zustand der WF-Instanz herangezogen. Um entscheiden zu können, ob eine Instanz mit einem geänderten Schema verträglich ist, wird außerdem ein Kriterium benötigt, anhand dessen überprüft werden kann, ob eine laufende Instanz sich nach der Migration wieder in einem korrekten Zustand befindet und es auch in der Folge nicht zu Fehlern kommen kann. Für den zentralen Fall, bei dem die Instanz eines autarken Workflows betrachtet wird, wurde ein solches Korrektheitskriterium bereits definiert und wird in Kapitel 5.1 vorgestellt. Inwiefern sich dieses Kriterium auch auf den verteilten Fall in einer Choreographie anwenden lässt, wird in Kapitel 5.2 besprochen.

Bei einer Choreographie-Instanz müssen, im Vergleich zum zentralen Fall, noch zwei zusätzliche Aspekte berücksichtigt werden. Erstens gibt es durch die Verteilung keinen aktuellen globalen Zustand der Choreographie. Dafür würde man die Zustände der Aktivitäten zur exakt gleichen Zeit benötigen und das ist, wie aus den verteilten Systemen bekannt ist, ohne zusätzliche Mechanismen nicht möglich [TS03]. Doch selbst wenn man mit einer geeigneten Methoden die aktuellen Zustände synchronisiert, gibt es bei Choreographie-Instanzen nach Ansatz 3 und 4 noch ein weiteres Problem. Der zweite Aspekt ergibt sich nämlich aus der Autonomie der Partner, dadurch sind nicht alle benötigten Informationen verfügbar, um die Auswirkungen der Änderung überschauen zu können. Verschiedene Möglichkeiten der Anwendung und der Zusicherung des Korrektheitskriterium in einer Choreographie-Instanz werden in Kapitel 5.3 vorgestellt. In den darauf folgenden Kapiteln werden diese Möglichkeiten dann anschließend ausführlich diskutiert.

5.1 Korrektheitskriterium für den zentralen Fall

Die Korrektheit für den statischen Fall wird durch formale Bedingungen sichergestellt und dadurch resultiert auf struktureller Ebene aus einem korrekten Schema wieder ein korrektes (geändertes) Schema. Ein geändertes Schema kann genau dann auf eine WF-Instanz propagiert werden, falls es nicht widersprüchlich zur bisherigen Ausführung ist, d.h. der gleiche Ablauf wäre mit dem geänderten Schema auch möglich gewesen. Für jede Instanz gibt es eine Ablaufhistorie, in welcher unter anderem Informationen zum Start und Ende von Aktivitäten protokolliert werden. Naheliegender wäre es nun den bisherigen Ablauf bei der Ausführung der Aktivität anhand des geänderten Schemas nochmals durchzuspielen. Falls dies möglich ist, ist die Verträglichkeit der WF-Instanz mit dem geänderten Schema garantiert. Dieser Ansatz wurde aber nicht nur aus Performance-Aspekten nicht weiter verfolgt, sondern er wäre auch für Instanzen mit Schleifen zu restriktiv [RRD04].

Infolgedessen wurden Korrektheitskriterien entwickelt, welche auf den Zustandsinformationen basieren (eine ausführliche und formale Beschreibung findet sich in [RRD02]). Geplante Änderungen können nach folgenden Kriterien beurteilt werden:

- **Additive Änderungsoperation:** Beim Einfügen von Aktivitäten kann Verträglichkeit zugesichert werden, wenn sich die (direkten) Nachfolger des neu eingefügten Schritts X aktuell in einem der beiden Zustände `ACTIVATED` oder `NOT ACTIVATED` befinden. Für das Einfügen einer Kontrollkante kann man Verträglichkeit zusichern, wenn die Zielaktivität der Kante noch nicht gestartet wurde.
- **Subtraktive Änderungsoperationen:** Es dürfen nur Aktivitäten gelöscht werden, die sich nicht im Zustand `RUNNING` oder `COMPLETED` befinden.

Ordnungsverändernde Operationen lassen sich prinzipiell auf das Einfügen und Löschen von Aktivitäten zurückführen.

Für diese Arbeit werden nur die Verträglichkeitsbedingungen für Kontrollflussänderungen vorgestellt, weil Datenflüsse bei der gewählten Darstellung für die Prozess-Choreographien nicht mit eingeschlossen werden (vgl. für die Datenflussänderungen im zentralen Fall die Ausführungen von [RRD02]).

5.2 Korrektheitskriterium für den verteilten Fall in einer Prozess-Choreographie

Die Änderungen auf Schemaebene werden für eine Prozess-Choreographie in [RWR06] diskutiert. Wie im zentralen Fall, wird aber zusätzlich noch ein Korrektheitskriterium benötigt, um

zu überprüfen, ob laufende Choreographie-Instanzen verträglich mit dem geänderten Schema sind und auf dieses migrieren können.

In einer laufenden Prozess-Choreographie gibt es mehrere private Instanzen, welche von einer Änderung betroffen sein können. Diese privaten Instanzen müssen demzufolge miteinbezogen werden, um eine Verträglichkeitsentscheidung für die umfassende Choreographie-Instanz treffen zu können. Es ist naheliegend, dass die Entscheidung über die Verträglichkeit der privaten WF-Instanzen nur lokal bei dem jeweiligen Partner getroffen werden kann, denn nur dieser kennt die genaue Struktur und den aktuellen Zustand seiner privaten Instanz. Es kann aber davon ausgegangen werden, dass jeder Partner für seine private WF-Instanz anhand des Korrektheitskriteriums für den zentralen Fall entscheiden kann, ob die private Instanz mit dem geänderten Schema verträglich ist oder nicht.

Darüber hinaus stellt sich aber noch die Frage nach der Verträglichkeit der Choreographie-Instanz. Das lokale Korrektheitskriterium für den zentralen Fall kann für die Choreographie-Instanz nicht angewendet werden, da diese einerseits mehrere lokale Instanzen umfasst, welche verteilt und autonom ausgeführt werden, und andererseits auch noch die öffentlichen Aktivitäten enthält. Von der Situation ausgehend, dass die privaten Instanzen bei den Partnern auf Verträglichkeit anhand des lokalen Korrektheitskriteriums überprüft werden können, kann eine Choreographie-Instanz genau dann auf ein geändertes Schema migrieren, falls die lokalen Instanzen aller von der Änderung betroffenen Partner verträglich mit dem geänderten Schema sind. Gleichwohl gehören auch die öffentlichen Prozesse zu einer Choreographie-Instanz. Wird deshalb nicht noch ein Kriterium benötigt, welches die Zustände der öffentlichen Aktivitäten berücksichtigt? Hierbei kann aber auf die Tatsache zurückgegriffen werden, dass die öffentlichen Aktivitäten eine Verbindung von einem zum anderen Partner darstellen. Falls eine öffentliche Aktivität den Zustand COMPLETED hat, ist die entsprechende sendende private Aktivität auch immer bereits im Zustand COMPLETED, weil die Nachrichten erst bei Beendigung der privaten Aktivitäten verschickt werden. Ein möglicher Konflikt einer Änderung bei einer öffentlichen Aktivität führt deshalb auch immer zu einem Konflikt bei einer privaten WF-Instanz. Dieser Sachverhalt soll anhand des folgenden Beispiels verdeutlicht werden.

Die Abbildung 5.1 zeigt eine Choreographie-Instanz mit zwei Partnern und einem öffentlichen Prozess. Partner 1 möchte seine kooperative Aktivität d löschen. Diese Änderung ist für den privaten Workflow von Partner 1 korrekt und dessen private Instanz wäre auch verträglich mit dem Löschen der Aktivität d, da sich diese im Zustand NOT_ACTIVATED befindet. In der Folge würden auch die öffentlichen Aktivitäten d' und c' gelöscht werden, dass Aktivität c' bereits den Zustand COMPLETED hat wird erst einmal nicht berücksichtigt. Da aber auch die Aktivität c des privaten Workflows von Partner 2 von der Änderung betroffen ist und diese Aktivität bereits den Zustand COMPLETED hat, kann die private Instanz von Partner 2 nicht migrieren und damit auch nicht die Choreographie-Instanz.

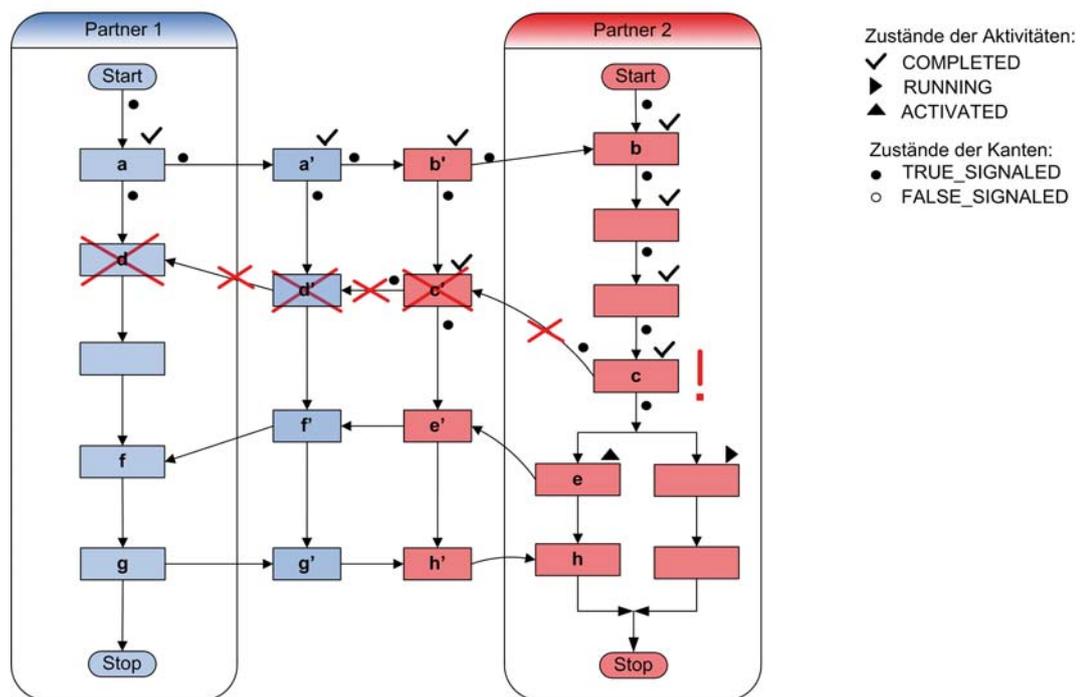


Abbildung 5.1: Private Instanz erkennt Konflikt

Die vorherigen Überlegungen gelten auch für eine Choreographie mit Schleifen, selbst wenn die Schleifen kooperative Aktivitäten enthalten. Diese Feststellung wiederum beruht auf der Gegebenheit, dass die Schleifenrücksprungbedingung synchronisiert wird. Wie in Kapitel 3.3 bereits angesprochen, setzen die privaten Instanzen die Aktivitäten- und Knotenzustände in der Schleife für eine neue Iteration auf NOT_ACTIVATED bzw. NOT_SIGNED zurück. Gleichmaßen wird natürlich auch bei den öffentlichen Aktivitäten und deren Kanten vorgegangen, so dass auch hier die Konsistenz der Choreographie-Instanz gegeben ist. Wie im zentralen Fall, wird die Restriktion, dass es durch eine Änderung nicht zu einer Verfälschung der Vergangenheit kommen darf, bei Schleifen etwas gelockert. Durch Anwendung des lokalen Korrektheitskriteriums wird immer nur der aktuelle Schleifendurchlauf herangezogen. Ansonsten wären Änderungen, die Aktivitäten innerhalb der Schleife betreffen, kaum möglich.

Aufgrund dieser Erkenntnisse wird kein zusätzliches Korrektheitskriterium für die öffentlichen Aktivitäten in einer Prozess-Choreographie benötigt. Dadurch wurde auf Basis des lokalen Korrektheitskriteriums für den zentralen Fall eine relativ unkomplizierte Möglichkeit gefunden, die Verträglichkeit einer Choreographie-Instanz mit einer Änderung zu überprüfen.

5.3 Vorgehen bei der Durchführung von Änderungen

Ein globales Zustandsbild der Choreographie existiert aufgrund der Verteilung und der Autonomie der Partner (mit Ausnahme von Ansatz 2) nicht. Deshalb muss jeder Partner für seinen privaten Workflow prüfen, ob dieser mit der Änderung verträglich ist. Jeder Partner sollte in folgedessen über die gewünschte Änderung informiert werden, seine private Instanz auf Verträglichkeit prüfen und berichten, ob er migrieren kann oder nicht. Nachdem die Entscheidung getroffen wurde, ob die gesamte Choreographie migrieren kann, müssen die Partner benachrichtigt werden, damit sie die Änderungen für ihre private WF-Instanz vornehmen können und die Choreographie sich danach wieder in einem korrekten Zustand befindet. Die verteilte Entscheidungsfindung, ob alle privaten Instanzen mit den Änderungen verträglich sind, und die verteilte Migration der Instanzen muss koordiniert werden. Ansonsten kann es zu Fehlern wie Inkonsistenzen oder Deadlocks kommen. Dabei gilt das “Alles-oder-Nichts-Prinzip“, d.h. entweder migrieren alle Partner oder keiner. Prinzipiell kann kein Partner gezwungen werden zu migrieren, selbst wenn es für seine private WF-Instanz die Migration theoretisch möglich wäre. Trotzdem muss sichergestellt werden, dass falls ein Partner einmal der Änderung zugestimmt hat, er auch tatsächlich migriert, wenn er nach der Entscheidungsfindung dazu aufgefordert wird. Außerdem darf sich der Zustand der privaten WF-Instanz eines Partners, von der Zeitspanne der Entscheidungsfindung bis zur tatsächlichen Migration, nicht insofern ändern, als dass sich die WF-Instanz, aufgrund des in der Zwischenzeit fortgeschrittenen Zustandes, nach der Migration nicht mehr in einem korrekten Zustand befindet.

Die Durchführung dynamischer Änderungen gestaltet sich bei einer Choreographie-Instanz komplexer als im zentralen Fall. Zwischen den Partnern muss koordiniert und kommuniziert werden. Die im zentralen Fall aufgeführte Einteilung bei dynamischen Änderungen in zwei Schritte, Prüfung der Zulässigkeit der Änderung und Modifikation des Ausführungsgraphen, wird im Folgenden im Kontext einer Choreographie erweitert. Für die Durchführung einer Änderungsoperation bei einer Choreographie-Instanz werden die nachstehenden Schritte ausgeführt:

1. Initiierung: Der Initiator einer Änderung prüft zuerst, ob die Änderung für seinen privaten Workflow strukturell zulässig ist und ob die laufende Instanz aufgrund ihres aktuellen Zustandes verträglich mit der Änderung ist. Falls die Änderung keine kooperativen Aktivitäten betrifft, kann sie lokal durchgeführt werden. Ansonsten müssen die betroffenen Partner mit einbezogen werden.

2. Lokale Entscheidung: Bevor der Initiator der Änderung die Partner befragt, kann er eventuell über die für ihn bereits vorhandenen Informationen eine Vorentscheidung treffen. Das Befragen der Partner ist "teuer", besonders wenn alle Workflows für die Entscheidungsfindung angehalten werden müssen. Falls aus der Vorentscheidung hervorgeht, dass ein Partner nicht migrieren kann, wird an dieser Stelle abgebrochen.
3. Globale Entscheidung: Die Partner werden nun befragt, ob die Änderung für ihren lokalen Workflow strukturell und vom Zustand der privaten Instanz her möglich ist. Falls nicht alle Partner der Änderung zustimmen, wird für alle die Änderungsoperation abgebrochen.
4. Migration: Alle Partner modifizieren ihren Ausführungsgraph und ihre öffentlichen Aktivitäten entsprechend und passen eventuell die Zustände an.

Bei dieser Vorgehensweise wird nicht unterschieden, ob es sich bei der strukturellen Änderung um eine Schemaänderung oder um eine Ad-Hoc-Änderung, d.h. eine Änderung welche nur eine Choreographie-Instanz betrifft, handelt. Diese Unterscheidung ist für die Durchführung der Änderung erstmal unerheblich, da hier die Verträglichkeit einer laufenden Choreographie-Instanz betrachtet wird. Ob die Änderung nur diese Choreographie-Instanz betrifft oder auch noch andere macht keinen Unterschied, da alle laufenden Choreographie-Instanzen einzeln auf Verträglichkeit mit der Änderung überprüft werden müssen.

In den folgenden Kapiteln werden Entscheidungs- und Migrationsverfahren vorgestellt und anhand der verschiedenen Ansätze aus Kapitel 4, die Vor- und Nachteile der Verfahren diskutiert. Die Abbildung 5.2 gibt einen Überblick über die Verfahren in Verbindung mit den Ansätzen 2 bis 4. Die Entscheidungsverfahren wurden aufgeteilt in lokale Entscheidungsverfahren (Kapitel 6), anhand deren mit den Informationen eines Teilnehmers eventuell bereits eine Vorentscheidung getroffen werden kann, und globale Entscheidungsverfahren (Kapitel 7), bei denen die Partner zu der Änderung befragt werden. Die Migrationsverfahren (Kapitel 8) beschreiben die Durchführung der Änderung.

	Ansatz 2	Ansatz 3	Ansatz 4
Lokale Entscheidungsverfahren*		Vorentscheidung evtl. möglich...	
Mittiger Entscheider – Bed. alle involvierten Partner stehen in bilateraler Beziehung mit nur dieser Einheit.	Nicht sinnvoll.	...ohne mit den Partnern kommunizieren zu müssen	
Verkettete Entscheidung - die Vorentscheidung muss über einen/mehrere Partner hinweg getroffen werden.		...ohne mit den Partnern kommunizieren zu müssen, aber in weniger Fällen.	...,aber Kommunikation mit den Partnern nötig – weniger sinnvoll.
Globale Entscheidungsverfahren**	Für die Entscheidungsfindung ist eine Kommunikation mit den Partnern nötig.		
Zentraler Änderungsmanager – zusätzliche Einheit ist für alle Änderungswünsche zuständig.	Entscheidung anhand eines globalen Zustandsbilds möglich.	Alle involvierten Partner werden befragt. Keine Kollision von Änderungen.	Aufwendig, da involvierte Partner ermittelt werden müssen. Keine Kollision von Änderungen.
Zentraler Koordinator – Initiator der Änderung kommuniziert direkt mit allen involvierten Partnern.	Entscheidung anhand eines globalen Zustandsbilds möglich. Evtl. parallele Änderungen.	Alle involvierten Partner werden befragt. Evtl. parallele Änderungen.	Nicht möglich.
Verkettete Abstimmung – Entscheidungsfindung wird durchgereicht.	Nicht sinnvoll.	Weniger sinnvoll.	Befragung wird durchgereicht jeder kontaktiert seine direkten Partner. Evtl. parallele Änderungen.
Migrationsverfahren***	Alle von der Änderung betroffenen Partner müssen migrieren.		
Zentraler Änderungsmanager – direkte Aufforderung der Partner zur Migration.	Sinnvoll, wenn vorhanden.	Evtl. Versionierung des globalen öffentlichen WFs.	Sinnvoll, wenn vorhanden.
Zentraler Koordinator – direkte Aufforderung der Partner zur Migration.	Gesamtstruktur muss bei allen Partnern aktualisiert werden.	Globaler öffentlicher WF muss bei allen Partnern aktualisiert werden.	Möglich, falls bei der verketteten Abstimmung die involvierten Partner bekannt wurden.
Verteilte Durchführung – Aufforderung zur Migration wird durchgereicht.	Nicht sinnvoll.	Weniger sinnvoll.	Verzögerte Migration bei den Partnern.
Optimistisches Verfahren	Nicht sinnvoll, Synchronisierung des Zustandsbilds einfacher.	Rücksetzbarkeit muss gegeben sein.	Rücksetzbarkeit muss gegeben sein. Fehler schwer zu überblicken.

* Basieren auf Markierungsinformationen aus Sicht eines Teilnehmers.

** Alle Partner werden befragt, ob für ihren WF eine Migration möglich ist.

*** Durchführung der Änderung (unterschiedliche Kombinationen mit globalen Entscheidungsverfahren möglich)

Abbildung 5.2: Überblick Entscheidungs- und Migrationsverfahren

6 Lokale Entscheidungsverfahren

Bei den lokalen Entscheidungsverfahren wird versucht, anhand der Informationen eines Teilnehmers, ein Zustandsbild zu erzeugen, anhand dessen auch Aussagen über die Partner getroffen werden können. Dadurch kann durch eine Art Vorentscheidung eventuell bereits feststehen, dass eine Änderung für den Partner nicht möglich ist.

Für dieses Zustandsbild muss erörtert werden, welche Informationen über die Partner verfügbar sind. Die verfügbaren Informationen hängen allerdings zunächst einmal vom zugrundeliegenden Ansatz ab. Bei Ansatz 2 kennt jeder Partner alle Aktivitäten und somit die gesamte Choreographie. Bei diesem Ansatz bietet ein lokales Entscheidungsverfahren keinen bedeutenden Vorteil, weil hier eine einfache Möglichkeit besteht die Zustände der privaten Instanzen zu synchronisieren und somit ein globales Zustandsbild zu erhalten, welches alle Aktivitäten und deren Zustände enthält (darauf wird in Kapitel 7 nochmals eingegangen).

Interessanter wird ein lokales Entscheidungsverfahren für Ansatz 3, bei dem alle involvierten Partner bekannt sind, aber jeder Teilnehmer nur die öffentlichen Aktivitäten der anderen Partner kennt. Wichtig ist, dass zwar die öffentlichen Aktivitäten der Partner bekannt sind, aber nicht deren aktueller Zustand. Trotzdem lassen sich diese Zustände, sowie Informationen über den privaten Workflow der Partner möglicherweise herleiten.

In Abbildung 6.1 wird eine Choreographie-Instanz nach Ansatz 3 mit deren aktuellen Zuständen gezeigt. Alle Informationen, die grau eingefärbt sind, kennt Partner 2 nicht. Alle farbigen Informationen stehen ihm zur Verfügung bzw. lassen sich erschließen. Seine eigenen privaten und öffentlichen Aktivitäten, sowie deren Zustände protokolliert jeder Partner selbst und sind jederzeit abrufbar.

Mit dem Empfang einer Nachricht (z.B. durch die Aktivitäten b' und h' in Abbildung 6.1) kann sich ein Teilnehmer folgende Information über seine direkten Partner (für Partner 2 in Abbildung 6.1 sind das Partner 1 und Partner 3) herleiten:

- Die entsprechenden sendenden öffentlichen Aktivitäten der Partner wurden abgeschlossen (z.B. haben die öffentlichen Aktivitäten a' und g' in Abbildung 6.1 den Zustand COMPLETED).
- Die öffentlichen Aktivitäten der Partner stehen in Verbindung mit kooperativen Aktivitäten des privaten Workflows, welche dann ebenfalls bereits abgeschlossen wurden (in

Abbildung 6.1 haben auch die kooperativen Aktivitäten a und g den Zustand COMPLETED).

- Die vom Teilnehmer bereits vorher an den Partner gesendeten Nachrichten (z.B. durch die Aktivität c' in Abbildung 6.1) wurden von dem Partner empfangen und die korrespondierenden Aktivitäten ausgeführt (die Aktivitäten d und d' in Abbildung 6.1 haben den Zustand COMPLETED).

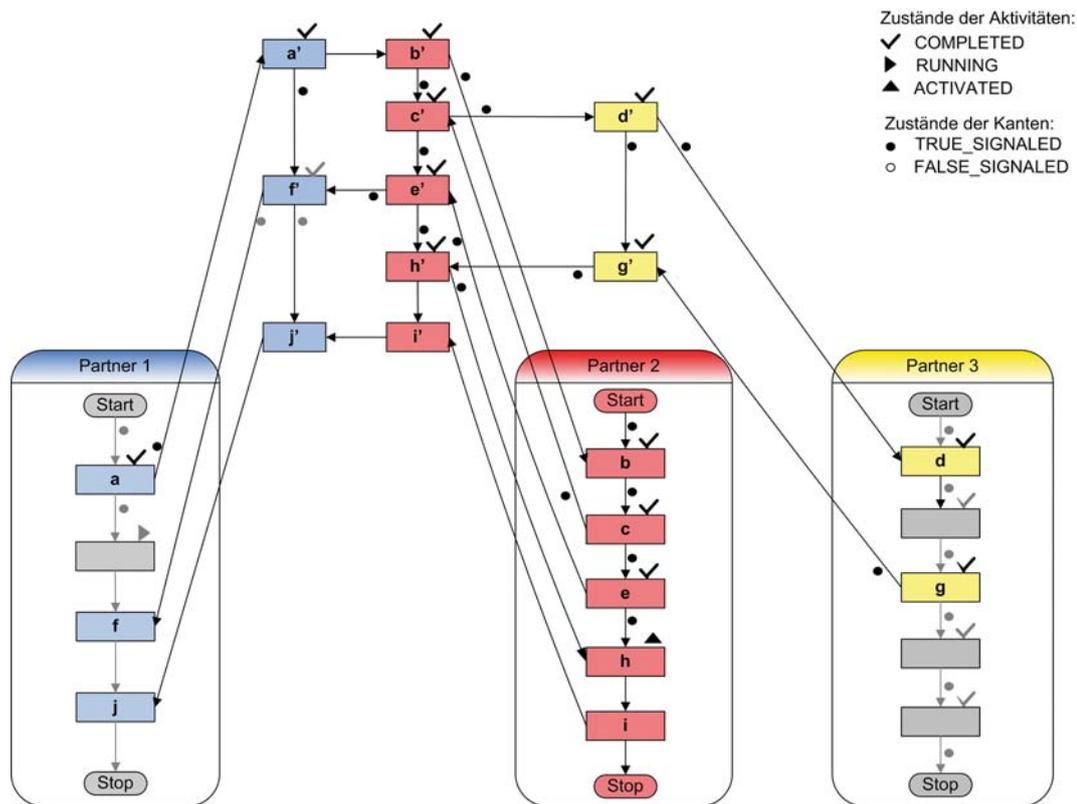


Abbildung 6.1: Zustandsbild aus Sicht des roten Partners

Anhand einer noch nicht an den Partner gesendeten Nachricht (z.B. durch die noch nicht abgeschlossene Aktivität i) erhält ein Teilnehmer folgende Information:

- Die korrespondierende empfangende öffentliche Aktivitäten des Partners hat den Zustand NOT_ACTIVATED (z.B. Aktivität j' in Abbildung 6.1).
- Auch die empfangende öffentliche Aktivität des Partners steht in Verbindung mit einer kooperierenden privaten Aktivität, welche ebenfalls den Zustand NOT_ACTIVATED hat (z.B. Aktivität j in Abbildung 6.1).
- Alle späteren Aktivitäten des Partners sind noch nicht aktiviert.

Vereinfachend ausgedrückt, werden durch den Empfang einer Nachricht vom direkten Partner, die vorher an diesen Partner gesendeten Nachrichten bestätigt und die entsprechenden Aktivitäten wurden bereits abgeschlossen. Desweiteren impliziert eine noch nicht gesendete Nachricht, dass die nachfolgenden Aktivitäten noch nicht laufen können. Diese Überlegungen dienen als Grundlage für das folgende Kapitel 6.1, in dem zusätzliche Markierungen für die Aktivitäten gesetzt werden. In Kapitel 6.2 und 6.3 werden Regeln vorgestellt, nach welchen diese Markierungen den Aktivitäten zugeteilt werden können und inwiefern diese Regeln auf die Ansätze angewendet werden können.

6.1 Markierungen

Zur Erweiterung der bereits bekannten Zustände von Aktivitäten, werden noch zusätzliche Markierungen eingeführt. Anhand dieser Markierungen kann anschließend die Durchführbarkeit einer Änderung aus Sicht eines Teilnehmers eingestuft werden. Es gibt drei verschiedene Markierungen IMPRACTICAL, INSECURE und SECURE, die von einem Teilnehmer für die Aktivitäten gesetzt werden können. Abbildung 6.2 zeigt das Zustandsbild aus Abbildung 6.1 mit den aus Sicht des roten Partner gesetzten Markierungen für die Aktivitäten. Jeder Teilnehmer kann sich ein Zustandsbild herleiten, das auch Informationen über die Partner beinhaltet, sowohl über die öffentlichen, als auch die kooperierenden privaten Aktivitäten. Anhand der Markierungen lassen sich folgende Aussagen über die Aktivitäten treffen.

Aktivitäten mit der Markierung IMPRACTICAL:

- Nicht löschar.
- Einfügen neuer Aktivitäten davor nicht möglich.
- Nicht sicher ob Einfügen neuer Aktivitäten danach möglich.

Aktivitäten mit der Markierung INSECURE:

- Nicht sicher ob löschar.
- Nicht sicher ob Einfügen neuer Aktivitäten davor möglich.
- Nicht sicher ob Einfügen neuer Aktivitäten danach möglich.

Aktivitäten mit der Markierung SECURE:

- Sind löschar.
- Vor diesen können neue Aktivitäten eingefügt werden.
- Nach diesen können neue Aktivitäten eingefügt werden.

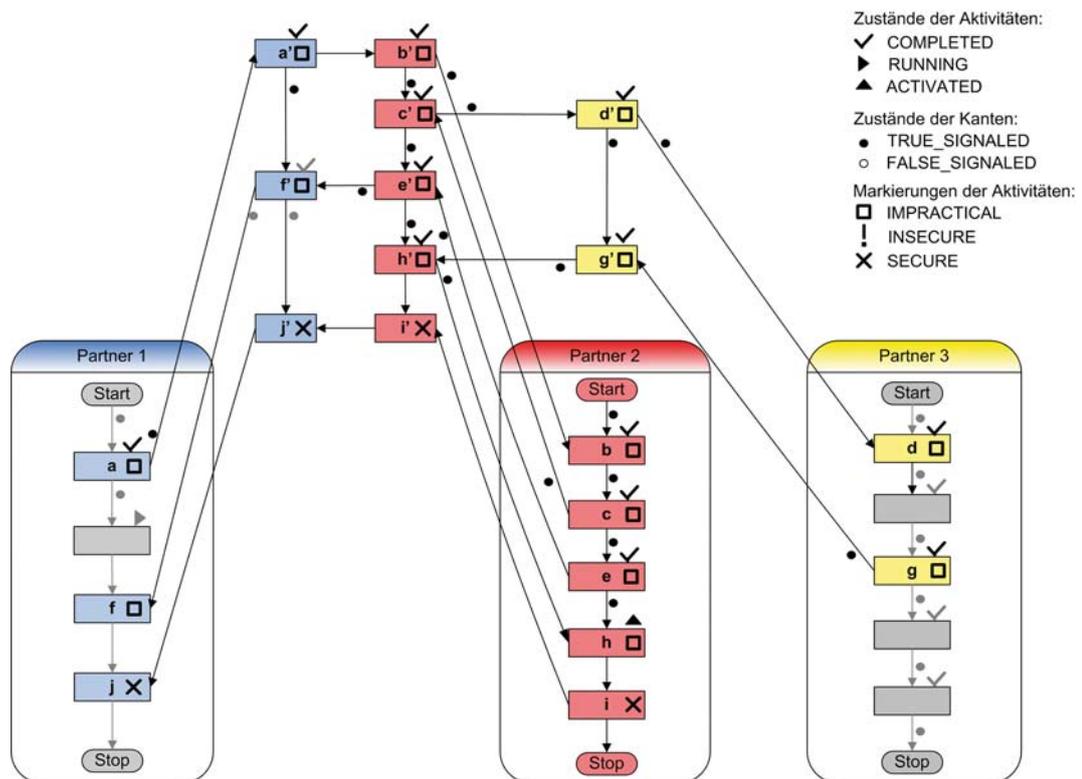


Abbildung 6.2: Zustandsbild mit Markierungen aus Sicht des roten Partners (Ansatz 3)

Über die Durchführbarkeit der gewünschten Änderungen gibt es für einen Teilnehmer anhand der Markierungen somit drei mögliche Vorentscheidungen:

- Änderung nicht möglich.
- Änderung aus Sicht dieses Teilnehmers möglich.
- Vorentscheidung kann nicht getroffen werden.

Durch das Senden und das Empfangen von Nachrichten besteht eine horizontale Abhängigkeit zwischen den Aktivitäten zweier Partner, wie z.B. in Abbildung 6.1 zwischen den Aktivitäten a, a', b' und b oder zwischen c, c', d' und d. Bei den Markierungen ergibt sich daraus die folgende Bedingung: Aktivitäten in horizontaler Abfolge müssen alle die gleichen Markierungen aufweisen. Durch diese Bedingung wird verhindert, dass z.B. die empfangende Aktivität als löscher eingestuft wird und die zugehörige sendende Aktivität nicht.

Anschließend werden die Regeln vorgestellt, nach welchen die Markierungen gesetzt werden können. Dabei muss unterschieden werden in welcher Ausgangslage sich der Entscheider, d.h. der Teilnehmer der eine Vorentscheidung treffen will, gegenüber den anderen Partnern be-

findet. Ein Entscheider, der mit allen anderen Partnern in direkter Beziehung steht, hat mehr Informationen als ein Partner in einer verketteten Konstellation.

6.2 Mittiger Entscheider

Einen mittigen Entscheider kann es sowohl bei Ansatz 3, als auch bei Ansatz 4 geben. Ein Entscheider setzt die Markierungen und kann dadurch eventuell eine Vorentscheidung über die geplante Änderung treffen. Meistens ist der Initiator der Änderung auch der Entscheider, dies ist aber nicht zwingend. In Abbildung 4.3 auf Seite 25 und in Abbildung 4.6 auf Seite 28 wäre jeweils der rote Partner ein mittiger Entscheider, da alle involvierten Partner in bilateraler Beziehung mit nur diesem roten Partner stehen. Eine solche Konstellation findet man vor allem bei sogenannten elektronischen Marktplätzen, wobei alle Teilnehmer mit der zentralen E-Marketplace-Einheit kommunizieren. Aber auch das Beispiel in Abbildung 3.5 auf Seite 19 mit Käufer, Hersteller und Zulieferer entspricht der genannten Bedingung.

Der mittige Entscheider kann nach folgenden Markierungsregeln die Markierungen für die Aktivitäten setzen.

Für die eigenen öffentlichen Aktivitäten:

- Markierung IMPRACTICAL, wenn
 - bei einer sendenden öffentlichen Aktivität die Nachricht an den Partner gesendet wurde, oder
 - bei einer empfangenden öffentlichen Aktivität die Nachricht vom Partner empfangen wurde.
- Markierung INSECURE, wenn eine empfangende öffentliche Aktivität die Nachricht des Partners noch nicht empfangen hat und die vorherige öffentliche Aktivität die Markierung IMPRACTICAL oder INSECURE hat.
- Markierung SECURE, wenn
 - eine sendende öffentliche Aktivität die Nachricht noch nicht geschickt hat, oder
 - bei einer empfangenden öffentlichen Aktivität die Nachricht noch nicht empfangen wurde und die vorhergehende öffentliche Aktivität die Markierung SECURE hat.

Die öffentlichen Aktivitäten der Partner werden gleich wie die entsprechenden eigenen öffentlichen Aktivitäten markiert.

Die privaten kooperativen Aktivitäten des Partners und die eigenen kooperativen Aktivitäten werden gleich wie die zugehörigen öffentlichen Aktivitäten markiert.

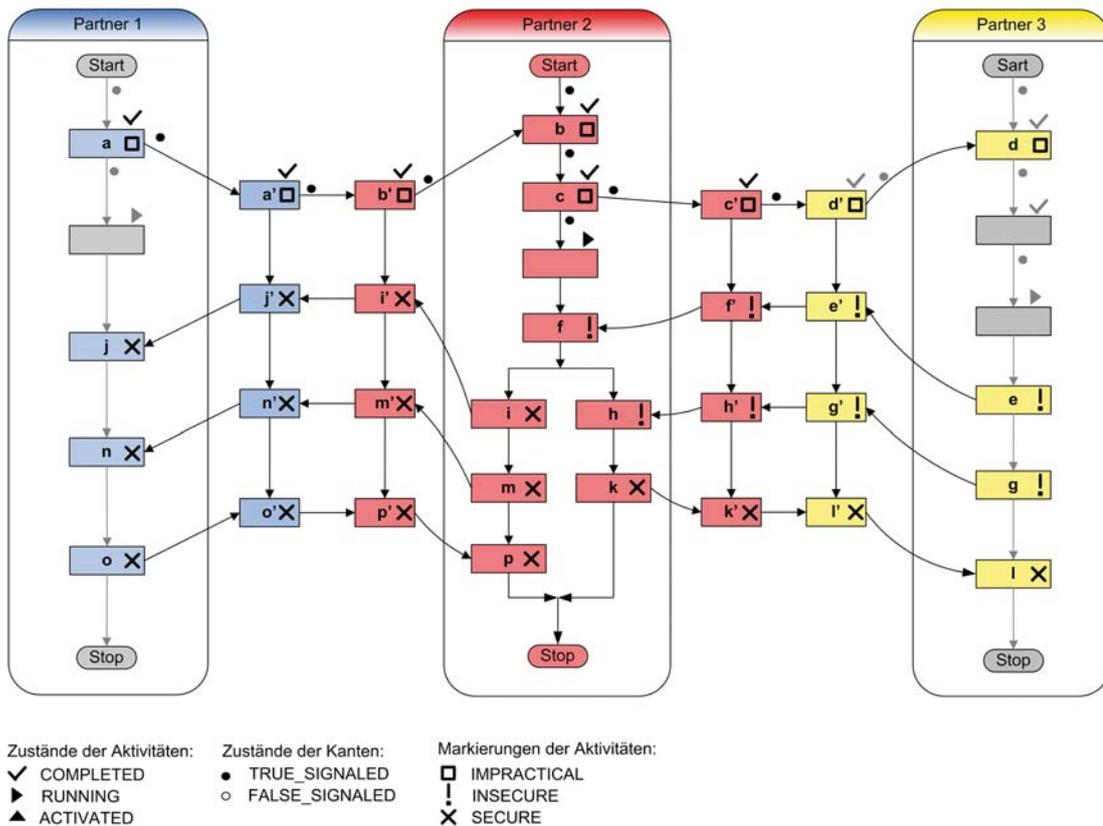


Abbildung 6.3: Zustandsbild mit Markierungen aus Sicht des roten Partners (Ansatz 4)

Die Abbildung 6.3 zeigt ein Zustandsbild einer Choreographie-Instanz nach Ansatz 4 mit den aus Sicht von Partner 2 gesetzten Markierungen. Alle Informationen der Partner 1 und 3, die grau eingefärbt sind, stehen dem roten Partner nicht zur Verfügung. Falls durch eine Änderung eine mit IMPRACTICAL markierte Aktivität gelöscht oder vor ihr eine neue Aktivität eingefügt werden müsste, kann Partner 2 ohne die anderen Partner zu befragen erkennen, dass die Änderung für diese Choreographie-Instanz nicht durchführbar ist. Falls die Änderung eine mit INSECURE markierte Aktivität betrifft, kann der rote Partner keine Aussage darüber machen, wie weit die Ausführung der Aktivitäten beim Partner vorangeschritten ist (z.B. bei den Aktivitäten e und h in Abbildung 6.3) und es eventuell dort zu Problemen kommen könnte. Als nächster Schritt müssen dann die Partner befragt werden, ob ihre privaten Workflows mit der Änderung verträglich sind. Dieser Schritt wird auf jeden Fall auch benötigt falls mit SECURE markierte Aktivitäten von der Änderung betroffen sind. Allerdings stehen die Chancen gut, dass die Änderung durchführbar ist und deshalb könnten aufgrund dieser Information weniger restriktive Anforderungen für das Weiterlaufen der Workflows während der Entscheidungs- und

Migrationsphase gesetzt werden oder auch ein optimistisches Migrationsverfahren in Betracht gezogen werden. Diese Aspekt wird dann in Kapitel 8 nochmals aufgegriffen.

6.3 Verkettete Entscheidung

Bei einer verketteten Entscheidung stehen nicht mehr alle Partner nur in bilateraler Beziehung mit dem Entscheider, sondern hier muss eine Vorentscheidung über einen oder sogar mehrere Partner hinweg getroffen werden.

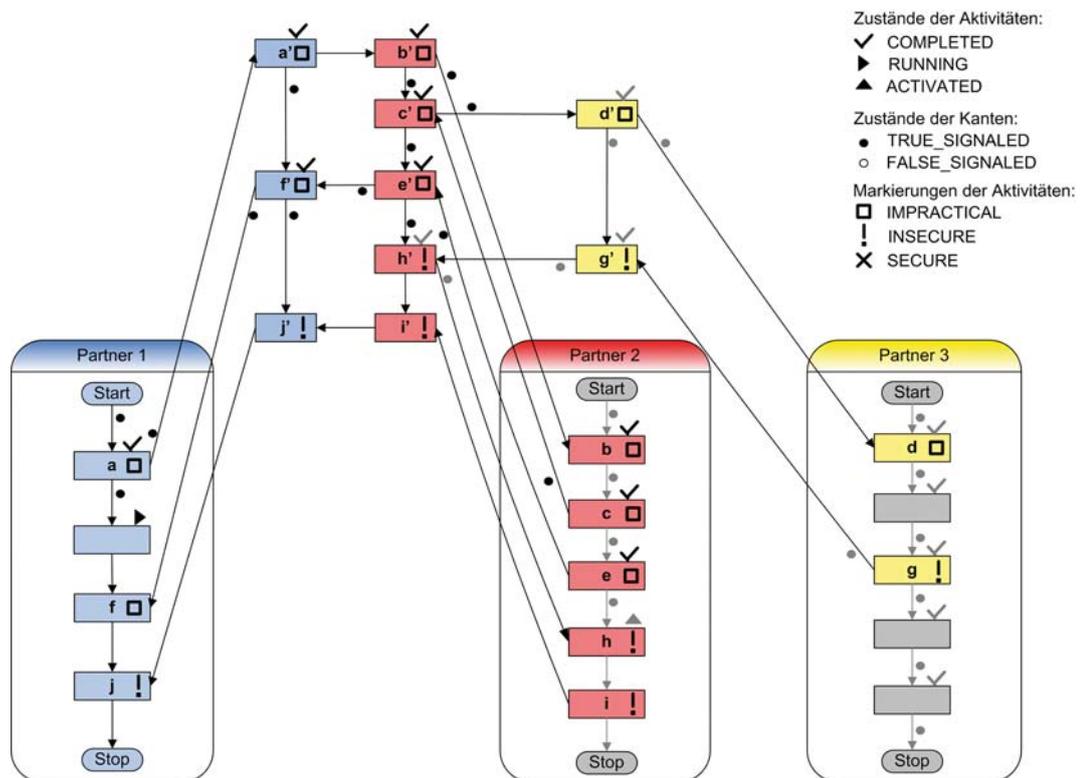


Abbildung 6.4: Zustandsbild mit Markierungen aus Sicht des blauen Partners (Ansatz 3)

Für eine Choreographie-Instanz nach Ansatz 3 stellt sich die Situation folgendermaßen dar. In Abbildung 6.4 wird ein Zustandsbild mit Markierungen aus Sicht des blauen Partners vorgestellt. Partner 1 setzt hier auch die Markierungen für Partner 3, obwohl beide als direkten Partner nur Partner 2 haben. An dieser Stelle soll nochmal betont werden, dass Partner 1 zwar den globalen öffentlichen Workflow mit den öffentlichen Aktivitäten von Partner 2 und Partner 3 kennt, aber nicht deren aktuellen Zustände. Der Entscheider (in Abbildung 6.4 ist das der blaue Partner), kann auch bei der verketteten Entscheidung auf die Markierungsregeln des

mittigen Entscheiders zurückgreifen, aber nur für die eigenen öffentlichen Aktivitäten und die entsprechenden öffentlichen Aktivitäten des direkten Partners (dies sind die Aktivitäten b', e' und i' des roten Partners in Abbildung 6.4). Es lassen sich zusätzlich auch Informationen über den entfernten Partner herleiten, z.B. weiß der blaue Partner in Abbildung 6.4 durch den Empfang der Nachricht von Aktivität e', dass Partner 2 die Nachricht von Aktivität c' bereits an den gelben Partner versendet haben muss. Falls die Markierungen für einen Partner über einen direkten Partner hinweg gesetzt werden müssen (z.B. für den gelben Partner in Abbildung 6.4), werden zusätzliche Markierungsregeln benötigt.

Die Markierungen werden dann in folgender Reihenfolge vorgenommen:

- Markierung der eigenen öffentlichen Aktivitäten nach den Regeln für die eigenen öffentlichen Aktivitäten des Entscheiders in Kapitel 6.2.
- Markierung der öffentlichen Aktivitäten des direkten Partners, die in bilateraler Beziehung mit dem Entscheider stehen, nach den Regeln für die öffentlichen Aktivitäten des Partners in Kapitel 6.2.
- Markierung der öffentlichen Aktivitäten des entfernten Partners nach den im nächsten Absatz vorgestellten Regeln.
- Markierung der kooperativen privaten Aktivitäten.

Die noch benötigten Markierungsregeln für die öffentlichen Aktivitäten des direkten Partners, welche mit einem entfernten Partner kooperieren, lauten:

- Markierung IMPRACTICAL, wenn die erste nachfolgende öffentliche Aktivität des direkten Partners, welcher mit dem Entscheider in bilateraler Beziehung steht, die Markierung IMPRACTICAL hat.
- Markierung INSECURE,
 - wenn die erste nachfolgende öffentliche Aktivität des direkten Partners, welche mit dem Entscheider in bilateraler Beziehung steht, die Markierung INSECURE hat, oder
 - wenn die erste nachfolgende öffentliche Aktivität des direkten Partners, welcher mit dem Entscheider in bilateraler Beziehung steht, die Markierung SECURE hat und die erste vorherige öffentliche Aktivität des direkten Partners, welcher mit dem Entscheider in bilateraler Beziehung steht, die Markierung INSECURE oder IMPRACTICAL hat, oder
 - wenn es keine nachfolgenden öffentlichen Aktivitäten des direkten Partners gibt, welcher mit dem Entscheider in bilateraler Beziehung steht.

- Markierung SECURE, wenn die erste vorherige öffentliche Aktivität des direkten Partners, welcher mit dem Entscheider in bilateraler Beziehung steht, die Markierung SECURE hat

Für die öffentlichen Aktivitäten des entfernten Partners, welcher mit einem direkten Partner des Entscheiders kooperiert, werden die gleichen Markierungen gesetzt, wie für die entsprechenden öffentlichen Aktivitäten des direkten Partners.

Die privaten kooperativen Aktivitäten des entfernten und des direkten Partners, sowie die eigenen kooperativen Aktivitäten, werden genauso wie die zugehörigen öffentlichen Aktivitäten markiert.

Es ist demnach auch bei einer verketteten Entscheidung möglich eventuell eine Vorentscheidung zu treffen. Tendenziell müssen jedoch in diesem Fall mehr Aktivitäten mit INSECURE markiert werden, als bei einem mittigen Entscheider. Es kann also in weniger Fällen eine Vorentscheidung mit dem Ergebnis getroffen werden, dass eine Änderung überhaupt nicht oder sicher durchführbar ist. Die hier gemachten Markierungsvorschläge sind für Choreographien nach Ansatz 3 anzuwenden, bei denen nur über einen Partner hinweg kommuniziert wird. Theoretisch könnte man auch noch Markierungen für länger verkettete Konstellationen definieren, wobei dann jedoch viele Aktivitäten mit INSECURE markiert werden müssen.

Bei einer Choreographie-Instanz nach Ansatz 4 ist eine verkettete Vorentscheidung nicht möglich. Bei diesem Ansatz gibt es keinen globalen öffentlichen Workflow, anhand dessen der Entscheider sich Markierungen über einen entfernten Partner machen könnte. Der blaue Partner in Abbildung 6.3 weiß z.B. nichts über die öffentlichen Aktivitäten zwischen Partner 2 und 3, obwohl alle zur gleichen Choreographie-Instanz gehören. Falls trotzdem anhand von Markierungen eine Vorentscheidung getroffen werden soll, müsste jeder Partner diese für seine direkten Partner treffen und das Ergebnis dann an den Entscheider zurückliefern. Da es hier aber zu einer Kommunikation zwischen den Partnern kommt, bringt dieser Ansatz nicht mehr viele Vorteile zum globalen Entscheidungsverfahren in Kapitel 7. Nur falls man eine Vorentscheidung treffen will, ohne die Workflows bei der Weiterarbeit zu stoppen (inwiefern das beim globalen Entscheidungsverfahren nötig ist, wird in Kapitel 7 diskutiert), macht es Sinn Markierungen nach dem lokalen Entscheidungsverfahren von mehreren Partnern durchführen zu lassen.

7 Globale Entscheidungsverfahren

Falls durch das lokale Entscheidungsverfahren bereits feststeht, dass eine Änderung nicht durchgeführt werden kann, müssen keine weiteren Verfahren mehr angewendet werden. Kommt das lokale Entscheidungsverfahren aber zum Ergebnis, dass nicht sicher ist, ob die Änderung durchführbar ist, muss als nächster Schritt mit den Partnern abgesprochen werden, ob deren privater Workflow mit der Änderung verträglich ist. Dieser Schritt der globalen Entscheidungsfindung muss auf jeden Fall auch stattfinden, falls es an dieser Stelle durch das lokale Entscheidungsverfahren bereits sehr wahrscheinlich ist, dass alle privaten WF-Instanzen migrieren können. Erstens wird dadurch jedem Partner eine Art "Vetorecht" eingeräumt, da aufgrund der Autonomie jedes einzelnen keiner gezwungen werden kann zu migrieren. Und zweitens wird diese Vorentscheidung aufgrund von abgeleiteten Informationen getroffen, weshalb die Möglichkeit besteht, dass die tatsächliche interne Struktur des privaten Workflows des Partners die Änderung überhaupt nicht erlaubt, oder die private Instanz in ihrer Ausführung bereits fortgeschritten ist. Die globale Entscheidungsfindung sollte bei der Evolution von Prozess-Choreographien also vor der Durchführung einer Migration stattfinden, wohingegen das lokale Entscheidungsverfahren nur eventuell eine Vorentscheidung ermöglicht.

Die Thematik, ob und wann die privaten Workflows für das globale Entscheidungsverfahren gestoppt werden müssen, wird in Kapitel 7.1 aufgegriffen. Bei der Analyse des lokalen Entscheidungsverfahrens in Kapitel 6 war die mittige Position des Entscheiders relevant. Diese Ausgangslage ist für die globalen Entscheidungsverfahren nicht weiter von Bedeutung, da es hier sowieso zu einer Befragung der Partner kommt. Die Aufteilung in Kapitel 7.2 Zentraler Änderungsmanager, Kapitel 7.3 Zentraler Koordinator und Kapitel 7.4 Verkettete Abstimmung ergibt sich aus dem Aspekt, wie die Entscheidungsfindung koordiniert werden kann.

7.1 Vermeidung von Inkonsistenzen und Deadlocks

Ein Partner, der beim globalen Entscheidungsverfahren der Migration zugestimmt hat, muss bei der darauf folgenden Durchführung der Migration auch tatsächlich migrieren können. Es darf also nicht passieren, dass sich die private WF-Instanz dann in einem Zustand befindet, bei dem es zu einem Fehler bei der Durchführung der Änderung kommen kann. Um dies zu Verhindern kann man die Instanz des privaten Workflows eines Partners stoppen, sobald eine Änderungsanfrage bei diesem Partner eintrifft. Der Partner prüft dann seine privaten WF-

Instanz auf Verträglichkeit und wartet darauf, ob er die Migration durchführen soll oder nicht. Erst nachdem die Migration dann durchgeführt wurde bzw. abgebrochen wird, kann der private Workflow weiterlaufen. Dies ist ein sehr restriktives Vorgehen, aber vor allem falls das lokale Entscheidungsverfahren keine Aussage treffen konnte, könnte man so sicher gehen, dass sich die Choreographie nach der Migration in einem korrekten Zustand befindet.

Aufgrund der relativ aufwendigen Entscheidungsfindung bei einer Choreographie und die eventuell weiten Übertragungswege zwischen den Partnern, führt dieses Vorgehen leider zu einem langen Blockieren der privaten Workflows. Ein etwas optimistischeres Vorgehen wäre, das Weiterlaufen der Workflows bis zu einem kritischen Bereich zu erlauben, konkreter, bis zu den von der Änderung betroffenen Aktivitäten. Falls in Abbildung 6.3 auf Seite 44 der rote Partner seine kooperative Aktivität p löschen möchte, kann er davon ausgehen, dass diese Änderung durchführbar ist. Trotzdem muss der blauen Partner in einem globalen Entscheidungsverfahren dieser Änderung noch zustimmen. Während der Entscheidungsfindung könnten aber beide privaten Workflows problemlos weiterlaufen, bis die Aktivitäten p und o den Zustand ACTIVATED erhalten. Erst dann müsste die weitere Ausführung unterbunden werden, um keine Inkonsistenzen durch das Löschen der Aktivität p und somit auch p' , o' und o zu erzeugen. In den folgenden Kapiteln wird vom Stoppen der privaten Workflows ausgegangen, um den Ablauf möglichst einfach zu halten. Stattdessen könnte der Partner aber auch, solange er auf eine Entscheidung wartet, seine Instanz bis zu den Aktivitäten weiterlaufen lassen, deren eingehende Kanten von der Änderung unberührt bleiben. Die Entscheidung, ob die private Instanz sofort gestoppt wird oder bis zu einem kritischen Bereich weiterlaufen darf, muss von den Partnern selbst getroffen werden. Denn bei den Partnern selbst liegt die Verantwortung, dass ihre Instanz im Fall der Migration mit der Änderung weiterhin verträglich ist.

Wenn noch einen Schritt weitergegangen wird, dann könnte ein Partner auch sofort migrieren. Aber nur unter der Voraussetzung, dass er im Fall eines Abbruchs (weil ein anderer Partner nicht migrieren kann), die Änderungen wieder rückgängig machen kann und seine Instanz auf den Zustand vor der Änderung zurücksetzen kann. Inwiefern sich ein privater Workflow zurücksetzen lässt, hängt allerdings vom eingesetzten WfMS ab. Auch wenn davon ausgegangen wird, dass die Rücksetzbarkeit der privaten WF-Instanzen einer Choreographie gegeben ist, könnten weitere Probleme entstehen, die eventuell zu Inkonsistenzen bei der Choreographie-Instanz führen. Falls z.B. der bereits migrierte Partner Nachrichten verschickt, die nur aufgrund der Änderung existieren und deshalb vom Empfänger nicht verarbeitet werden können. Dieses Vorgehen basiert im Gegensatz zu anderen Verfahren nicht auf einer globalen Entscheidungsfindung vor der Migration, sondern setzt auf Rücksetzbarkeit und wird in Kapitel 8.5 als optimistisches Migrationsverfahren nochmals aufgegriffen.

Um die Choreographie-Instanz möglichst wenig einzuschränken, sollten nur die private WF-Instanzen, die von der Änderung auch tatsächlich betroffen sind, für die Entscheidungsfindung

und die Migration blockiert werden. Ob ein Partner und somit dessen private Instanz miteinbezogen werden muss, kann bereits auf Schemaebene festgestellt werden. Wie in Kapitel 2.4 bereits angesprochen wurde, wird im Projekt DYCHOR [RWR06] hierfür der Schnitt zweier endlicher Automaten, welche jeweils die öffentlichen Aktivitäten der Partner beschreiben, auf Konsistenz geprüft. Bei Inkonsistenz ist der Partner ebenfalls von der Änderung betroffen und seine öffentlichen Aktivitäten, sowie sein privater Workflow müssen angepasst werden. Falls es nicht zu Inkonsistenzen kommt, muss der private Workflow des Partners in die Entscheidungsfindung und die eventuell darauf folgende Migration auch nicht mit einbezogen werden. Auf der Grundlage der in [RWR06] verwendeten Zustandsautomaten, lässt sich auch ermitteln, welche Information dem Partner zur Verfügung gestellt werden muss, damit dieser weiß in welcher Form seine öffentlichen Aktivitäten anzupassen sind. Für eine additive Änderung lässt sich z.B. die neu eingefügte Nachrichtensequenz errechnen. Die Vereinigung dieser Nachrichtensequenz mit den öffentlichen Aktivitäten des Partners ergibt einen Zustandsautomaten, der die nötigen Anpassungen für den Partner aufzeigt [RWR06].

7.2 Zentraler Änderungsmanager

Bei dieser Variante eines globalen Entscheidungsverfahrens ist ein Änderungsmanager (eine unabhängige “dritte“ Partei), für alle Partner der Choreographie zuständig. Jeder Partner, der eine Änderung durchführen will, muss diese Änderungsanfrage an den Änderungsmanager übergeben. Der Änderungsmanager koordiniert dann die Entscheidungsfindung für die gesamte Choreographie. Dieses Vorgehen hat den entscheidenden Vorteil, dass es nicht zu einer Kollision von Änderungen in der Choreographie kommen kann. Der Änderungsmanager hält für den Fall, dass er bereits eine Änderung bearbeitet, weitere Änderungswünsche zurück, bis die laufende Änderung abgeschlossen ist. Ansonsten würde die Gefahr bestehen, dass sich Änderungsanfragen gegenseitig blockieren und die Choreographie-Instanz nicht mehr weiterlaufen kann. Jeder Partner kann nur eine Änderungsanfrage entgegennehmen und stoppt seine private Instanz bis er die Aufforderung zum migrieren erhält. In dieser Zeitspanne, vom Eintreffen der Änderungsanfrage bis zur abgeschlossenen Migration, befindet der Partner sich in einem Änderungszustand. Eine klassische Verklemmungssituation würde dann entstehen, wenn zwei konkurrierende Änderungsanfragen die Partner blockieren und deshalb keine Anfrage abgeschlossen werden kann. Ein weiterer Vorteil eines Änderungsmanagers besteht darin, dass der Initiator einer Änderung entlastet wird. Er muss nicht die Fähigkeit besitzen die Entscheidungsfindung zu koordinieren. Inwiefern eine globale Entscheidung mit einem Änderungsmanager durchführbar ist, wird im Folgenden anhand der Ansätze 2 bis 4 aus Kapitel 4 besprochen.

7.2.1 Ablauf bei Ansatz 2

Aufgrund dessen, dass es bei Ansatz 2 keine öffentlichen Aktivitäten gibt und alle Aktivitäten der Choreographie bekannt sind, kann der Änderungsmanager die strukturellen Auswirkungen einer Änderung auf alle privaten WF-Instanzen überblicken. Trotzdem benötigt er, um Entscheiden zu können, ob eine Migration möglich ist, noch die Zustandsinformationen der privaten Workflows. Bei diesem Ansatz besteht ausnahmsweise die Möglichkeit, die aktuellen Zustände der Partner zu synchronisieren. Somit ergibt sich ein globales Zustandsbild über alle Aktivitäten der Choreographie. Ein globales Zustandsbild kann bei den Ansätzen 3 und 4 aufgrund der Autonomie der Partner nicht erstellt werden. Das lokale Korrektheitskriterium kann bei Ansatz 2 direkt auf die umfassende Choreographie-Instanz angewendet werden und die privaten WF-Instanzen der Partner müssen nicht einzeln getestet werden. Eine einfache Möglichkeit, den Änderungsmanager über die aktuellen Zustände zu informieren, ist die Ablaufhistorie zu übermitteln. Anhand dieser Information kann der Änderungsmanager ableiten, wie weit die private WF-Instanz in der Ausführung ist, indem er z.B. die vorhandenen Einträge "nachspielt". Dieses Vorgehen ist vergleichbar mit der Synchronisierung bei einem verteilten Workflow [BRD01a]. Es wird also davon ausgegangen, dass die WfMS der Partner alle eine Ablaufhistorie haben und der Änderungsmanager mit diesen Aufzeichnungen arbeiten kann. Falls nicht mit Ablaufhistorien gearbeitet werden kann, besteht auch die Möglichkeit die privaten WF-Instanzen, wie bei den anderen Ansätzen, einzeln auf Verträglichkeit zu prüfen.

Ein globales Entscheidungsverfahren mit Änderungsmanager läuft bei Ansatz 2 folgendermaßen ab:

- Der Initiator einer Änderung teilt dem Änderungsmanager seinen Änderungswunsch mit.
- Der Änderungsmanager erschließt sich alle von der Änderung strukturell betroffenen Partner.
- Der Änderungsmanager fordert bei den betroffenen Partnern deren Ablaufhistorie an.
- Mit dem Eintreffen der Anforderung stoppen die Partner ihre privaten Workflows und geben ihre Ablaufhistorie an den Änderungsmanager.
- Der Änderungsmanager überprüft anhand des globalen Zustandsbildes, ob die Choreographie-Instanz mit der Änderung verträglich ist.
- Falls nicht migriert wird, teilt der Änderungsmanager dies den betroffenen Partnern mit und deren private WF-Instanzen können weiter laufen. Ansonsten leitet der Änderungsmanager ein Migrationsverfahren ein.

7.2.2 Ablauf bei Ansatz 3

Der Änderungsmanager kennt bei Ansatz 3 alle in die Choreographie-Instanz involvierten Partner und den globalen öffentlichen Prozess. Die privaten Workflows sind ihm aber nicht bekannt. Der Initiator einer Änderung muss deshalb ableiten, inwiefern seine öffentlichen Aktivitäten aufgrund der geplanten Modifikation des privaten Workflows anzupassen sind. Der Änderungsmanager kann danach, anhand dieser Information, die strukturell von der Änderung betroffenen Partner ausmachen und auch die nötigen Anpassungen für deren öffentliche Aktivitäten bestimmen. Ein von der Änderung betroffener Partner kann aufgrund der nötigen Anpassungen für seine öffentlichen Aktivitäten, die entsprechende Modifikation seiner privaten Aktivitäten ableiten. Jeder von der Änderung betroffene Partner muss dann für seine private WF-Instanz nach dem lokalen Korrektheitskriterium bestimmen, ob seine Instanz mit der Änderung verträglich ist.

Der Ablauf bei einem globalen Entscheidungsverfahren mit Änderungsmanager ist folgender:

- Der Initiator einer Änderung bestimmt die Anpassung seiner öffentlichen Aktivitäten und teilt diese dem Änderungsmanager mit.
- Der Änderungsmanager bestimmt die Modifikation des globalen öffentlichen Prozesses und die von der Änderung betroffenen Partner.
- Der Änderungsmanager sendet eine Änderungsanfrage an die betroffenen Partner (auch an den Initiator) mit den nötigen Änderungen für deren öffentliche Aktivitäten.
- Die betroffenen Partner stoppen ihre privaten Instanzen und überprüfen, ob die Anpassung ihres privaten Workflows strukturell möglich ist und ob die private Instanz mit der strukturellen Änderung verträglich ist.
- Die betroffenen Partner teilen dem Änderungsmanager ihr Ergebnis mit und warten auf die Entscheidung. Falls ein Partner aber zum Ergebnis kommt, dass seine Instanz mit der Änderung nicht verträglich ist, kann er den Abbruch der Änderungsanfrage vorwegnehmen und seine private Instanz weiterlaufen lassen.
- Der Änderungsmanager wertet alle Antworten aus und teilt den betroffenen Partnern mit, falls nicht migriert wird. Ansonsten leitet er ein Migrationsverfahren ein.

7.2.3 Ablauf bei Ansatz 4

Die Schwierigkeit bei Ansatz 4 besteht darin, dass es keinen globalen öffentlichen Prozess gibt. Für jede bilaterale Beziehung zwischen zwei Partnern existiert jeweils ein öffentlicher Prozess. Diese Konstellation bietet die größte Autonomie für die Partner. Es gibt bei bestehenden Modellen manchmal bereits eine Überwachungseinheit (z.B. bei [ZLY05] und [CTD04]) bei der

sich alle Partner anmelden, diese könnte dann auch noch die Funktion des Änderungsmanagers übernehmen. Davon kann bei Ansatz 4 aber nicht allgemeingültig ausgegangen werden. Der Änderungsmanager kennt grundsätzlich vor einer Änderung weder alle in einer Choreographie-Instanz involvierten Partner, noch deren öffentliche Workflows. Diese Informationen müssen dann bei einem Änderungswunsch erst ermittelt werden, was zu einem gewissen Kommunikationsaufwand führt. Der Änderungsmanager ermittelt dann, wie bei Ansatz 3, die nötigen strukturellen Änderungen für die öffentlichen Aktivitäten der Partner.

Folgender Ablauf ergibt sich bei Ansatz 4:

- Der Initiator einer Änderung bestimmt die Anpassung seiner öffentlichen Aktivitäten und damit auch die direkten Partner, die von der Änderung betroffen sind. Er teilt dem Änderungsmanager seine öffentlichen Aktivitäten, die nötigen Anpassungen und die betroffenen direkten Partner mit.
- Der Änderungsmanager fordert von den genannten Partnern deren öffentliche Aktivitäten an.
- Anhand der nun verfügbaren öffentlichen Prozesse bestimmt der Änderungsmanager welche Anpassungen die Partner an ihren öffentlichen Aktivitäten vornehmen müssen und teilt dies den Partnern mit.
- Die Partner stoppen ihre privaten Instanzen und überprüfen, ob die entsprechende Änderung ihres privaten Workflows strukturell möglich ist und ob die private Instanz mit der strukturellen Änderung verträglich ist. Außerdem überprüfen die Partner, ob in der Folge der Modifikation ihres privaten Workflows noch weitere Partner, mit denen sie auch in bilateraler Beziehung stehen, betroffen sind. Die entsprechenden Informationen geben sie dann an den Änderungsmanager.
- Der Änderungsmanager verfährt bei den Partnern der Partner wie oben und wiederholt die letzten drei Punkte bis keine neuen involvierten Partner mehr hinzukommen.
- Dann wertet der Änderungsmanager alle Antworten aus und teilt den betroffenen Partnern mit, falls nicht migriert wird. Ansonsten leitet er ein Modifikationsverfahren ein.

7.3 Zentraler Koordinator

Bei dieser Variante übernimmt der Initiator einer Änderung auch die Entscheidungsfindung. Der Koordinator wechselt mit dem Initiator einer Änderung und damit ist nicht, wie beim Änderungsmanager, immer die gleiche Einheit für alle Änderungswünsche zuständig. Deshalb kann es hier eventuell zu Kollisionen von Änderungswünschen kommen, die von verschiedenen Partnern initiiert werden. Folglich werden Mechanismen benötigt, um Verklemmungen zu

vermeiden bzw. aufzulösen. Diese Thematik wird in Kapitel 9 mit entsprechenden Lösungsansätzen nochmals aufgegriffen.

Der Ablauf der Entscheidungsfindung mit zentralem Koordinator entspricht bei Ansatz 2 dem Verlauf bei Verwendung eines Änderungsmanagers, welcher in Kapitel 7.2.1 vorgestellt wurde. Allerdings übernimmt nun der zentrale Koordinator die Rolle des Änderungsmanagers. Der Koordinator kann aber nur dann eine Entscheidungsfindung initiieren, falls nicht bereits ein anderer Änderungswunsch (der eventuell von einem anderen Partner koordiniert wird) geprüft wird.

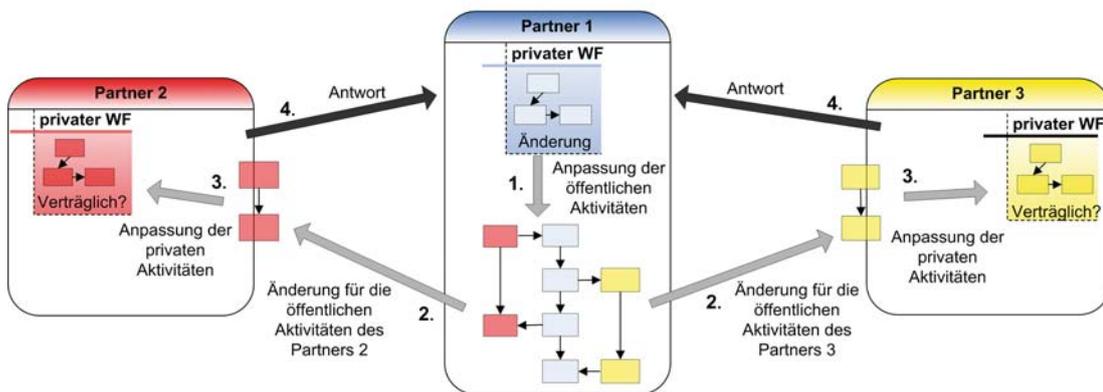


Abbildung 7.1: Entscheidungsfindung mit zentralem Koordinator (Ansatz 3)

Der Ablauf für Ansatz 3 wird anhand von Abbildung 7.1 exemplarisch dargestellt.

1. Partner 1 möchte eine Änderung an seinem privaten Workflow vornehmen. Seine private WF-Instanz wird gestoppt und ist verträglich mit der Änderung.
2. Partner 1 bestimmt die von der Änderung betroffenen Partner und die Modifikation des globalen öffentlichen Prozesses.
3. Partner 1 sendet eine Änderungsanfrage an Partner 2 und Partner 3 mit den nötigen Änderungen für deren öffentliche Aktivitäten.
4. Partner 2 und Partner 3 leiten die Modifikationen für ihren privaten Workflow ab, stoppen ihre private WF-Instanz und führen eine Verträglichkeitsprüfung durch (falls eine Instanz unverträglich ist, kann sie an dieser Stelle gleich weiterlaufen).
5. Partner 2 und Partner 3 informieren Partner 1, ob sie migrieren können.
6. Partner 1 leitet ein Migrationsverfahren ein, falls er nur positive Antworten erhalten hat. Ansonsten meldet er beiden Partnern den Abbruch der Änderungsanfrage.

Es handelt sich um einen zentralen Koordinator, weil der für die Entscheidungsfindung zuständige Koordinator alle betroffenen Partner über die nötigen Änderungen informiert.

Für Ansatz 4 ist ein zentraler Koordinator nicht zu realisieren. Der Initiator einer Änderung würde dann alle öffentlichen Prozesse und Partner kennen und das würde dem Aufbau dieses Ansatzes widersprechen.

7.4 Verkettete Abstimmung

Bei der verketteten Abstimmung steht keine zentrale Einheit zur Verfügung, welche die Entscheidungsfindung übernimmt. Der Änderungswunsch wird vielmehr von einem zum anderen Partner durchgereicht. Der wichtigste Unterschied zu den zentralen Varianten besteht darin, dass bei der verketteten Abstimmung jeder Teilnehmer die nötigen Änderungen für die öffentlichen Aktivitäten seiner direkten Partner selbst bestimmen muss. Außerdem übernimmt er die Entscheidungsfindung für seine direkten Partner und die weiteren Partner in diesem Verteilungszweig. Ein Vorteil dieser Vorgehensweise gegenüber den zentralen Verfahren ist, dass logische Abhängigkeiten die zu kaskadierenden Änderungen führen (siehe Kapitel 4.4.1), bei den Partnern erkannt werden und die betroffenen Partner miteinbezogen werden können. Bei den zentralen Entscheidungsverfahren führen solche Abhängigkeiten im privaten Workflow eines Partners dazu, dass dieser nicht migrieren kann, oder es muss für solche Fälle eine Ausnahmebehandlung vorgesehen werden.

Der Ablauf einer verketteten Abstimmung wird für Ansatz 4 anhand der Abbildung 7.2 exemplarisch dargestellt. Es ergibt sich eine Baumstruktur und die Ablaufschritte in den verschiedenen Zweigen können parallel ablaufen, wodurch es in den Zweigen zu einer unabhängigen Entscheidungsfindung kommt, welche beim Initiator wieder zusammenläuft.

1. Partner 1 möchte eine Änderung an seinem privaten Workflow vornehmen. Seine private WF-Instanz wird gestoppt und ist verträglich mit der Änderung. Betroffen sind auch die öffentlichen Prozesse mit Partner 2 und Partner 3. Partner 1 leitet sich die strukturellen Anpassungen für seine öffentlichen Aktivitäten ab.
2. Anhand seiner geänderten öffentlichen Aktivitäten kann Partner 1 die strukturell nötigen Änderungen für die öffentlichen Aktivitäten von Partner 2 und Partner 3 bestimmen (vgl. [RWR06] für Änderungen auf Schemaebene) und Partner 1 sendet seine Änderungsanfrage.
3. Partner 2 und Partner 3 leiten die Modifikationen für ihren privaten Workflow ab, stoppen ihre private WF-Instanz und führen eine Verträglichkeitsprüfung durch. Falls eine Instanz mit der Änderung unverträglich ist, kann sie weiterlaufen, die Änderungsanfrage wird abgebrochen und das Ergebnis Partner 1 mitgeteilt.

4. Partner 2 und Partner 3 untersuchen, ob es durch die Anpassung ihres privaten Workflows zu weiteren Modifikationen bei einem ihrer bilateralen öffentlichen Prozesse kommt.
5. Partner 2 sendet eine Änderungsanfrage an Partner 4. Bei Partner 3 sind zwei direkte Partner betroffen und Partner 3 sendet eine Änderungsanfrage an Partner 5 und Partner 6.
6. Partner 4, Partner 5 und Partner 6 leiten die Modifikationen für ihren privaten Workflow ab, stoppen ihre private WF-Instanz und führen eine Verträglichkeitsprüfung durch. Falls eine Instanz mit der Änderung unverträglich ist, kann sie weiterlaufen, die rekursive Änderungsanfrage wird abgebrochen.
7. Falls die private Instanz von Partner 4 verträglich mit der Änderung ist, gibt er Partner 2 eine positive Antwort. Ebenso geben die Partner 5 und 6 an Partner 3 eine positive Antwort bei Verträglichkeit.
8. Partner 2 und Partner 3 werten alle Antworten aus und falls alle positiv sind, geben sie an Partner 1 ebenfalls eine positive Antwort.
9. Partner 1 leitet ein Migrationsverfahren ein, falls er nur positive Antworten erhalten hat. Ansonsten sendet er an beide direkten Partner den Abbruch der Änderungsanfrage (diese wird dann weiter durchgereicht), damit auch die Zweige die positiv abgestimmt haben ihre privaten WF-Instanzen weiterlaufen lassen.

Ein Problem bei der verketteten Abstimmung besteht darin, dass der Initiator einer Änderung nicht weiß, wie lange er auf die Antwort einer Änderungsanfrage warten muss. Falls ein Partner nicht antwortet, wird es schwierig sein, herauszufinden an welcher Stelle des Zweiges das Problem liegt. Solange keine Entscheidung getroffen ist, können aber die privaten WF-Instanzen nicht weiterlaufen. Vor allem bei der verketteten Abstimmung bietet es sich deshalb an, die privaten Instanzen bis zu einem kritischen Bereich, wie in Kapitel 7.1 vorgestellt, weiterlaufen zu lassen bis eine Entscheidung getroffen wurde. Damit die privaten WF-Instanzen nicht für immer blockiert bleiben, falls der Initiator durch einen Fehler aus einem Zweig keine Antwort zurück erhält, sollte ein Timeout gesetzt werden. Nach Ablauf des Timeouts, bricht der Initiator die Entscheidungsfindung ab.

Aufgrund der schwierigeren Entscheidungsfindung stellt sich dieses Verfahren für Ansatz 3 als weniger geeignet heraus. Außerdem wird dann auch nicht der Vorteil des globalen öffentlichen Workflows ausgenutzt, alle Änderungen der öffentlichen Aktivitäten zu überblicken und jeden betroffenen Partner direkt anzusprechen. Bei Ansatz 4 dagegen, muss bei einem zentralen Verfahren die Information über die öffentlichen Prozesse dem Änderungsmanager erst zugetragen werden, weshalb sich hinsichtlich der Autonomie hier die verkettete Abstimmung als geeigneter herausstellt. Eine Ausnahme stellt ein Modell nach Ansatz 4 dar, bei dem eine Überwachungs-Einheit existiert, welche bereits über die öffentlichen Prozesse informiert ist.

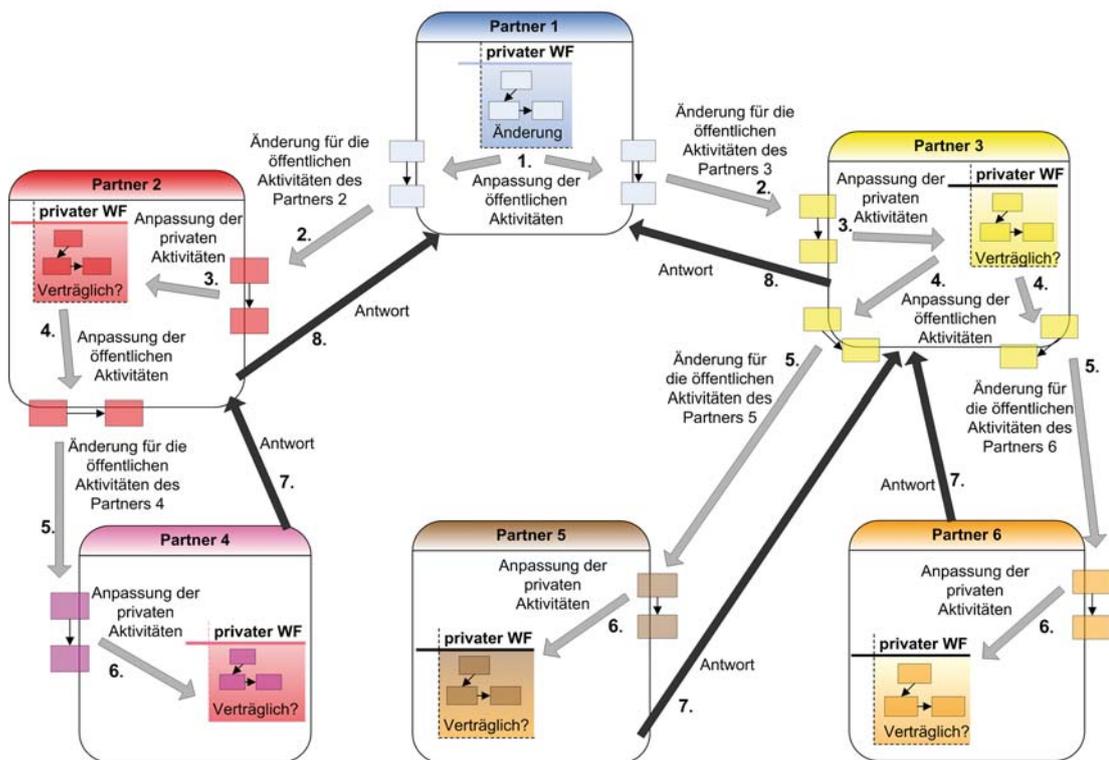


Abbildung 7.2: Verkettete Abstimmung (Ansatz 4)

Auch bei dieser Variante kann es zu einer Kollision von Änderungswünschen kommen. Wie dem entgegengewirkt werden kann, wird in Kapitel 9 aufgegriffen. Theoretisch kann es aber auch passieren, dass sich die Partner bei der Entscheidungsfindung für eine einzelne Änderungsanfrage gegenseitig blockieren. Dieser Fall kann aber nur eintreten, wenn es zu einer Ringkonstellation bei den bilateralen Kooperationsbeziehungen kommt. In Abbildung 7.2 wäre das z.B. der Fall wenn Partner 2 und Partner 5 in derselben Choreographie ebenfalls eine Kooperation eingegangen wären. Die von Partner 1 eingeleitete Änderungsanfrage könnte dann Partner 5 sowohl durch Partner 2 als auch durch Partner 3 erreichen. Demzufolge sollten solche Ringkonstellation in Prozess-Choreographien vermieden werden.

8 Migrationsverfahren

Die Migrationsverfahren beschreiben wie die Änderung durchgeführt wird. Die Einteilung bei den Migrationsverfahren ist prinzipiell die gleiche wie bei den globalen Entscheidungsverfahren, sie können aber durchaus unterschiedlich kombiniert werden. Zusätzlich wird noch ein optimistisches Verfahren vorgestellt, bei dem im Gegensatz zu den anderen Migrationsverfahren auf die vorangestellte globale Entscheidungsfindung verzichtet wird.

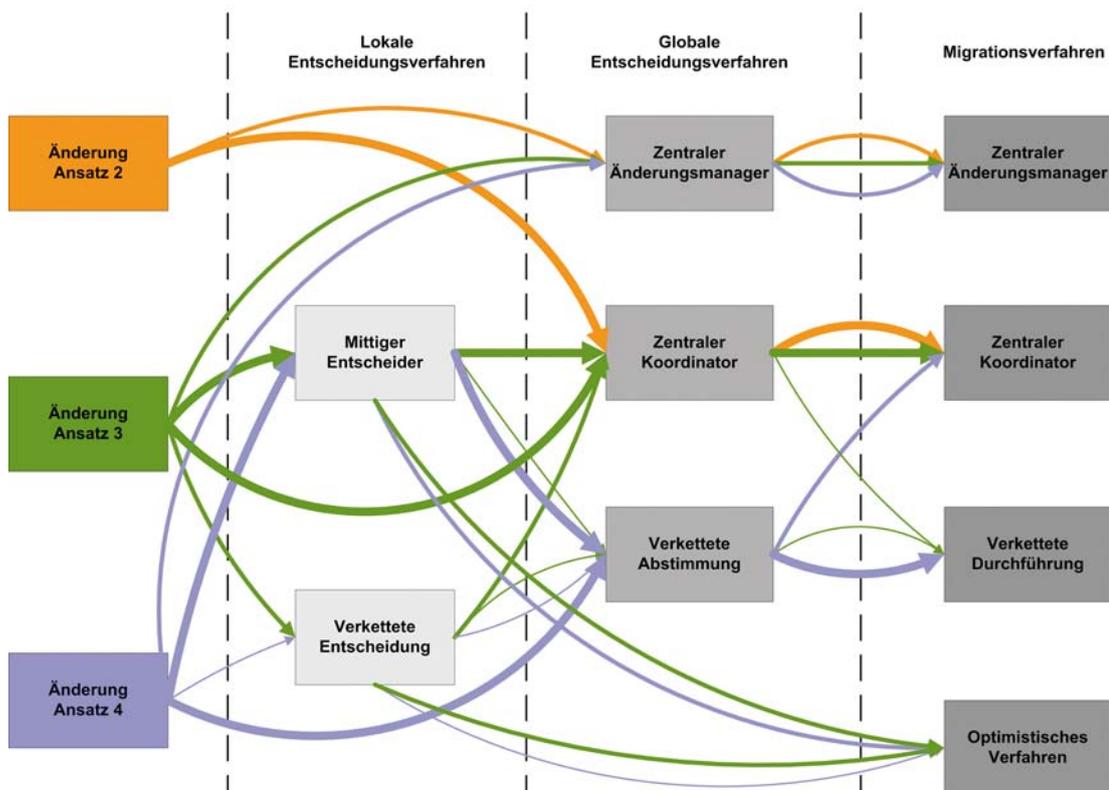


Abbildung 8.1: Mögliche Kombinationen der Verfahren

In Abbildung 8.1 sind die möglichen Kombinationen der Verfahren dargestellt. Nicht jedes Verfahren kann aufgrund der unterschiedlichen Voraussetzungen für alle Ansätze realisiert werden. Die drei verschiedenen Stärken der Pfeile in Abbildung 8.1 geben Auskunft über die Eignung des Verfahrens für den jeweiligen Ansatz. Je stärker der Pfeil, desto mehr Vorteile bringt das

Verfahren für den Ansatz und desto größer ist die Wahrscheinlichkeit, dass die benötigten Bedingungen erfüllt werden. Wann welches lokale und globale Entscheidungsverfahren für einen Ansatz in Frage kommt, wurde bereits in Kapitel 6 und Kapitel 7 diskutiert. Dieses Kapitel setzt sich mit den verschiedenen Verfahren für die Koordination der Migration anhand der unterschiedlichen Ansätze auseinander.

Während die Entscheidungsverfahren die Konsistenz der Choreographie-Instanz bei einer Änderung sicherstellen, sorgen die Migrationsverfahren für die Atomarität einer Änderung. Im Kontext einer Choreographie-Instanz wird unter Atomarität verstanden, dass eine Änderung als eine logische Einheit gesehen wird und entweder alle von der Änderung betroffenen privaten Instanzen einer Choreographie-Instanz migrieren oder keine.

8.1 Anpassung der Zustände

Bei der Migration werden die vorher mit der WF-Instanz auf Verträglichkeit überprüften Änderungen durchgeführt. Es wurde durch die Entscheidungsverfahren bereits sichergestellt, dass der Zustand der WF-Instanz die Änderung zulässt. Trotzdem müssen eventuell noch Zustandsanpassungen vorgenommen werden. Es kann z.B. beim Einfügen einer Aktivität erforderlich werden, dass eine Aktivität, welche sich bisher im Zustand ACTIVATED befindet, wieder auf NOT_ACTIVATED zurückgesetzt werden muss.

In Abbildung 8.2 wird ein solcher Fall anhand eines Beispiels dargestellt. Die Aktivität *c* hat vor der Migration den Zustand ACTIVATED. Weil aber vor der Aktivität *c* eine neue Aktivität *x* eingefügt wird, muss *c* der Zustand NOT_ACTIVATED zugewiesen werden. Der Zustand der Aktivität *x* wird dagegen auf ACTIVATED gesetzt, da die eingehende Kontrollkante bereits den Zustand TRUE_SINGALED hat. Ansonsten würde die neu eingefügte Aktivität Zustand NOT_ACTIVATED bekommen. Prinzipiell muss beim Einfügen einer Aktivität der direkte Nachfolger und die Kontrollkanten neu bewertet werden. Beim Hinzufügen einer Kontrollkante muss sowohl der Zustand für die Kante als auch für die Zielaktivität neu ermittelt werden. Verfahren, mit denen sich Knoten- und Kantenmarkierungen privater Instanzen automatisch anpassen lassen, werden in [RRD02] vorgestellt.

Die öffentlichen Aktivitäten können nur die Zustände NOT_ACTIVATED oder COMPLETED haben. Da eine Anpassung der Zustände nur in Verbindung mit Aktivitäten relevant ist, welche auch den Zustand ACTIVATED einnehmen können, müssen die öffentlichen Aktivitäten bei Änderungsoperationen nicht neu bewertet werden. Eine öffentliche Aktivität kann nur gelöscht werden, wenn sie den Zustand NOT_ACTIVATED hat. Alle neu eingefügten öffentlichen Aktivitäten werden mit NOT_ACTIVATED initialisiert. Lediglich beim Hinzufügen einer Kontrollkante, deren Start- und Zielaktivität eine öffentliche Aktivität ist, muss der Zustand dieser

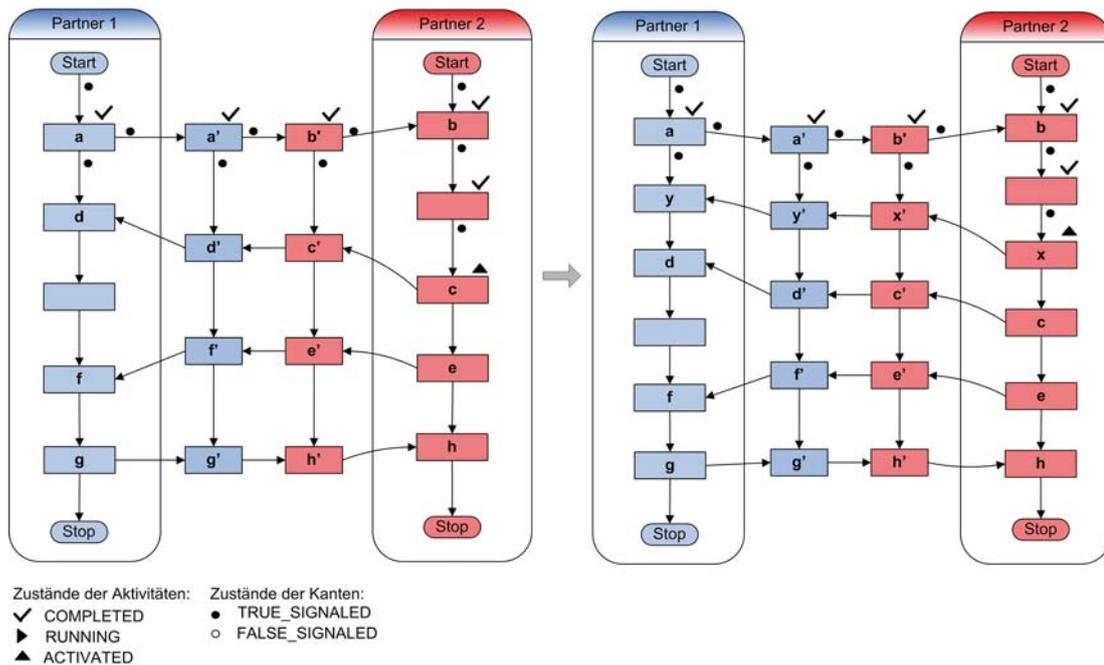


Abbildung 8.2: Migration mit Zustandsanpassung

Kante neu ermittelt werden. Falls die Startaktivität dieser neuen Kontrollkante den Zustand COMPLETED hat, wird sie auf TRUE_SIGNED gesetzt (wie zum Beispiel die Kontrollkante zwischen Aktivitäten b' und x' in Abbildung 8.2).

8.2 Zentraler Änderungsmanager

Die Migration von einem Änderungsmanager koordinieren zu lassen, bietet sich für alle Ansätze an, falls ein solcher existiert und er auch schon die Entscheidungsfindung übernommen hat. Durch einen Änderungsmanager wird, wie schon bei der Entscheidungsfindung, der Initiator einer Änderung entlastet, weil er nicht deren Durchführung übernehmen muss. Außerdem ist sichergestellt, dass nur eine Änderung zur gleichen Zeit durchgeführt wird. Nachdem alle von der Änderung betroffenen Partner dem Änderungsmanager die Verträglichkeit ihrer privaten WF-Instanz mitgeteilt haben, leitet der Änderungsmanager das Migrationsverfahren ein.

Das Migrationsverfahren läuft folgendermaßen ab:

- Der Änderungsmanager sendet an alle von der Änderung betroffenen Partner die Aufforderung zu migrieren.

- Die Partner führen die strukturelle Änderung bei ihrer privaten WF-Instanz durch und passen gegebenenfalls die Zustände an.
- Die Partner leiten ihre öffentlichen Aktivitäten von ihren privaten kooperativen Aktivitäten erneut ab und setzen den Zustand auf COMPLETED für bereits gesendete bzw. empfangene Nachrichten (dieser Schritt entfällt bei Ansatz 2).
- Daraufhin können die privaten WF-Instanzen weiter laufen und die Migration ist abgeschlossen.

Nach einer Migration kommt es bei Ansatz 2 zu einer interessanten Situation, was die Struktur der Choreographie-Instanz aus Sicht der Partner betrifft. Nur der Änderungsmanager hat das aktuelle Schema zu allen Partner Workflows und damit auch die aktuelle Struktur der Choreographie. Das kommt daher, dass nur die von der Änderung betroffenen Partner auch über die Änderung informiert werden. Man könnte auch ein Update an alle Partner schicken, damit alle Partner über die aktuelle Struktur der privaten Workflows informiert sind. Dies ist aber bei der Verwendung eines Änderungsmanagers nicht zwingend und würde zu mehr Kommunikation führen. Solange jeder Partner das aktuelle Schema für seinen privaten Workflow hat, gibt es auch für spätere Änderungswünsche keine Probleme, da diese sowieso vom Änderungsmanager bearbeitet werden und der ist strukturell auf dem aktuellen Stand aller in die Choreographie involvierter privater Instanzen.

Zu einer ähnlichen Konstellation, wie bei Ansatz 2, kann es bei Verwendung eines Änderungsmanagers auch bei Ansatz 3 kommen. Zwar kennen die Partner bei diesem Ansatz den privaten Workflow der anderen nicht, aber es wird davon ausgegangen, dass allen Partnern die Struktur des globalen öffentlichen Prozesses bekannt ist. Nach einer Änderung haben aber die Partner, welche nicht von der Änderung betroffen waren, eine ältere Version des globalen öffentlichen Prozesses. Dies führt allerdings bei der Verwendung eines Änderungsmanagers zu keinen Problemen, solange die eigenen öffentlichen Aktivitäten jedes Partners mit den entsprechenden öffentlichen Aktivitäten im globalen öffentlichen Prozess des Änderungsmanagers übereinstimmen. In Abbildung 8.3 wurden durch eine Änderung die Aktivitäten x' und y' dem globalen öffentlichen Prozess hinzugefügt. Von der Änderung waren nur Partner 2 und Partner 3 betroffen, weshalb auch nur diese Partner (und der Änderungsmanager) die Version 2 des globalen öffentlichen Prozesses zur Verfügung haben.

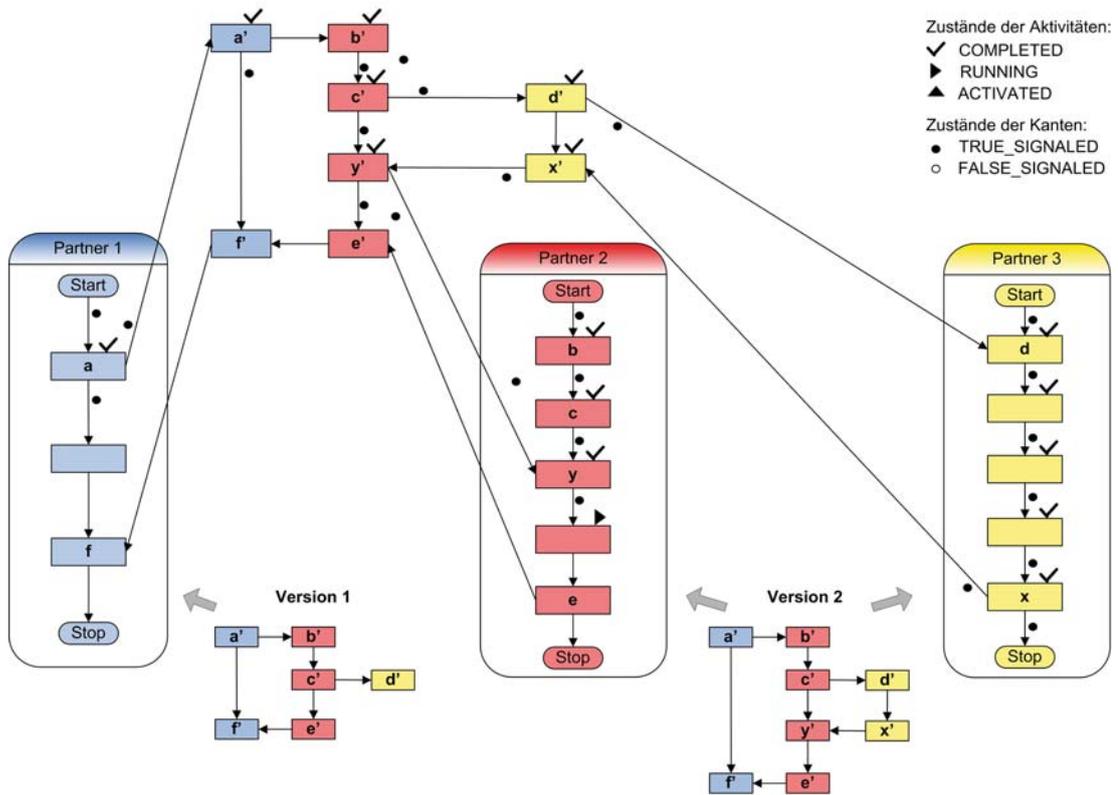


Abbildung 8.3: Verschiedene Versionen nach der Migration

8.3 Zentraler Koordinator

Bei der Migration hat der zentrale Koordinator dieselben Eigenschaften wie beim globalen Entscheidungsverfahren, d.h. der Initiator einer Änderung übernimmt auch deren Durchführung und informiert alle anderen Partner. Der Ablauf des Migrationsverfahrens ist grundsätzlich derselbe wie beim zentralen Änderungsmanager. Auf die Besonderheiten der Ansätze bei der Verwendung eines zentralen Koordinators wird im Folgenden eingegangen.

Bei Ansatz 2 kann es bei der Verwendung eines zentralen Koordinators wie beim zentralen Änderungsmanager dazu kommen, dass ein Teilnehmer nicht die aktuelle Struktur eines Partner-Workflows hat. Beim Migrationsverfahren mit zentralem Koordinator, dessen Rolle ja der initiierende Partner einer Änderung einnimmt, ist dieser Sachverhalt aber nicht tolerierbar. Der Grund dafür ist, dass der Koordinator einer neuen Änderung auf dem aktuellen Stand über alle privaten Workflows sein muss, um die Auswirkungen der Änderung abzusehen und die Durchführung zu koordinieren. Eine Möglichkeit die Strukturinformationen zu aktualisieren besteht darin, dass bei einer neuen Änderung der Koordinator von allen Partnern deren Änderungshistorie anfordert, um dann die aktuelle Struktur für alle privaten Workflows ableiten zu können. Eine Änderungshistorie wird z.B. bei ADEPT geführt, darin werden alle Änderungen, die für ein WF-Schema gemacht wurden protokolliert. Dieses aktive Anfordern der Information durch den neuen Koordinator eignet sich vor allem dann, wenn Änderungen der Choreographie selten sind. Eine andere Möglichkeit zur Aktualisierung besteht darin, den nicht von der Änderung betroffenen Partnern, die strukturellen Anpassungen für die betroffenen privaten Workflows als weiteren Schritt im Migrationsverfahren mitzuteilen.

Für Ansatz 3 muss bei der Verwendung eines zentralen Koordinators sichergestellt werden, dass alle Partner die aktuelle Struktur des globalen öffentlichen Prozesses kennen. Beim zentralen Änderungsmanager gab es dafür zwar keine Notwendigkeit, aber bei diesem Migrationsverfahren könnte der Koordinator einer neuen Änderung aufgrund einer veralteten Struktur für den globalen öffentlichen Prozess eventuell falsche Schlussfolgerungen ziehen. Als erster Schritt beim Migrationsverfahren sendet der Koordinator bei Ansatz 3, vor der Aufforderung zu migrieren, an alle Partner noch den neuen globalen öffentlichen Prozess. Dabei werden auch die von der Änderung betroffenen Partner miteinbezogen, da diese im Entscheidungsverfahren nur über die für ihre öffentlichen Aktivitäten relevanten Modifikationen informiert wurden.

Für Ansatz 4 hat sich der zentrale Koordinator in Kapitel 7.3 für die Entscheidungsfindung als ungeeignet herausgestellt. Trotzdem besteht die Möglichkeit nach einer verketteten Abstimmung ein Migrationsverfahren mit zentralem Koordinator durchzuführen. Der Koordinator spricht bei diesem Verfahren alle Partner direkt an, deshalb muss er über die von der Änderung betroffenen Partner informiert werden. Dafür muss bei der Antwort eines Partners auf die Änderungsanfrage, die für jeden Zweig über die Partner zurückgegeben wird (wie in Abbildung

7.2 auf Seite 57 dargestellt), jeder Partner seine URI¹ in eine Queue eintragen. Der Koordinator vereinigt alle zurückgegebenen Queues und kann anhand der URI alle von der Änderung betroffenen Partner direkt ansprechen. Der Koordinator erhält dann zwar auch die Information, welche entfernten Partner an der Choreographie beteiligt sind, aber nur wenn diese entfernten Partner auch wirklich von der Änderung betroffen sind. Außerdem hat der Koordinator keinerlei Informationen über den öffentlichen Prozess des entfernten Partners, weil dieser ja bereits über die nötigen strukturellen Modifikationen durch seinen direkten Partner während der Entscheidungsfindung informiert wurde. Der Koordinator sendet also lediglich die Aufforderung zur Migration an die betroffenen Partner.

8.4 Verkettete Durchführung

Bei der verketteten Durchführung wird die Aufforderung zur Migration, auf dem gleichen Weg, wie bei der verketteten Abstimmung, an die betroffenen Partner durchgereicht. Vor allem bei dieser Variante kommt es zu einer verzögerten Durchführung der Migration bei den verschiedenen Partnern. Theoretisch könnte es dann zu einer wie in Abbildung 8.4 dargestellten Situation kommen. Partner 2 hat die Migration bereits durchgeführt und seine private WF-Instanz weiterlaufen lassen. Die Aktivität x ist abgeschlossen und an Partner 1 wurde bereits die Nachricht an y' durch x' gesendet. Aber Partner 1 ist noch nicht migriert und die Aktivitäten y' und y wurden noch nicht eingefügt. Die private WF-Instanz von Partner 1 befindet sich zu dieser Zeit aber bereits in einem Änderungszustand, d.h. er hat der Migration bereits zugestimmt, sie aber noch nicht ausgeführt. Die Nachrichten, die empfangen werden, während sich eine private WF-Instanz im Änderungszustand befindet, müssen zwischengepuffert werden. Diese werden dann ausgeliefert, wenn die private Instanz wieder anläuft. Selbstverständlich würde diese Problematik nicht auftreten, wenn alle Partner exakt zur gleichen Zeit migrieren, aber das ist kaum möglich. Selbst wenn ein Zeitpunkt angegeben wird, an dem alle migrieren sollen, weichen die lokalen Zeiten in einem verteilten System voneinander ab. Außerdem müsste dann unter anderem eine zusätzliche Zeitspanne mit einberechnet werden, damit die Aufforderung zur Migration auch bei allen angekommen ist. Dadurch würde es nur zu weiteren Verzögerungen kommen.

Für Ansatz 3 bietet sich die verkettete Durchführung aufgrund der größeren Verzögerung bei der Migration der einzelnen Partner weniger an. Außerdem sind sowieso alle Partner bekannt und diese können direkt angesprochen werden. Nur falls kein Änderungsmanager zur Verfügung steht und man den Koordinator entlasten will, könnte man die Aufforderung zur Migration und den globalen öffentlichen Workflow durchreichen lassen.

¹URI steht für Uniform Resource Identifier

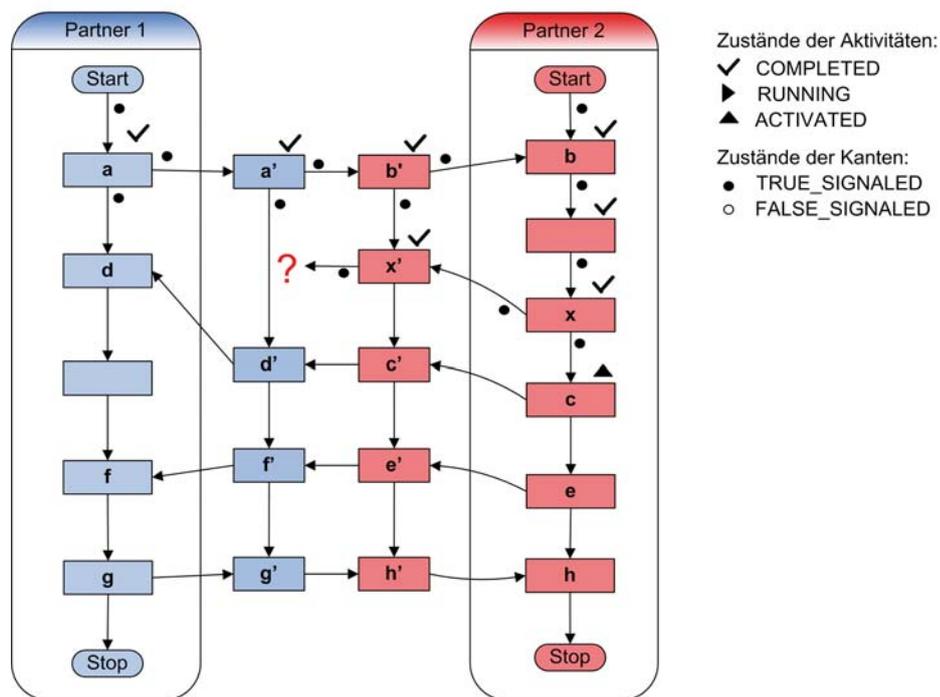


Abbildung 8.4: Verzögerte Migration

Bei Ansatz 4 wird die verkettete Durchführung angewendet, falls die Partner ihre URI nicht mit der Antwort angeben und der Initiator sie deshalb nicht direkt kontaktieren kann. Eventuell will man vermeiden, dass dem Koordinator entfernte Partner bekannt gegeben werden. Falls ein Hersteller sich z.B. das beste Angebot von verschiedenen Zulieferern aussucht, dann sollte der eine Zulieferer nichts von der Existenz des Konkurrenten wissen.

8.5 Optimistisches Verfahren

Beim optimistischen Verfahren entfällt der Schritt der globalen Entscheidungsfindung, d.h. der Initiator einer Änderung migriert ohne vorher sicherzustellen, dass auch die Partner migrieren können. Die Rücksetzbarkeit der Migration muss deshalb bei diesem Verfahren auf jeden Fall gewährleistet sein. Dabei müssen zwei Aspekte berücksichtigt werden.

Der erste Aspekt bezieht sich auf die Rücksetzbarkeit der Änderung und die Wiederherstellung des Zustandes der privaten Instanz des Partners vor der Migration. Zuerst muss also der Ablauf der Instanz bis zu dem Zeitpunkt vor der Änderung durch ein partielles Rollback "zurückgespult" werden [CCPP98]. ADEPT bietet z.B. solche Rollback-Mechanismen an. Sie gestatten es, die WF-Bearbeitung vor die Ausführung einer bestimmten Aktivität zurückzusetzen

(vorausgesetzt, die zugeordneten Aktivitätenprogramme unterstützen dies durch Bereitstellung von Storno- oder Kompensationsaktionen) [RRD02]. Zweitens müssen die strukturell vorgenommen Änderungen wieder rückgängig gemacht werden. Diese sind in der Ablaufhistorie (vorausgesetzt es wird eine geführt) dokumentiert und können in umgekehrter Reihenfolge nochmals angewendet werden.

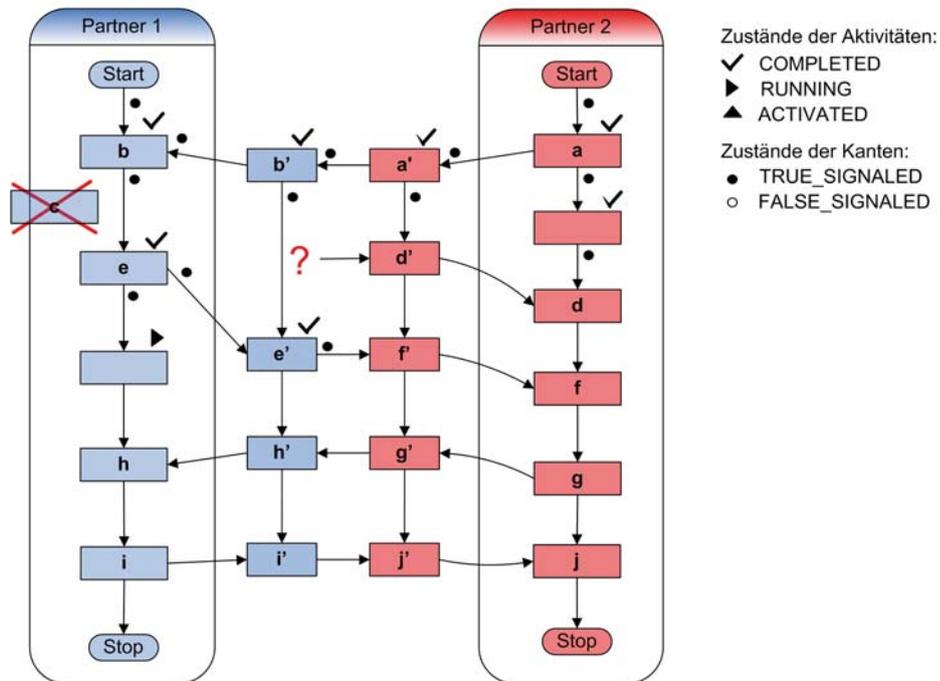


Abbildung 8.5: Konflikt - ein Partner migriert nicht

Diese lokale Rücksetzbarkeit genügt aber bei einer verteilten Choreographie-Instanz nicht. Bei dieser muss noch ein zweiter Aspekt berücksichtigt werden, der sich auf die Kooperation mit den Partnern bezieht. Es kann z.B. passieren, dass ein Partner der bereits migriert ist eine Nachricht verschickt, die nur aufgrund der Änderung existiert und von einem Partner der nicht migrieren konnte, deshalb nicht verarbeitet werden kann. Auch falls durch die Änderung eine vorher geplante Nachricht nicht mehr verschickt wird, kommt es zu Problemen. Eine solche Situation ist in Abbildung 8.5 exemplarisch dargestellt. Partner 1 ist bereits migriert und hat die kooperative Aktivität c gelöscht. Partner 2 führt die Migration seiner privaten Instanz aber nicht aus, z.B. aufgrund von Problemen durch kaskadierende Auswirkungen der Änderung, oder weil er mit der Änderung aus anderen Gründen nicht einverstanden ist. Die private Instanz von Partner 1 ist in ihrer Ausführung vorangeschritten und die kooperative Aktivität e wurde bereits beendet (bevor Partner 1 erfährt, dass Partner 2 nicht migriert ist). Partner 2 wartet aber auf die Nachricht der Aktivität c' und kann deshalb die Nachricht e' nicht verarbeiten und

es kommt zu einem Konflikt. Bei dieser Situation reicht es nicht nur die private Instanz von Partner 1 zurückzusetzen, es muss eine Lösung für die Einordnung der ‘falschen‘ Nachricht e’ gefunden werden. Dieses Problem wird in Kapitel 9 nochmals aufgegriffen.

Der Ablauf beim optimistischen Verfahren ist für Ansatz 3 ähnlich, wie die Entscheidungsfindung durch einen zentralen Koordinator.

- Ein Partner möchte eine Änderung an seinem privaten Workflow vornehmen. Deshalb überprüft er die Verträglichkeit seiner privaten WF-Instanz. Ist die Instanz verträglich, migriert er.
- Der Initiator der Änderung bestimmt die von der Änderung betroffenen Partner und die Modifikation des globalen öffentlichen Prozesses.
- Der Initiator sendet die nötigen Änderungen für die öffentlichen Aktivitäten an die betroffenen Partner.
- Die von der Änderung betroffenen Partner leiten die Modifikationen für ihren privaten Workflow ab und führen eine Verträglichkeitsprüfung durch.
- Die von der Änderung betroffenen Partner informieren den Initiator darüber, ob sie migrieren konnten.
- Falls der Initiator nur positive Antworten erhalten hat, ist die Migration erfolgreich verlaufen. Ansonsten meldet er allen betroffenen Partnern das Scheitern der Migration und alle bereits migrierten Partner müssen ein Rollback durchführen.

Bei Ansatz 4 ist der Ablauf prinzipiell derselbe wie bei Ansatz 3, nur die nötigen Änderungen müssen, wie bei den verketteten Verfahren, von Partner zu Partner durchgereicht werden.

Ein optimistisches Verfahren bietet sich eventuell an, falls der Initiator einer Änderung durch das lokale Entscheidungsverfahren bereits relativ sicher sein kann, dass auch die Partner mit der Änderung verträglich sind. Liefert das lokale Entscheidungsverfahren aber bereits das Ergebnis, dass es sich um eine unsichere Änderung handelt, besteht eine große Wahrscheinlichkeit, dass die Partner nicht migrieren können.

8.6 Versionierung

Bei einer Versionierung kommt es im Allgemeinen zu verschiedenen Versionen desselben Schemas und es existieren gleichzeitig laufende Instanzen sowohl auf Basis der einen Version als auch auf Basis einer anderen Version. Zu einer Versionierung kommt es dann, wenn eine Instanz nicht auf das neue Schema migrieren kann und deshalb auf dem alten Schema zu Ende läuft. Alternativ könnte man diese Instanz auch abrechnen oder nach Möglichkeit partiell zurücksetzen, bis sie mit dem neuen Schema verträglich ist. Ein besonderer Fall ergibt

sich bei Schleifen. Auch wenn eine Instanz im Moment mit der Änderung nicht verträglich ist, kann es sein, dass durch einen Schleifenrücksprung und dem damit verbundenen Zurücksetzen der Markierungen, die Verträglichkeit der Instanz zu einem späteren Zeitpunkt gegeben ist [RRD02].

Diese im vorherigen Absatz erläuterte Form der Versionierung, ergibt sich bei Prozess-Choreographien, wenn z.B. bei Ansatz 3 auf dem Schema eines globalen öffentlichen Prozesses mehrere Choreographie-Instanzen laufen und die Struktur nun angepasst werden soll. Dann muss für jede Choreographie-Instanz geprüft werden, ob sie migrieren kann und eventuell muss eine Kooperation nach dem alten "Vertrag" zu Ende laufen, während andere migrieren können. Eine Organisation kann in unterschiedliche unabhängige Kooperationen verwickelt sein und muss deshalb auch mehrere Instanzen verwalten. Die Koexistenz von Instanzen neuer und alter Versionen kann sichergestellt werden, indem eine Kopie des ursprünglichen Schemas angelegt wird, daran die Modifikationen ausgeführt werden und dann alle migrierbaren Instanzen dem neuen Schema zugeordnet werden [LRR05]. Etwas komplizierter wird es, wenn auch noch so genannte Ad-Hoc-Änderungen bei einer Instanz zugelassen werden. Das sind strukturelle Änderungen, die nur diese Instanz und nicht das zugrunde liegende Schema betreffen. Kommt es dann noch zusätzlich zu einer Schemaänderung, müssen die Ad-Hoc-Änderungen für diese Instanz bei der Verträglichkeitsprüfung zusätzlich berücksichtigt werden.

Zur einer zweiten Erscheinungsform einer Versionierung innerhalb einer Choreographie-Instanz kann es kommen, wenn die Partner eventuell veraltete Strukturen für die privaten Workflows der anderen (bei Ansatz 2) oder für den globalen öffentlichen Prozess (bei Ansatz 3 siehe Abbildung 8.3) vorliegen haben. Als Nebeneffekt dieser Versionierung kann es passieren, dass das lokale Entscheidungsverfahren die Änderungsmöglichkeiten positiver einschätzt, als ohne Versionierung. Theoretisch kommt es auch bei Ansatz 4 zu einer Versionierung, da nur die Partner eine Migration durchführen, die auch von der Änderung betroffen sind. Allerdings besteht in diesem Fall kein übergreifendes Schema auf dessen Basis die Choreographie-Instanz gestartet wird, sondern die bilaterale Beziehungen sind bei den Partner definiert und es können dynamisch Kooperationen mit neuen Partnern eingegangen werden. Ein öffentlicher Prozess, durch welchen bei Ansatz 4 der Ablauf jeder bilateralen Kooperation beschrieben wird, kann selbst aber nicht versioniert werden. Genauer gesagt darf es nicht passieren, dass wenn zwei Partner von einer Änderung betroffen sind, die Instanz des einen migriert und die des anderen nach dem alten Schema weiterläuft. Ansonsten würde es zu Inkonsistenzen beim Kommunikationsfluss zwischen den Partnern kommen und damit auch bei der Choreographie-Instanz.

9 Weiterführende Aspekte

In diesem Kapitel wird diskutiert, inwiefern sich die Probleme der bisher erarbeiteten Vorgehensweisen, durch die Anwendung von Methoden aus verwandten Themengebieten, lösen lassen. Außerdem wird untersucht, ob sich die bisher geforderte Bedingung, dass nur eine Änderung zur gleichen Zeit stattfindet, etwas aufweichen lässt oder sogar weniger restriktiv gesehen werden muss, wenn z.B. nicht alle Partner bekannt sind.

9.1 Verteilte Schnappschussgewinnung

Beim optimistischen Verfahren in Kapitel 8.5, kann es zu Problemen kommen, falls ein Partner nicht migrieren kann und ein migrierter Partner bereits weiterarbeitet. An dieser Stelle wäre es von Vorteil, wenn der nicht migrierte Partner bei den folgenden Nachrichten unterscheiden kann, ob der sendende Partner diese erst nach der Migration verschickt hat oder noch davor. In den verteilten Systemen gibt es die verteilte Schnappschussgewinnung, bei der Nachrichten markiert werden, welche erst nach einem bestimmten Ereignis verschickt worden sind, um zu erkennen, ob der Sender das Ereignis bereits ausgeführt hat oder nicht.

Für die verteilte Schnappschussgewinnung wird eine Methode von Chandy und Lamport eingesetzt [TS03]. Damit lässt sich eine verteilte Momentaufnahme gewinnen, welche den globalen Status reflektiert, in welchem sich das verteilte System zu einem bestimmten Zeitpunkt befunden hat. Der globale Status eines verteilten Systems besteht aus dem lokalen Status jedes Prozesses, zusammen mit den Nachrichten, die gerade übertragen werden. Die verteilte Schnappschussgewinnung berücksichtigt auch die Kanalzustände und arbeitet nebenläufig zur Anwendung (weshalb sie auch nicht für die Entscheidungsfindung bei Choreographien geeignet wäre).

Die Methode von Chandy und Lamport lässt sich auf das Problem im Zusammenhang mit dem optimistischen Verfahren nicht vollständig anwenden, da hier nicht alle Partner von einer Änderung betroffen sind und somit auch nicht bei allen dieses Ereignis eintritt. Deshalb wird nur die Grundidee dieser Methode, Nachrichten zu kennzeichnen, die erst nach einem bestimmten Ereignis verschickt werden, für das optimistische Verfahren aufgegriffen. Um verschiedene Änderungen unterscheiden zu können, bekommt jede Änderung hier eine eindeutige ID zugeteilt (z.B. die ID des Initiators plus eine laufende Nummer). In Anlehnung an das Beispiel

aus Abbildung 8.5 auf Seite 66, werden im folgenden Beispiel Nachrichten, die nach einer Migration versendet werden, für die anderen Partner gekennzeichnet. In Abbildung 9.1 ist der Ablauf der Kommunikation zwischen Partner 1 und 2 auf einer Zeitleiste dargestellt. Die private Instanz von Partner 1 migriert aufgrund der Änderung mit der ID 1. Jeder Nachricht, die der blaue Partner von nun an sendet, fügt er die ID der Änderung hinzu. Um das Beispiel möglichst einfach zu halten, ist der blaue Partner hier auch der Initiator der Änderung und fordert deshalb Partner 2 ebenfalls auf, die Änderung 1 durchzuführen. Doch Partner 2 kann nicht migrieren und gibt Partner 1 Bescheid, dass er nicht migrieren konnte. Daraufhin setzt Partner 2 seine private Instanz auf den Zustand vor der Änderung zurück. Leider hatte Partner 1 vor der Rücksetzung bereits eine Nachricht durch die öffentliche Aktivität e' an Partner 2 gesendet. Diese Nachricht führt bei Partner 2 zu einem Konflikt, da er auf eine Nachricht von c' wartet. Die Nachricht von der Aktivität e' ist aber durch die Änderungs-ID gekennzeichnet. Dadurch weiß Partner 2, dass diese Nachricht nach der Migration von Partner 2 verschickt wurde, und da die Änderung 1 abgebrochen wurde, kann Partner 2 ohne weitere Folgen die Nachricht verwerfen. Danach kommt es noch zu einer zweiten Änderung, bei der beide Partner migrieren können.

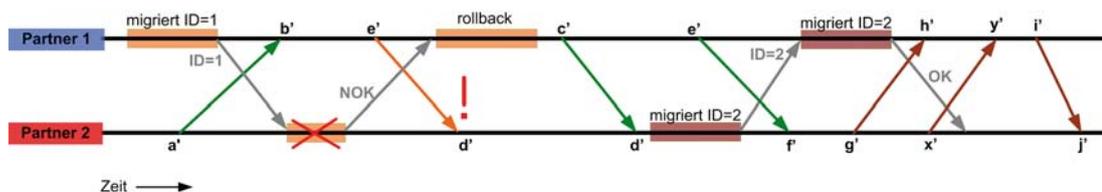


Abbildung 9.1: Verteilter Schnappschuss bei einer Choreographie-Instanz

Durch dieses Vorgehen besteht also die Möglichkeit zu verhindern, dass die Choreographie sich nach nur partieller Migration in einem Fehlerzustand befindet. Indem die nicht migrierten Partner wissen, ob die empfangene Nachricht vor oder nach der Migration eines Partners verschickt wurde, kann über die Handhabung entschieden werden. Trotzdem müssen noch weitere Überlegungen angestellt werden. Da eventuell nicht alle Partner von Änderung betroffen sind, bekommen auch nicht alle eine Aufforderung zur Migration. Die nicht von der Änderung betroffenen Partner bekommen durch den Abbruch einer Migration eventuell eine Nachricht von einem betroffenen Partner zweimal, einmal mit und einmal ohne die Änderungs-ID. Die zweite Nachricht ohne Änderungs-ID kann dann verworfen werden, obwohl sie die aktuellere ist, da sich an der Kooperation dieser beiden Partner nichts geändert hat.

9.2 Verteilter wechselseitiger Ausschluss

Bei der Verwendung eines zentralen Koordinators für die Durchführung von Änderungen einer Choreographie-Instanz, könnte es passieren, dass verschiedene Änderungen gleichzeitig durchgeführt werden. Falls dadurch Partner von mehreren Änderungen auf einmal betroffen sind, kann es zu Konflikten kommen. Diese Problematik ist allgemein betrachtet ebenso aus den verteilten Systemen bekannt. Durch den wechselseitigen Ausschluss kann in verteilten Systemen sichergestellt werden, dass auf eine gemeinsam genutzte Ressource nicht durch mehrere Prozesse¹ gleichzeitig zugegriffen wird. Wenn ein Prozess bestimmte gemeinsam genutzte Daten lesen oder aktualisieren muss, tritt er zuerst in einen kritischen Bereich ein, um einen wechselseitigen Ausschluss sicherzustellen und damit auch, dass kein anderer Prozess gleichzeitig diese Daten benutzt [TS03].

Mit der verteilten Lösung nach Lamport [TS03] ist es möglich eine Reihenfolge zu erzielen, die festlegt, welches Ereignis zuerst aufgetreten ist. Diese Reihenfolge kann genutzt werden, um Zeitstempel für einen verteilten wechselseitigen Ausschluss zu realisieren. Es wird mit logischen Zeitstempeln gearbeitet, weil es nicht wichtig ist welche genaue Zeit vorliegt (diese müsste aufwendig synchronisiert werden), sondern, dass sich alle über die Reihenfolge der Ereignisse einig sind.

Bei einer Choreographie-Instanz soll erreicht werden, dass nicht mehrere Änderungen gleichzeitig durchgeführt werden und die Partner mit verschiedenen Änderungswünschen konfrontiert werden. Demzufolge könnte eine Änderung in diesem Zusammenhang als die gemeinsam genutzte Ressource betrachtet werden, welche vor gleichzeitigem Zugriff geschützt werden soll. Nachfolgend wird versucht den Algorithmus für einen verteilten wechselseitigen Ausschluss bei einer Choreographie-Instanz anhand zweier Beispiele anzuwenden.

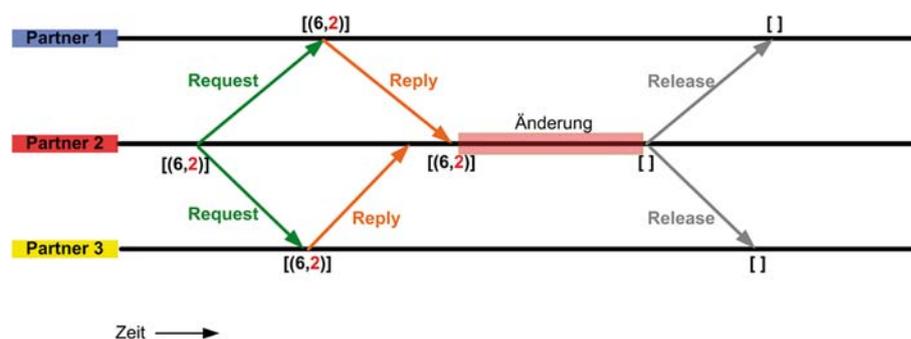


Abbildung 9.2: Verteilter wechselseitiger Ausschluss bei einer Änderung

¹Hier wird unter dem Begriff Prozess ein sich in Ausführung befindliches Programm verstanden.

In den Beispielen wird von einer Choreographie-Instanz mit drei involvierten Partner ausgegangen. In Abbildung 9.2 möchte Partner 2 eine Änderung durchführen. Deshalb sendet er an die beiden anderen Partner eine Request-Nachricht mit Lamport-Zeitstempel und seiner ID. Jeder Partner hat eine nach Zeitstempeln sortierte Warteschlange. Partner 1 und Partner 3 nehmen den Request von Partner zwei in ihre Warteschlange auf und antworten mit einer Reply-Nachricht. Sobald Partner 2 beide Reply-Nachrichten erhalten hat, kann er mit der Durchführung der Änderung (globales Entscheidungsverfahren und Migrationsverfahren) beginnen. Partner 2 sendet an die anderen Partner eine Release-Nachricht, sobald die Änderung abgeschlossen ist.

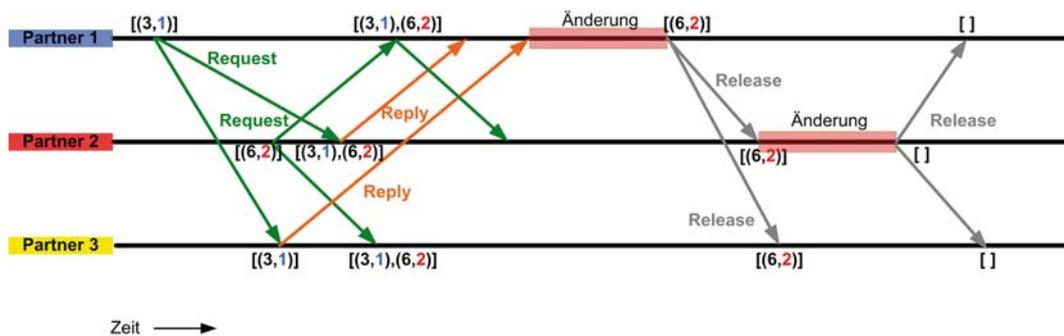


Abbildung 9.3: Verteilter wechselseitiger Ausschluss bei zwei Änderungen

In Abbildung 9.3 ist eine Situation dargestellt, bei der zwei Partner gleichzeitig eine Änderung durchführen wollen. Partner 1 und Partner 2 haben eine Änderung geplant und senden eine Request-Nachricht an die anderen Partner. Alle Partner nehmen die Anfragen in ihre Warteschlange auf und sortieren diese nach dem Lamport-Zeitstempel. Partner 1 kann seine Änderung durchführen, sobald er alle Reply-Nachrichten erhalten hat, da sein Request am Anfang der Warteschlange steht. Nach Beendigung der Änderung, entfernt Partner 1 seinen Request aus seiner Warteschlange und sendet eine Release-Nachricht an Partner 2 und Partner 3. Wenn an dieser Stelle sicher gestellt werden soll, dass auch die Partner 2 und 3 die Migration beendet haben, bevor die Release-Nachricht verschickt wird, könnte verlangt werden, dass die Partner nach der Migration noch eine Bestätigung an den Initiator senden und dieser erst dann mit Release die Änderung frei gibt. Diese entfernen dann ebenfalls den Request von Partner 1 aus ihrer Warteschlange und somit kann Partner 2 jetzt mit der gewünschte Änderung beginnen.

Offensichtlich besteht die Möglichkeit in einer Choreographie mit dem Algorithmus für den verteilten gegenseitigen Ausschluss alle Änderungen in serieller Reihenfolge durchzuführen. Allerdings gilt dies nur für eine Choreographie nach Ansatz 2 und 3, da bei diesen Ansätzen alle Partner bekannt sind. Außerdem müssen alle Partner eine Lamport-Uhr besitzen und durch

die Implementierung müssen zuverlässige FIFO-Kanäle² zwischen den Partnern gewährleistet werden. Der zusätzliche Kommunikationsaufwand, der bei diesem Vorgehen anfällt, ist aber nicht außer Acht zu lassen. Außerdem kommt es zu einer Verzögerungen bei der Änderungsdurchführung.

9.3 Verteilte Transaktionen

In einem anderen Themengebiet, den verteilten Datenbanken, kommt es zu einer ähnlichen Problematik wie bei der Evolution von Prozess-Choreographien. Deshalb wird in diesem Kapitel darauf eingegangen, inwiefern sich die Lösungsansätze der verteilten Datenbanken auch bei Änderungen in einer Prozess-Choreographie anwenden lassen.

Bei den verteilten Datenbanken wird mit dem Konzept der verteilten Transaktionen gearbeitet [Dad96]. Bei einer verteilten Transaktion, handelt es sich um eine Transaktion, die aus logisch zusammengehörigen Einzeloperationen besteht und auf verteilten Daten arbeitet. Dabei zerfällt die verteilte Transaktion in so genannte Subtransaktionen. Nach demselben Konzept laufen auch die Änderungen in einer Choreographie ab. Wenn der Initiator oder Koordinator eine Änderung bei der Choreographie-Instanz durchführen will, werden bei den anderen Partnern “Sub“-Änderungen ausgeführt. Außerdem lassen sich die in [Dad96] vorgestellten Aufrufstrukturen bei vollständigem Verteilungswissen und bei begrenztem Verteilungswissen auf das zentrale und verkettete Verfahren abbilden.

Die in dieser Arbeit gewählte Vorgehensweise zur Durchführung einer Änderung mit globaler Entscheidungsfindung und anschließender Migration, weist Parallelen zum Zwei-Phasen-Commit-Protokoll (2PC-Protokoll) auf, welches bei verteilten Transaktionen eingesetzt wird. Dabei werden zuerst alle beteiligten Transaktionen mit einer Prepare-to-Commit-Anweisung aufgefordert einen Ready-to-Commit-Zustand einzunehmen (falls die Transaktion erfolgreich abgeschlossen werden kann). Alle Transaktionen, die sich in diesem Zwischenzustand (welcher vergleichbar mit dem Änderungszustand in dieser Arbeit ist) befinden, garantieren, dass das Commit (falls gewünscht) durchgeführt wird, aber auch noch ein Abbruch möglich ist. Falls von allen Transaktionen ein “ok“ zurückgegeben wird, werden die Transaktionen mit einem Commit-Aufruf zur Durchführung aufgefordert, ansonsten erhalten sie eine Abort-Nachricht für den Abbruch der Transaktion.

Verteilte Transaktionen sorgen aber nicht nur für die Einhaltung des “Alles-oder-Nichts-Prinzips“ bei der Durchführung einer atomaren Operation, sondern sie können auch einen wechselseitigen Ausschluss realisieren, um eine Ressource vor gleichzeitigem Zugriff zu schützen.

²Die FIFO (First In First Out)-Kanäle garantieren, dass alle vor einem Reply gesendeten Requests auch vorher ankommen.

Dabei ermöglichen die verteilten Transaktionen mehr Nebenläufigkeit, d.h. auch gleichzeitig durchgeführte Änderungen, im Gegensatz zu dem im vorherigen Kapitel vorgestellten verteilten wechselseitigen Ausschluss. Es muss aber die Eigenschaft der Serialisierbarkeit erhalten bleiben, d.h. wenn zwei oder mehr Transaktionen gleichzeitig ausgeführt werden, sieht das Ergebnis so aus, als wären alle Transaktionen nacheinander in einer bestimmten Reihenfolge ausgeführt worden [Dad96]. Inwiefern das bei Choreographie-Instanzen auch erreicht werden kann wird im Folgenden diskutiert.

9.3.1 Nebenläufigkeitskontrolle

Das Ziel der Nebenläufigkeitskontrolle ist es mehrere Transaktionen gleichzeitig zu erlauben. Grundsätzlich unterscheidet man zwei Verfahren, die pessimistischen Verfahren und die optimistischen Verfahren. Bei den pessimistischen Verfahren geht man davon aus, dass Konflikte zwischen Transaktionen auftreten. Deshalb werden die Operationen synchronisiert, bevor sie ausgeführt werden, d.h. Konflikte werden gelöst, bevor sie überhaupt auftreten können. Optimistische Verfahren dagegen basieren auf dem Konzept, dass im Allgemeinen keine Konflikte auftreten. Die Operationen werden deshalb einfach ausgeführt und die Synchronisierung findet am Ende einer Transaktion statt. Stellt sich dann heraus, dass ein Konflikt aufgetreten ist, werden eine oder mehrere Transaktionen zurückgesetzt.

Das Konzept hinter der Nebenläufigkeitskontrolle ist, Konflikte erzeugende Operationen korrekt einzuplanen, wobei zwei Operationen einen Konflikt erzeugen, wenn sie auf demselben Datenelement arbeiten und mindestens eine davon eine Schreiboperation ist [TS03]. Das bedeutet für eine Choreographie-Instanz, dass es dann zu einem Konflikt kommt, falls die private Instanz eines Partners (die Teil der Choreographie-Instanz ist) von zwei Änderungsoperationen gleichzeitig betroffen ist. Ein pessimistisches Verfahren wäre, die private Instanz mit einer Sperre zu belegen, falls sie von einer Transaktion (also einer Änderungsdurchführung) verwendet wird.

Eine Realisierung der verketteten Verfahren mittels verteilten Transaktionen ist in Abbildung 9.4 dargestellt. Dabei wird mit Sperren für die privaten WF-Instanzen gearbeitet, damit eine private Instanz nur in eine Änderung gleichzeitig involviert sein kann. Jeder Partner verwaltet die Sperre für seine private Instanz selbst. Eine private Instanz ist gesperrt, solange sie sich im Änderungsstatus befindet, danach wird die Sperre durch die Transaktion aufgehoben und sie kann weiterlaufen. Transaktion T1 und alle Sub-Transaktionen realisieren bei erfolgreicher Durchführung sowohl die verkettete Abstimmung als auch die verkettete Migration. Bei diesem Lösungsansatz können aber durchaus mehrere Änderungen gleichzeitig in einer Choreographie-Instanz vollzogen werden, solange sie nicht die gleichen privaten Instanzen betreffen. Leider können bei der Verwendung von Sperren Verklemmungen auftreten, weshalb

Maßnahmen ergriffen werden müssen, um diese zu erkennen bzw. aufzulösen. Eine Möglichkeit wäre z.B. die Transaktion abubrechen, falls eine Sperre für eine private WF-Instanz, auch nach einem Timeout, nicht angefordert werden kann. Falls es zu einem Abbruch der Transaktion kommt, findet dieser immer bereits in der Phase der Entscheidungsfindung vor der Migration statt. Die Entscheidungsfindung wird dann einfach abgebrochen, die privaten Instanzen werden freigegeben und es sind keine weiteren Maßnahmen mehr notwendig.

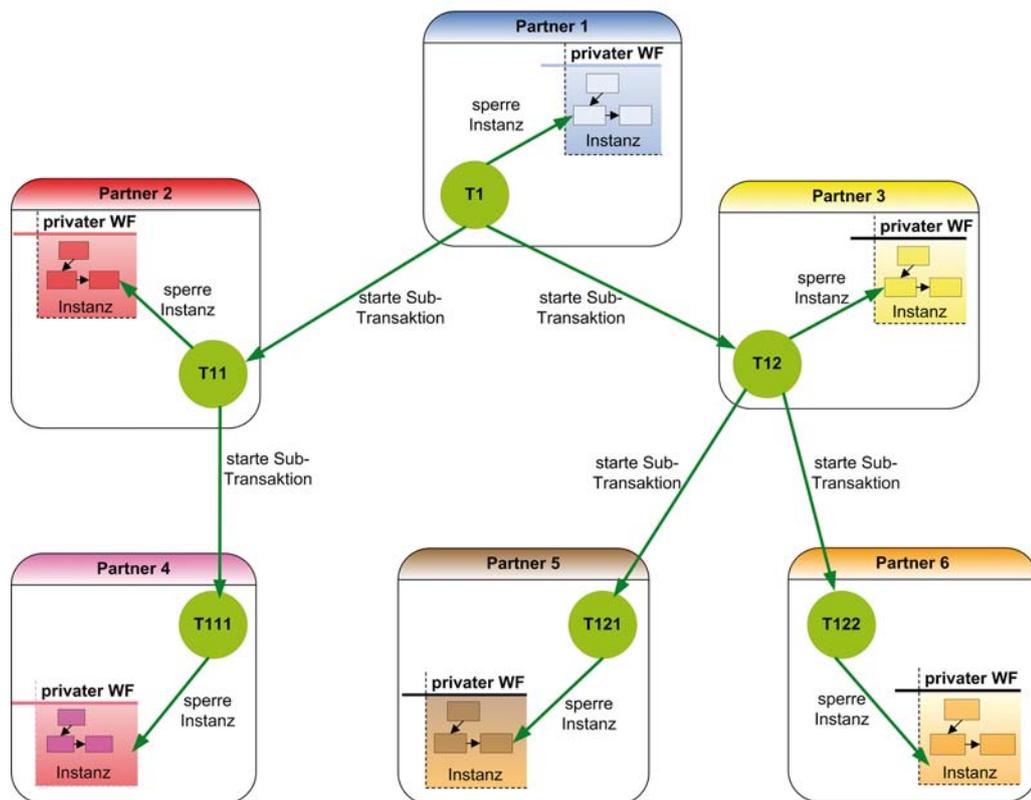


Abbildung 9.4: Verteilte Transaktion mit Sperren

Ein optimistisches Verfahren ermöglicht zwar mehr Nebenläufigkeit als ein pessimistisches Verfahren, aber die Implementierung ist vor allem im verteilten Fall recht aufwendig. Außerdem wird bei einer Änderung der Choreographie-Instanz innerhalb einer Transaktion immer lesend und schreibend auf die privaten Instanzen zugegriffen, weshalb es zwischen zwei parallel ausgeführten Transaktion immer zum Konflikt kommt, sobald sie auf die gleiche private Instanz zugreifen. In dieser Arbeit wurde ein optimistisches Migrationsverfahren vorgestellt, welches die Rücksetzbarkeit der nur partiell ausgeführten Migration verlangt. Die Beschreibung "optimistisch" bezieht sich in diesem Zusammenhang auf die Annahme, dass alle von der Änderung betroffenen Partner migrieren können und nicht wie im Kontext der Transaktionen

darauf, dass sich zwei parallele Änderungen nicht stören. Um beim optimistischen Migrationsverfahren zu gewährleisten, dass nur eine Änderung gleichzeitig stattfindet, kann auch auf das im vorherigen Absatz verwendete pessimistische Verfahren mit Sperren zurückgegriffen werden.

9.3.2 Eigenschaften einer Transaktion

Im Allgemeinen wird gefordert, dass eine Transaktion vier Eigenschaften, die so genannten ACID³-Eigenschaften, aufweist: atomar, konsistent, isoliert, dauerhaft [TS03]. Auf diese Eigenschaften wird im Folgenden im Zusammenhang mit dem Beispiel in Abbildung 9.4 kurz eingegangen.

Atomar bedeutet, dass die Operationen, welche innerhalb einer Transaktion ausgeführt werden, als eine logische Einheit gesehen werden. Die Transaktion T1 kann nur erfolgreich abgeschlossen werden, wenn alle ihre Subtransaktionen ebenfalls erfolgreich abgeschlossen werden können. Außerdem müssen alle Subtransaktionen durchgeführt werden, falls T1 durchgeführt wird. Dies wird durch das 2PC-Protokoll sichergestellt.

Eine Transaktion ist konsistent, wenn nach der vollständigen Durchführung der Transaktion die Konsistenz der Choreographie-Instanz nicht verletzt wird. Durch die Verträglichkeitsprüfung, die innerhalb einer Transaktion stattfindet und dadurch, dass die Änderungen erst durch das Commit (der Aufforderung zur Migration) "sichtbar" werden, ist auch diese Eigenschaft gegeben.

Die dritte Eigenschaft besagt, dass eine Transaktion isoliert oder serialisierbar sein muss. Die Serialisierbarkeit bei parallelen Transaktionen ist durch die Sperren auf die privaten Instanzen gewährleistet. Außerdem muss die Zurücksetzbarkeit einer Transaktion ermöglicht werden, falls es zu einem Konflikt kommt. Falls die private Instanz angehalten wird, sobald sie sich im Änderungsstatus befindet, können die strukturellen Änderungen für die Verträglichkeitsprüfung direkt am Objekt vorgenommen werden. In diesem Fall muss aber eine Änderungshistorie geführt werden, damit die vorgenommenen Änderungen bei einem Abbruch wieder rückgängig gemacht werden können. Falls die private Instanz im Änderungszustand jedoch bis zu den von der Änderung betroffenen Aktivitäten weiterlaufen darf, muss auf einer Kopie gearbeitet werden und erst bei der Migration wird die Original-Instanz ersetzt.

Eine Transaktion ist dauerhaft, wenn nach Durchführung der Transaktion die Änderungen erhalten bleiben. Nachdem die Migration durchgeführt wurde und die Sperren aufgehoben sind, wird mit den geänderten Instanzen weitergearbeitet. Die Änderungen können dann auch nicht mehr in Folge eines Systemabsturzes verloren gehen.

³ACID steht für Atomicity, Consistency, Isolation, Durability

Die Instanz-Migration in einer Prozess-Choreographie scheint sich demzufolge mit dem Transaktions-Konzept realisieren zu lassen. Auch die verkettete Entscheidungsfindung und Migrationsdurchführung einer Choreographie-Instanz nach Ansatz 4, kann bei mehreren gleichzeitigen Änderungen koordiniert werden, obwohl nicht alle Partner bekannt sind und somit auch nicht direkt angesprochen werden können. Die Forderung, dass nur eine Änderung auf einmal an einer Choreographie-Instanz vorgenommen werden darf, wird dabei etwas gelockert. Dies führt zwar eventuell zu behebbaren Konflikten zwischen den Änderungen, aber auch zu mehr Nebenläufigkeit.

10 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Problematik der Migration laufender Instanzen bei der Evolution von Prozess-Choreographien anhand von unterschiedlichen Voraussetzungen aufgearbeitet, um darauf aufbauend verschiedene Lösungsansätze für die Durchführung der Migration zu ermitteln. Dabei ging es vor allem darum, Möglichkeiten aufzuzeigen, wann und wie laufende Instanzen korrekt migrieren können. Die Darstellung der Prozess-Choreographien, welche dieser Arbeit zu Grunde gelegt wird, wurde erarbeitet, um die Problematik abstrahiert von der Komplexität des Themengebiets erörtern zu können. Da es Choreographien mit unterschiedlichem Hintergrund gibt, wurden vier Ansätze für eine Prozess-Choreographie vorgestellt, welche von verschiedenen Voraussetzungen ausgehen. Das Ziel bei der Durchführung einer Änderung ist, dass sich eine Instanz, welche sich vor der Änderung in einem korrekten Zustand befunden hat, sich danach auch wieder in einem korrekten Zustand befindet. Um dies zu gewährleisten, wurde ein Korrektheitskriterium abgeleitet, bei dessen Einhaltung die Korrektheit bei dynamischen Änderungen in einer Prozess-Choreographie gegeben sein sollte. Da die Änderungen aber mehrere Partner betreffen können und dann verteilt durchgeführt werden, mussten Verfahren erarbeitet werden, um diese Durchführung zu koordinieren. Zuerst wurde durch die lokalen Entscheidungsverfahren versucht, anhand der Informationen, die über die Partner vorhanden sind, eine Vorentscheidung zu treffen. Falls es sich hier bereits herausstellt, dass eine Migration der Choreographie-Instanz nicht möglich ist, müssen die Partner auch nicht befragt werden. Um aber wirklich sicher gehen zu können, dass auch alle Partner bei ihrer privaten Instanz, die für eine Migration nötigen Änderungen, durchführen können, wurden globale Entscheidungsverfahren entwickelt. Damit wird eine Entscheidungsfindung ermöglicht, welche alle betroffenen Partner einschließt. Ein globales Entscheidungsverfahren kann sowohl von einer zentralen Einheit, als auch verteilt durchgeführt werden. Welche Variante sich am besten eignet, hängt von dem jeweiligen Choreographie-Ansatz ab. Das gleiche gilt für die daran anschließenden Migrationsverfahren, welche die Möglichkeiten aufzeigen, die Durchführung der Migration über mehrere Partner hinweg zu realisieren. Nur das optimistische Migrationsverfahren verzichtet auf eine vorherige globale Entscheidungsfindung und verlässt sich darauf, dass alle Partner migrieren können. Aus diesem Grund muss bei diesem Verfahren auch die Rücksetzbarkeit einer bereits durchgeführten Migration gegeben sein. Abschließend wurde versucht, die erarbeiteten Vorgehensweisen bzw. die aufgezeigten Probleme bei der Durchführung von Änderungen in einer Choreographie-Instanz anhand von Methoden aus verwandten Themenbereichen zu lösen.

Die Erkenntnisse dieser Arbeit sollen als Basis für weitere Diskussionen bei der dynamischen Evolution von Prozess-Choreographien dienen. Als nächster Schritt, können die Überlegungen dieser Arbeit für eine bestimmte Prozess-Choreographie weiter ausgearbeitet und verfeinert werden. Interessant wäre noch die Datenflüsse eines Workflows und alternative Verzweigungen für eine Kooperation in der Darstellung zu realisieren und gegebenenfalls das Korrektheitskriterium und die Verfahren anzupassen.

Literaturverzeichnis

- [Aal99] AALST, W.M.P. van d.: Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets. In: *Systems Analysis - Modelling - Simulation* 34(3) (1999), S. 335–367
- [ACD⁺05] ANDREWS, Tony ; CURBERA, Francisco ; DHOLAKIA, Hitesh ; GOLAND, Yaron ; KLEIN, Johannes ; LEYMANN, Frank ; LIU, Kevin ; ROLLER, Dieter ; SMITH, Doug ; THATTE, Satish ; TRICKOVIC, Ivana ; WEERAWARANA, Sanjiva: Business Process Execution Language for Web Services (BPEL4WS) Version 1.1. (2005)
- [ADO⁺05] AALST, W.M.P. van d. ; DUMAS, M. ; OUYANG, C. ; ROZINAT, A. ; VERBEEK, H.M.W.: Choreography Conformance Checking: An Approach based on BPEL and Petri Nets. In: *BPM Center Report BPM-05-25* (2005)
- [AW01] AALST, Wil M. P. d. ; WESKE, Mathias: The P2P Approach to Interorganizational Workflows. In: *Lecture Notes in Computer Science* 2068 (2001), S. 140–156
- [BKKR03] BERNAUER, M. ; KAPPEL, G. ; KRAMLER, G. ; RETSCHITZEGGER, W.: Specification of Interorganizational Workflows - A Comparison of Approaches. In: *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)* (2003), S. 30–36
- [BR05] BENYOUCEF, Morad ; RINDERLE, Stefanie: Desel, J.; Frank, U. (Herausgeber): A Model-Driven Approach for the Rapid Development of E-Negotiation Systems. (2005), S. 80–93
- [BRD01a] BAUER, T. ; REICHERT, M. ; DADAM, P.: Effiziente Übertragung von Prozessinstanzdaten in verteilten Workflow-Management-Systemen. In: *Informatik - Forschung und Entwicklung* Band 16, Heft 2 (2001), S. 76–92
- [BRD01b] BAUER, Thomas ; REICHERT, Manfred ; DADAM, Peter: Adaptives und verteiltes Workflow-Management. In: *Datenbanksysteme in Büro, Technik und Wissenschaft* März (2001), S. 47–66
- [BRD01c] BAUER, Thomas ; REICHERT, Manfred ; DADAM, Peter: Dynamische Ablaufänderungen in verteilten Workflow-Management-Systemen. In: *Datenbank-Spektrum* 1 (2001), S. 68–77

- [CCPP98] CASATI, F. ; CERI, S. ; PERNICI, B. ; POZZI, G.: Workflow evolution. In: *Data Knowl. Eng.* 24 (1998), S. 211–238
- [CM05] CIMPIAN, Emilia ; MOCAN, Adrian: WSMX Process Mediation Based on Choreographies. In: *Proceedings of the 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management at the BPM 2005* (2005)
- [CTD04] CHEBBI, I. ; TATA, S. ; DUSTDAR, S.: The view-based approach to dynamic inter-organizational workflow cooperation. In: *Technical Report TUV-1841-2004-23; Technical University of Vienna* (2004)
- [Dad96] DADAM, Peter: *Verteilte Datenbanken und Client/server-systeme.: Grundlagen, Konzepte und Realisierungsformen.* Springer-Verlag, 1996
- [DBD06] DUMAS, Marlon ; BARROS, Alistair ; DECKER, Gero: Multi-staged and Multi-viewpoint Service Choreography Modelling. In: *Queensland University of Technology, QUT ePrints* (2006)
- [DBRS04] DUMAS, Marlon ; BENATALLAH, Boualem ; RUSSELL, Nick ; SPORK, Murray: A configurable matchmaking framework for electronic marketplaces. In: *Electronic Commerce Research and Applications* 3(1) (2004), S. 95–106
- [DZD06] DECKER, Gero ; ZAHA, Johannes M. ; DUMAS, Marlon: Execution Semantics for Service Choreographies. In: *Queensland University of Technology, QUT ePrints* (2006)
- [GALH01] GREFEN, Paul W. P. J. ; ABERER, Karl ; LUDWIG, Heiko ; HOFFNER, Yigal: CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. In: *IEEE Data Engineering Bulletin* 24 (2001), S. 52–57
- [LASS00] LAZCANO, A. ; ALONSO, G. ; SCHULDT, H. ; SCHULER, C.: The WISE approach to electronic commerce. In: *International Journal of Computer Systems Science and Engineering* 15 (2000)
- [LRR05] LAUER, Markus ; RINDERLE, Stefanie ; REICHERT, Manfred: Repräsentation von Schema- und Instanzobjekten in adaptiven ProzessManagementSystemen. In: *Proc. Workshop Geschäftsprozessorientierte Architekturen LNI P-51* (2005), September, S. 555–560
- [MBB⁺03] MEDJAHED, B. ; BENATALLAH, B. ; BOUGUETTAYA, A. ; NGU, A.H.H. ; ELMAGARMID, A.K.: Business-to-business interactions: issues and enabling technologies. In: *The VLDB Journal The International Journal on Very Large Data Bases* Volume 12, Number 1 / May, 2003 (2003), S. 59–85

- [MH05] MENDLING, J. ; HAFNER, M.: From Inter-Organizational Workflows to Process Execution: Generating BPEL from WS-CDL. In: *Meersman, R., Tari, Z., Herrero, P., eds.: Proceedings of OTM 2005*. (2005)
- [OY05a] ORRIENS, Bart ; YANG, Jian: Establishing and maintaining compatibility in service oriented business collaboration. In: *ICEC '05: Proceedings of the 7th international conference on Electronic commerce*. New York, NY, USA : ACM Press, 2005, S. 446–453
- [OY05b] ORRIENS, Bart ; YANG, Jian: Specification and Management of Policies in Service Oriented Business Collaboration. In: *Business Process Management (2005)*, S. 422–427
- [RD98] REICHERT, M. ; DADAM, P.: ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. In: *Journal of Intelligent Information Systems. Kluwer Academic Publ.* 10, No. 2 (1998), S. 93–129
- [RPG05] ROUACHED, Mohsen ; PERRIN, Olivier ; GODART, Claude: A Contract Layered Architecture for Regulating Cross-Organisational Business Processes. In: *Business Process Management (2005)*, S. 410–415
- [RRD02] RINDERLE, Stefanie ; REICHERT, Manfred ; DADAM, Peter: Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata. In: *Informatik - Forschung und Entwicklung* 17, Number 4 (2002), S. 177–197
- [RRD04] RINDERLE, Stefanie ; REICHERT, Manfred ; DADAM, Peter: Flexible Support of Team Processes by Adaptive Workflow Systems. In: *Distributed and Parallel Databases* 16, Number1 (2004), Juli, S. 91–116
- [RS04] REICHERT, Manfred ; STOLL, Dietmar: Komposition, Choreographie und Orchestrierung von Web Services. Ein Überblick. In: *EMISA Forum* Band 24, Heft 2 (2004), S. 21–32
- [RWR06] RINDERLE, Stefanie ; WOMBACHER, Andreas ; REICHERT, Manfred: Evolution of Process Choreographies in DYCHOR. In: *Proc. 14th Int'l Conf. on Cooperative Information Systems (CoopIS'06), 1 - 3 Nov 2006, Montpellier, France (2006)*, S. 273–290
- [Tat02] TATA, Samir: Policies for Cooperative Virtual Teams. In: *Lecture Notes in Computer Science* 2315/2002 (2002), S. 340
- [TS03] TANENBAUM, Andrew S. ; STEEN, Maarten van: *Verteilte Systeme : Grundlagen und Paradigmen*. Pearson Studium, 2003

- [VG03] VONK, Jochem ; GREFEN, Paul: Cross-Organizational Transaction Support for E-Services in Virtual Enterprises. In: *Distributed and Parallel Databases* Volume 14, Number 2 (2003), S. 137–172
- [WFMN04] WOMBACHER, A. ; FANKHAUSER, P. ; MAHLEKO, B. ; NEUHOLD, E.: Matchmaking for business processes based on choreographies. In: *e-Technology, e-Commerce and e-Service, 2004. EEE '04. 2004 IEEE International Conference on* (2004), S. 359– 368
- [Wom05] WOMBACHER, Andreas: Decentralized decision making protocol for service composition. In: *ICWS '05: Proceedings of the IEEE International Conference on Web Services (ICWS'05)* (2005), S. 203–210
- [ZLY05] ZHAO, Xiaohui ; LIU, Chengfei ; YANG, Yun: An Organisational Perspective on Collaborative Business Processes. In: *Business Process Management* (2005), S. 17–31

Name: Carolin Susanna Hitzer

Matrikelnummer: 430645

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Carolin Hitzer