

Adaptives Workflow-Management für kooperierende Workflows - ein Überblick

Ulrike Greiner

Abt. Datenbanken, Institut für Informatik, Universität Leipzig
greiner@informatik.uni-leipzig.de

Zusammenfassung

Der Einsatz von Workflow-Management-Systemen zur Ausführung von Geschäftsprozessen nimmt vor allem durch die Verbreitung des E-Commerce immer mehr zu. Gerade der elektronische Handel ist von der Kooperation zwischen verschiedenen Unternehmen (Business2Business) bzw. zwischen Unternehmen und Kunden (Business2Consumer) geprägt. Workflow-Systeme müssen deshalb kooperative Prozesse zwischen unterschiedlichen Unternehmen unterstützen. Eine weitere Anforderung ist, dass sie unerwartete Fehler und Ausnahmen während der Workflow-Ausführung behandeln können. In diesem Artikel werden zuerst Kooperationsformen für Workflows und Anforderungen an Workflow-Systeme für die Unterstützung kooperierender Workflows erarbeitet. Anschließend werden verschiedene Forschungsansätze bzgl. dieser Anforderungen verglichen, um einen Überblick über den gegenwärtigen Stand der Forschung und Entwicklung auf dem Gebiet kooperierender Workflow-Management-Systeme zu erhalten.

1 Motivation

Workflow-Management-Systeme werden immer öfter zur Ausführung von stark strukturierten und oft wiederholten Geschäftsprozessen in Unternehmen und Organisationen eingesetzt. Sie ermöglichen die getrennte Definition von Daten- und Kontrollfluss eines Workflows und bieten somit eine bessere Übersicht und Kontrolle über die eigentliche Prozesslogik (vgl. [41, 12, 25, 43]). Wird eine Instanz eines Workflows gestartet, so kontrolliert das Workflow-System die Ausführung, indem es beispielsweise die Aktivitäten den richtigen Bearbeitern zur richtigen Zeit zuweist und für die Bereitstellung der Daten sorgt. Auch CSCW-Systeme (Computer Supported Cooperated Work) unterstützen Zusammenarbeit und Datenaustausch zwischen mehreren Bearbeitern, kommen aber vor allem dann zum Einsatz, wenn kein stark strukturierter Prozess zugrunde liegt, der in einen Workflow abgebildet werden könnte (vgl. [3, 20]).

Durch die fortschreitende Verbreitung des elektronischen Handels (E-Commerce) zwischen Unternehmen (Business-to-Business, B2B) und zwischen Unternehmen und Kunden (Business-to-Consumer, B2C) entstehen immer häufiger inter-organisatorische Geschäftsprozesse, die Unternehmensgrenzen überschreiten und nicht von einem Unternehmen alleine sondern in Kooperation mit anderen ausgeführt werden. Ein Beispiel dafür ist das Outsourcing (Auslagern) von Teilen eines Prozesses, die ein Unternehmen nicht selber durchführen kann oder will. So kann man sich einen Versandhandel vorstellen, der die Waren durch einen externen Paketdienst ausliefern lässt.

Die an einem inter-organisatorischen Prozess beteiligten Partner benutzen unterschiedliche Workflow-Systeme, die die entsprechenden Workflows, die den inter-organisatorischen Prozess abbilden, ausführen sollen. Dazu ist es nötig, dass die Workflow-Systeme und die von ihnen ausgeführten Workflows kooperieren. So muss im Beispiel aus dem Versandhandel der Händler den Paketdienst beauftragen und ihm unter anderem die Ware, die Lieferadresse und bestimmte Lieferbedingungen übergeben. Der Paketdienst informiert den Händler nach erfolgreicher Zustellung.

Gerade in komplexen Anwendungsgebieten wie dem E-Commerce können aber immer auch Ausnahmen und Fehler auftreten, die es nötig machen, von der im Workflow vordefinierten Prozesslogik abzuweichen. Stellt der Paketzusteller aus dem obigen Beispiel fest, dass die Lie-

feradresse falsch ist, kann die Ware nicht wie geplant zugestellt werden, sondern es muss erst eine Rückfrage beim Versandhandel gestartet werden, um die korrekte Adresse zu ermitteln. Solche Ausnahmen können zum Teil schon bei der Definition des Workflows berücksichtigt und z. B. durch die Angabe alternativer Pfade abgefangen werden. Andere unerwartete Fehler müssen dynamisch zur Laufzeit, z. B. durch eine manuelle oder teilautomatische Adaptation des Kontroll- und Datenflusses, behandelt werden.

Tritt in einem Workflow des inter-organisatorischen Prozesses ein Fehler auf, der lokal z. B. durch Adaptationen behandelt wird, können sich diese Adaptationen auch auf kooperierende Workflows des gleichen Prozesses in anderen Workflow-Systemen auswirken. Der Versandhändler muss beispielsweise bei einer falschen Lieferanschrift die richtige Adresse erst durch Nachfrage beim Kunden ermitteln. Dadurch verzögert sich die weitere Ausführung der Workflows und ursprünglich vereinbarten Liefer- und Zahlungstermine sind eventuell nicht mehr einzuhalten. Wird der Händler aber über den Fehler beim Zusteller informiert, kann er parallel zur Ermittlung der richtigen Anschrift den noch nicht ausgeführten Teil des Workflows überprüfen, ob z. B. Fristen für Lieferung und Zahlung durch die Verzögerung überschritten werden, und den Kunden entsprechend informieren bzw. Zeitlimits für das Erstellen von Mahnungen verschieben.

Workflow-Kooperation und Ausnahmebehandlung bei unvorhergesehenen Ereignissen hängen also bei der Ausführung inter-organisatorischer Prozesse eng zusammen, da eine lokale Ausnahmebehandlung Implikationen für die kooperierenden Workflows haben kann. Bisher wurden die beiden Aspekte der Kooperation und der Ausnahmebehandlung häufig getrennt untersucht. So wurde beispielsweise durch die Workflow Management Coalition (WfMC) eine Interface-Beschreibung (Interface 4) für die Kooperation von Workflow-Engines entwickelt ([40]) jedoch ohne Ausnahmebehandlung. Im Rahmen von dynamischem Workflow-Management wurde vor allem dynamische Ausnahmebehandlung innerhalb eines Workflows untersucht ([33, 36, 21]).

In den meisten kommerziellen Systemen sind keine Mechanismen für eine dynamische Fehlerbehandlung zur Laufzeit vorhanden und auch eine Kooperation zwischen verschiedenen Workflows wird nicht unterstützt. In der Forschung gibt es für die Kooperation von Workflows unterschiedliche Ansätze, die die Implikationen, die lokale Ausnahmebehandlungen für kooperierende Workflows haben können, teilweise auch berücksichtigen. Dieser Artikel soll zu einem besseren Verständnis der Anforderungen beitragen, die an Workflow-Management-Systeme bzgl. der Unterstützung kooperierender Workflows und der Ausnahmebehandlung gestellt werden. Darauf aufbauend können die verschiedenen Forschungsansätze dann untersucht und verglichen werden.

Der Artikel ist folgendermaßen aufgebaut: Zuerst werden verschiedene Kooperationsformen für Workflows klassifiziert (Abschnitt 2). Danach werden Vergleichskriterien für die Beurteilung verschiedener Forschungsansätze zur Workflow-Kooperation erstellt (Abschnitt 3) und anschließend auf einige Ansätze angewendet (Abschnitt 4-6). Zum Schluss werden die Ergebnisse zusammengefasst und weitere Schritte skizziert (Abschnitt 7).

2 Kooperationsformen für Workflows in inter-organisatorischen Prozessen

Wir unterscheiden zwei Kooperationsformen: *synchrone* und *asynchrone* Kooperation. Bei einer synchronen Kooperation wird ähnlich wie bei einem Remote Procedure Call (RPC) ein Aufruf an den kooperierenden Workflow geschickt und dann auf die Rückgabe von Ergebnissen gewartet. Die Ausführung des aufrufenden Workflows wird solange unterbrochen. Bei einer

asynchronen Kooperation hingegen wird nur ein Aufruf an den kooperierenden Workflow geschickt, die Rückgabe von Ergebnissen erfolgt - falls vorgesehen - in einem neuen Kooperationsschritt, der vom aufgerufenen Workflow ausgelöst wird. Eine Unterbrechung des aufrufenden Workflow ist also nicht nötig.

Die synchrone Workflow-Kooperation wird typischerweise über sogenannte *Dienste* realisiert. Dienste bestehen aus einer Folge von Aktivitäten (als separater Workflow oder als Teil eines anderen Workflows), die nach Aufruf durch einen kooperierenden Workflow für diesen ausgeführt werden und ein Ergebnis zurückliefern. Dienste haben einen Namen, eine Beschreibung und Parameter. Sie unterstützen vor allem klar strukturierte Kooperationsaufgaben.

Typischerweise erfolgt die Realisierung einer asynchronen Kooperation über sogenannte *Ereignisse*. Ereignisse repräsentieren einen Nachrichten-basierten Aufrufmechanismus, der es ermöglicht, Daten und Informationen, die von Aktivitäten eines Workflows erzeugt werden, an kooperierende Workflows zu schicken, deren Aktivitäten diese Ereignisse verarbeiten. Ereignisse sind typisiert, d. h. sie haben einen Namen, eine Struktur und bestimmte Operationen, die auf den Ausprägungen diese Typs aufgerufen werden können. Sie ermöglichen eine sehr flexible Kooperation, da sie beliebig zwischen den kooperierenden Workflows hin- und hergeschickt werden können.

Tabelle 1 fasst die beiden Kooperationsformen, die im Folgenden genauer beschrieben werden, ihre Realisierungsformen und Eigenschaften zusammen. Beide Kooperationsformen sind sowohl bei homogenen (alle Organisationen verwenden das gleiche Workflow-System) als auch bei heterogenen Workflow-Systemen (die Organisationen verwenden unterschiedliche Workflow-Systeme) einsetzbar, deshalb werden die beiden Fälle im Folgenden nicht unterschieden. Da kommerzielle Systeme jedoch meist nur eine Kooperation homogener Workflow-Systemen unterstützen, wird in Abschnitt 4 nochmal kurz auf diesen Spezialfall eingegangen.

In Abschnitt 2.3 werden Kooperationsbedingungen definiert und erläutert, Abschnitt 2.4 stellt intelligente Agenten und ihre Einsatzmöglichkeiten beim Workflow-Management vor und in Abschnitt 2.5 wird noch auf verschiedene Aspekte der technischen Durchführung der Kooperation wie Kommunikationsmechanismen oder das Datenformat eingegangen.

Kooperationsform	synchron	asynchron
typische Realisierungsform	Dienste	Ereignisse / Dienste
Definition	Folge von Aktivitäten, die nach Aufruf für den kooperierenden Workflow ausgeführt werden und Ergebnisse zurückliefern	Nachrichten-basierter Aufrufmechanismus, mit dem Daten und Informationen, die Aktivitäten eines Workflow erzeugen, an kooperierende Workflows geschickt werden, deren Aktivitäten diese verarbeiten
Eigenschaften	klar strukturierte Kooperation	hohe Flexibilität der Kooperation

Tabelle 1: Kooperationsformen.

2.1 Synchrone Kooperation

Bei der synchronen Kooperation werden in einem Katalog (beispielsweise im Internet) die verfügbaren Dienste der beteiligten Organisationen veröffentlicht. Die veröffentlichten Informationen umfassen den Dienstenamen, die Dienstbeschreibung einschließlich der Rückgabewerte, eine Aufrufschnittstelle, über die der kooperierende Workflow den Dienst starten kann, und

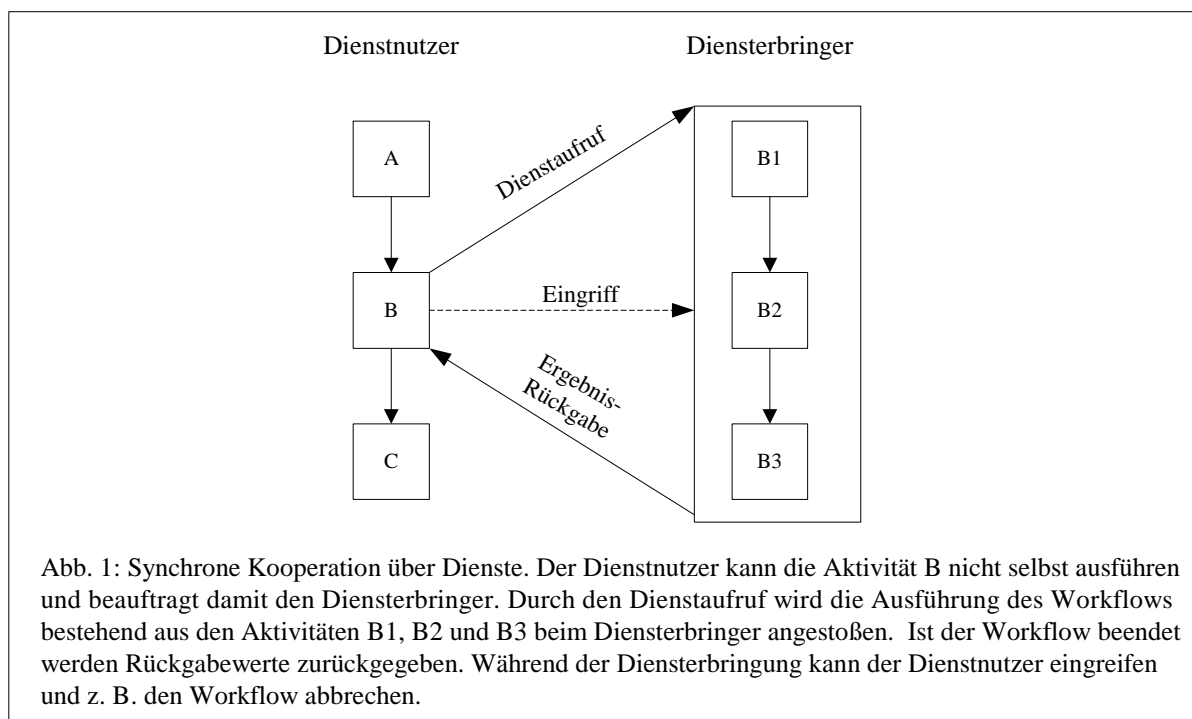
eventuell weitere Parameter wie zum Beispiel Kosten oder Übergabeparameter.

Will eine Organisation einen Dienst in ihren Workflow integrieren, sucht sie über den Katalog einen passenden Anbieter und fügt in ihren Workflow eine Aktivität ein, die über die öffentliche Schnittstelle den Dienst startet und dann solange wartet, bis der kooperierende Workflow Rückgabewerte liefert. Danach wird die Ausführung des aufrufenden Workflows fortgesetzt.

Der Zugriff auf den Workflow des Diensteanbieters erfolgt nur über die im Katalog spezifizierte Schnittstelle, somit kann der Anbieter seine Autonomie bzgl. der internen Realisierung des Dienstes wahren. Nach [30] besteht ein synchroner Kooperationsschritt aus drei Teilschritten:

- (i) dem Start des Dienstes,
- (ii) einem optionalen Eingriff während der Dienstauführung, um beispielsweise die Workflow-Ausführung abbrechen, weil der aufrufende Workflow abgebrochen wurde,
- (iii) der Rückgabe von Ergebnissen.

In Abbildung 1 ist der Ablauf einer synchronen Kooperation zwischen zwei Workflows schematisch dargestellt.



Die Dienste einer synchronen Kooperation können einfache Dienste wie Informationsdienste sein, die zu einem Namen die passende E-Mail-Adresse oder Telefonnummer zurückliefern. Vor allem im E-Commerce treten aber meist komplexe Dienste wie die Erstellung von Gutachten, die Auslieferung eines Produkts, die Herstellung oder Beschaffung von Gütern auf, für die es häufig auch unterschiedliche Anbieter mit unterschiedlichen Konditionen gibt. So kann eine Organisation, die ein Paket ausliefern will, unter mehreren Anbietern wählen. Sind für die Dienstbringung besondere Konditionen nötig (z. B. bei besonders empfindlicher oder sperriger Ware), die in der Dienstbeschreibung nicht enthalten sind, oder erfüllt kein Angebot alle Anforderungen des Dienstnutzers, können Verhandlungen zwischen Dienstanbieter und Dienstnutzer erforderlich werden. Das Ergebnis der Verhandlungen wird in einem Vertrag festgehalten.

Ob Verhandlungen und ein Vertragsabschluss nötig sind, hängt von der Komplexität eines Dienstes ab. Bei einem einfachen Informationsdienst wäre dieser Aufwand sicher übertrieben, während bei umfangreichen Diensten wie der Auslieferung von Produkten oder der Erstellung von Gutachten meist eine Menge von Randbedingungen festgelegt werden müssen, die einen Vertrag rechtfertigen. Die Verhandlungen und der Abschluss eines Vertrages können z. B. durch intelligente Agenten durchgeführt werden, wobei die Kooperation zwischen den Agenten synchron oder asynchron sein kann. Auf die Eigenschaften intelligenter Agenten und ihre Einsatzmöglichkeiten bei der Ausführung inter-organisatorischer Workflows wird in Abschnitt 2.4 genauer eingegangen.

Die WfMC-Spezifikation zur Interoperabilität von Workflow-Engines (Interface 4, vgl. [40]) enthält das Modell *geschachtelter Subprozesse* (nested subprocess), das unserer synchronen Kooperation entspricht. Dabei veranlasst ein Workflow den Start einer Workflow-Instanz auf einer anderen Workflow-Engine und unterbricht seine Ausführung bis diese Prozessinstanz beendet ist.

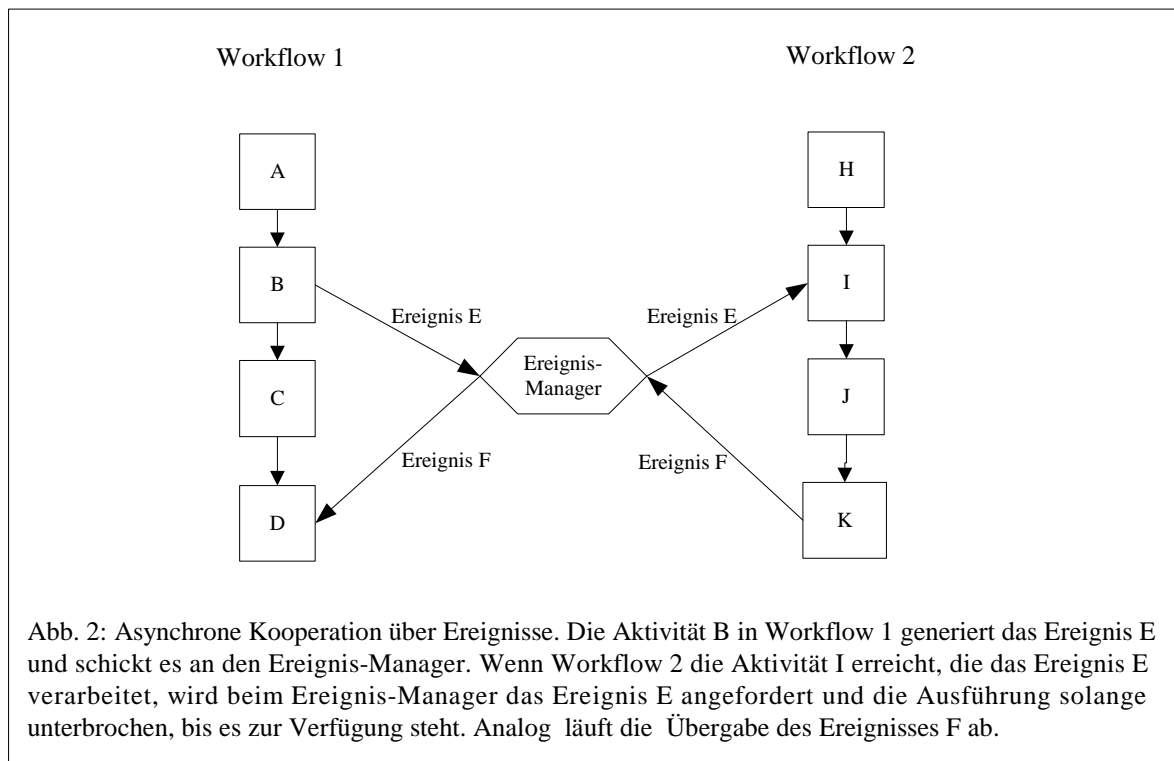
2.2 Asynchrone Kooperation

Die asynchrone Kooperation läuft über einen *send/receive*-Mechanismus ab. Aktivitäten, die Daten oder Informationen erzeugen, die Aktivitäten in einem anderen Workflow benötigen, schicken diese Daten, sobald sie sie erzeugt haben, als Ereignis über eine *send*-Funktion an einen Ereignis-Manager. Aktivitäten in einem anderen Workflow, die Daten aus anderen Workflows benötigen, holen die entsprechenden Ereignisse über eine *receive*-Funktion vom Ereignis-Manager und setzen dann ihre Ausführung fort. Liegt zum Zeitpunkt der Anforderung kein passendes Ereignis für eine Aktivität vor, wartet sie bis ein Ereignis eintrifft. Um zu verhindern, dass eine Aktivität ewig wartet, können z. B. Zeitlimits definiert werden. Liegt bei Ablauf eines solchen Limits immer noch kein passendes Ereignis vor, können die benötigten Daten z. B. über eine Nutzereingabe angefordert werden. Abbildung 2 zeigt den Ablauf der Kooperation zweier Workflows über Ereignisse.

Ereignisse sind typisiert, d. h. jedes Ereignis ist Instanz eines bestimmten Ereignistyps. Jede Aktivität kann nur Ereignisse bestimmter Typen erzeugen und verarbeiten. Die entsprechenden Typen können beispielsweise in der Aktivitätsdefinition festgelegt werden. Ereignissen und Anfragen an den Ereignis-Manager können Prioritäten zugeordnet werden, wobei dieser jeweils das Ereignis mit der größten Priorität zuerst ausliefert bzw. die Anfrage mit der höchsten Priorität zuerst bearbeitet. Bleiben dann für weitere Anfragen anderer Aktivitäten keine Ereignis-Instanzen mehr übrig, müssen diese warten, bis neue Ereignisse eintreffen.

Der Einsatz eines Ereignis-Managers ermöglicht eine flexible asynchrone Kooperation, da die kooperierenden Workflows nicht so eng gekoppelt sind wie bei der synchronen Kooperation. So muss beispielsweise der Erzeugeraktivität die Empfängeraktivität des Ereignisses nicht bekannt sein, da diese sich das Ereignis selbst vom Ereignis-Manager holt. Außerdem wird die Ausführung eines Workflows nur dann unterbrochen, wenn eine Aktivität auf ein Ereignis warten muss.

Der Ereignis-Manager kann entweder Teil der Workflow-Engine oder ein externes Werkzeug sein. Ist er Teil der Engine, können über ihn nur Ereignisse zwischen den Workflows ausgetauscht werden, die von dieser Engine ausgeführt werden. Sollen Ereignisse auch Workflows in anderen Workflow-Systemen zur Verfügung stehen, erfolgt der Austausch zwischen den Ereignis-Managern über einen *Publish-/Subscribe*-Mechanismus. Dabei veröffentlicht jeder Ereignis-Manager, zu welchen Ereignistypen bei ihm Ereignisse vorliegen, und die anderen Ereignis-Manager können diese abonnieren. Ist der Ereignis-Manager hingegen unabhängig von der Workflow-Engine, kann ein Ereignis-Manager den Austausch von Ereignissen zwischen mehreren Workflow-Systemen koordinieren. Es ist aber natürlich auch hier möglich zwischen meh-



reren Ereignis-Managern Ereignisse über einen Publish-/Subscribe-Mechanismus auszutauschen.

Die WfMC-Spezifikation für Interface 4 (vgl. [40]) enthält kein Modell, das einer asynchronen Kooperation über Ereignisse in unserem Sinne entspricht. Das Modell *verketteter Prozesse* (chained processes), bei dem eine Workflow-Instanz eine Workflow-Instanz auf einer anderen Workflow-Engine startet und danach ihre Ausführung fortsetzt ohne auf Rückgabewerte zu warten, ist eine asynchrone Kooperation, die aber keinen Austausch von Daten und Informationen enthält. Mit unseren Definitionen könnte man dieses Modell als *asynchronen Dienst* bezeichnen, da ebenfalls ein Workflow auf Initiative eines anderen Workflows ausgeführt wird, aber keine Ergebnisse zurückgegeben werden. Die Beschreibung von Interface 4 enthält noch das Modell *parallel synchronisierter Interoperabilität* (parallel synchronized). Dabei werden zwei Workflow-Instanzen auf unterschiedlichen Workflow-Engines parallel und unabhängig voneinander bis zu einem bestimmten, bei der Workflow-Definition festgelegten Rendezvous-Punkt ausgeführt. An diesem Punkt warten die Instanzen, bis die andere den Punkt auch erreicht hat und tauschen dann Daten aus. Damit wird der Datenaustausch zwischen Workflow-Instanzen unterstützt, die Kooperation ist aber nicht mehr asynchron, da auf die jeweils langsamere Instanz gewartet werden muss.

2.3 Kooperationsbedingungen

Sowohl bei einer synchronen also auch bei einer asynchronen Kooperation können sogenannte *Kooperationsbedingungen* gestellt werden. Diese beschreiben, welche temporalen und inhaltlichen Bedingungen für die Kooperation gelten.

Temporale Bedingungen können z. B. in Form von Zeitlimits angeben, dass ein Dienst (bei synchroner Kooperation) bis zu einem bestimmten Termin ausgeführt werden soll. Gelingt es dem Diensterbringer nicht, die Limits einzuhalten, muss er entweder Maßnahmen ergreifen, um die Limits doch noch einzuhalten oder den kooperierenden Workflow über die Verletzung des Li-

mits rechtzeitig benachrichtigen. Auch bei einer asynchronen Kooperation ist die Angabe von Zeitlimits möglich, beispielsweise für die Antwort auf einen receive-Aufruf an den Ereignis-Manager, damit die Ausführung des Workflows nicht zu lange verzögert wird. Ist bekannt, dass ein kooperierender Workflow ein Ereignis spätestens zu einem bestimmten Zeitpunkt benötigt, kann es sinnvoll sein, ein Zeitlimit festzulegen, bis zu dem ein Ereignis von einer Aktivität erzeugt werden muss.

Inhaltliche Bedingungen können z. B. beschreiben, welche Parameter bei einem Dienstaufwurf obligatorisch oder optional sind, welche Wertebereiche für Rückgabewerte oder Daten von Ereignissen gelten, welche Kosten für die Dienstleistung anfallen oder welche Schranken für diese Kosten gelten. Inhaltliche Bedingungen für die Daten von Ereignissen können auch verwendet werden, um deren Vollständigkeit zu sichern. Werden sie bei einem receive-Aufruf an den Ereignis-Manager angegeben, kann dieser bereits prüfen, ob die vorliegende Ereignis-Instanz auch wirklich den Anforderungen der Aktivität entspricht. So wird verhindert, dass die Aktivität ein Ereignis bekommt, das nicht ihren Anforderungen entspricht. Bekommt die Aktivität bis zum Ablauf des Zeitlimits des receive-Aufruf kein passendes Ereignis, kann sie z. B. einen receive-Aufruf mit geringeren inhaltlichen Bedingungen senden, die ein bereits beim Ereignis-Manager vorliegendes Ereignis erfüllt.

Kooperationsbedingungen sind auch bei der Suche nach einem Kooperationspartner und Vertragsverhandlungen, wie sie z. B. bei einer synchronen Kooperation über Dienste erforderlich sein können, hilfreich, da mit ihnen die Ziele der jeweiligen Workflows und Organisationen spezifiziert werden können. Wenn z. B. in einem Reisebuchungs-Workflow ein Angebot für einen Flug gesucht wird, können eine obere Preisgrenze, eine Beschränkung der Umsteigehäufigkeit und Parameter für ein Ranking der Ergebnisse (z. B. nach Preis, nach Umsteigehäufigkeit) angegeben werden. Diese Bedingungen können von intelligenten Agenten oder sonstigen geeigneten Mechanismen, die die Suche und die Aushandlung der Verträge übernehmen, benutzt werden. Werden die Kooperationsbedingungen als Folge von Verhandlungen Bestandteil eines Vertrags, werden sie auch als *Kooperationsvereinbarungen* bezeichnet. In der Literatur ist auch von Qualitätsbedingungen die Rede (vgl. [34]), womit meist nur inhaltliche Kooperationsbedingungen gemeint sind. Temporale Bedingungen werden davon unterschieden. Wir haben den allgemeineren Begriff Kooperationsbedingungen gewählt, weil auch temporale Bedingungen eine Form von Qualität darstellen und deshalb nicht von inhaltlichen Bedingungen getrennt werden sollten.

2.4 Intelligente Agenten

Wie schon weiter oben erwähnt können verschiedene Aufgaben bei der Ausführung kooperierender Workflows von intelligenten Agenten übernommen werden. In diesem Abschnitt soll nun näher erläutert werden, welche Eigenschaften intelligente Agenten besitzen und wie sie bei der Ausführung inter-organisatorischer Workflows eingesetzt werden können.

Nach [44] haben intelligente Agenten folgende Eigenschaften:

- *Autonomie*: Die Agenten können innerhalb bestimmter vom Nutzer vorgegebener Grenzen selbständig handeln und überwachen ihre Handlungen und ihren internen Zustand.
- *Soziale Fähigkeiten*: Agenten kommunizieren und kooperieren mit anderen Agenten und menschlichen Benutzer über eine Agenten-Kommunikationssprache, um ihre Ziele zu erreichen. Die Kommunikation ermöglicht es dabei nach [5] den Agenten nur, eine Menge vorgegebener Anfragen (festgelegt in einem Protokoll) an andere Agenten oder externe Ressourcen zu stellen und eine Menge vorgegebener Antworten zu erhalten, während die Kooperation erweiterte Fähigkeiten umfasst, die nötig sind, um die Lösung

komplexer Aufgaben durch mehrere Agenten zu realisieren.

- *Reaktivität*: Agenten überwachen ihre Umgebung und reagieren auf Änderungen.
- *Proaktivität*: Dies bezeichnet die Fähigkeit von Agenten nicht nur auf Änderungen in ihrer Umgebung zu reagieren, sondern auch zu agieren, um ihre Ziele zu erreichen.

Weitere Fähigkeiten intelligenter Agenten nach [5] sind *Lernfähigkeit*, d. h. die Fähigkeit Schlüsse aus Ereignissen oder Nutzerbewertungen zu ziehen, um z. B. Benutzerprofile anzupassen, und *Mobilität*, also die Fähigkeit sich in einem Netzwerk zu bewegen.

Agenten unterscheiden sich von Workflows darin, dass sie nur für genau definierte Aufgaben (z. B. die Suche nach Angeboten nach bestimmten Kriterien) mit einer beschränkten Dauer (etwa einige Stunden) eingesetzt werden, während Workflows langlebige Prozesse modellieren, die aus einer Menge solcher exakt definierten Teilaufgaben bestehen. Zur Erledigung ihrer Aufgaben können sich Agenten beliebig in Rechnernetzen bewegen, also über das Internet unterschiedliche Organisationen in verschiedenen Ländern besuchen. Die Ausführung von Workflows dagegen beschränkt sich meist auf ein Unternehmen oder bei Kooperation zwischen mehreren Workflows auf eine begrenzte Anzahl von Unternehmen.

Agenten und Workflows lassen sich koppeln, indem z. B. die einzelnen Aktivitäten des Workflows durch Agenten ausgeführt werden, die auf die jeweilige Aktivität spezialisiert sind. Dadurch vergrößert sich der Aktionsradius des Workflows, weil die Agenten sich zur Erledigung ihrer Teilaufgaben beliebig bewegen können, und die Beschränkung der Agenten auf spezifische, zeitlich begrenzte Aufgaben fällt nicht mehr so stark ins Gewicht, weil für die Ausführung eines Workflows mehrere Agenten zusammenarbeiten.

Bei dem in [38] beschriebenen Ansatz (vgl. Abb. 3) wird eine zusätzliche Agentenschicht über ein Workflow-System gelegt, deren Agenten nicht für die Ausführung einzelner Aktivitäten sondern für die Bereitstellung von Workflows, die Überwachung der Ausführung, die Fehlerbehebung und die Kooperation verantwortlich sind. Die Bereitstellung von Workflows erfolgt dadurch, dass den Agenten ein bestimmtes Ziel vorgegeben wird, für das sie in Zusammenarbeit einen Plan (also den Workflow) erstellen, den das Workflow-System dann ausführt. Treten Fehler auf, muss der Plan (also der Workflow) wiederum in Zusammenarbeit mit den anderen Agenten geändert werden. Für die Kooperation zwischen verschiedenen Workflow-Systemen sind Broker-Agenten verantwortlich. Jedes kooperierende Workflow-System muss einen solchen Broker-Agenten besitzen, der die Übersetzung zwischen dem lokalen Workflow-Modell und den Modellen der Kooperationspartner vornimmt.

Für unsere Untersuchungen ist der Einsatz von Agenten vor allem bei der Suche nach Dienst Anbietern bei der synchronen Kooperation und bei der Aushandlung von Verträgen sinnvoll, da die Agenten dabei nach Vorgabe bestimmter Ziele autonom handeln können. Solche Ziele wären im E-Commerce z. B. Preis- oder Terminvorgaben bei der Suche nach einem Flug in einem Reisebuchungs-Workflow. Im medizinischen Bereich könnte man Agenten u. a. in Workflows in der Logistik z. B. für die Suche nach dem günstigsten Anbieter für Verbrauchsgüter wie Spritzen oder Kanülen einsetzen. Denkbar wäre der Einsatz von Agenten auch bei der Suche nach einer Praxis oder Klinik für bestimmte Untersuchungen, die an einem Patienten in einem Workflow durchgeführt werden sollen und die nicht in der behandelnden Praxis oder Klinik durchgeführt werden können. Der Nutzer kann die Autonomie der Agenten natürlich auch beschränken, wenn er beispielsweise nicht möchte, dass sie Verträge abschließen ohne ihm vorher die Angebote vorgelegt zu haben.

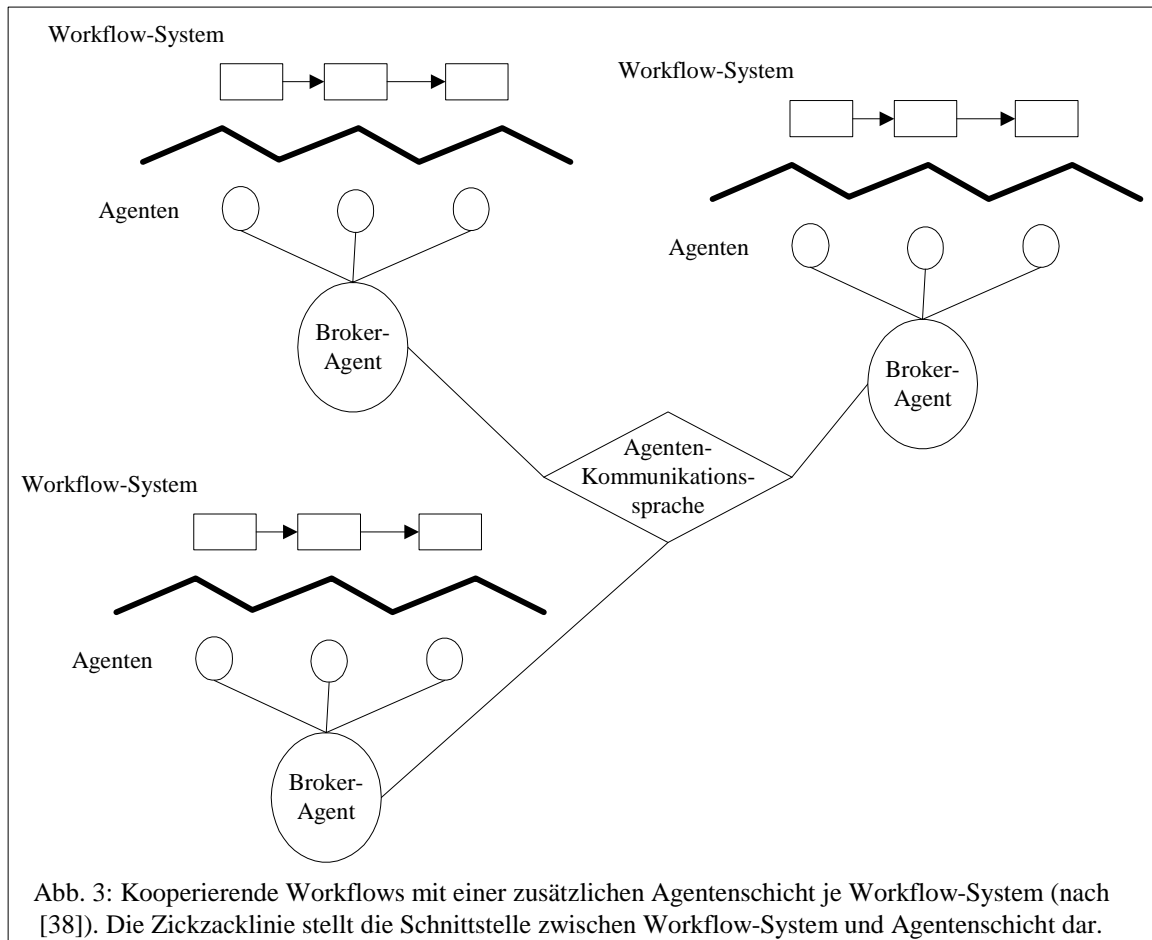


Abb. 3: Kooperierende Workflows mit einer zusätzlichen Agentenschicht je Workflow-System (nach [38]). Die Zickzacklinie stellt die Schnittstelle zwischen Workflow-System und Agentenschicht dar.

2.5 Kommunikationsebene

Bisher wurde nur die logische Ebene einer Workflow-Kooperation betrachtet. Im Folgenden werden einige wichtige Aspekte der technischen Umsetzung, die wir als Kommunikation bezeichnen, betrachtet.

Man kann grundsätzlich zwischen einer *symmetrischen* und einer *asymmetrischen* Kommunikation unterscheiden. Bei der symmetrischen Kommunikation gibt es gleichberechtigte Partner, die z. B. Nachrichten austauschen. Einer asynchronen Kooperation liegt also typischerweise eine symmetrische Kommunikation zugrunde. Bei der asymmetrischen Kommunikation liegt eine Client/Server-Konfiguration vor, in der der Client einen Dienst vom Server anfordert und wartet bis der Server ein Ergebnis zurückliefert. Eine synchrone Kooperation kann also über eine asymmetrische Kommunikation realisiert werden

Für eine erfolgreiche Kooperation spielt auch der Kommunikationsmechanismus eine entscheidende Rolle. Dabei können zwei Varianten unterschieden werden: Entweder wird eine Middleware eingesetzt oder es werden einfache Transportprotokolle wie HTTP (HyperText Transfer Protocol) verwendet, wenn nicht vorausgesetzt werden kann, dass alle Kooperationspartner über eine Middleware verfügen.

Beim Einsatz einer Middleware gibt es mehrere Möglichkeiten. Die erste ist der Einsatz einer sogenannten Message-Oriented-Middleware (MOM), die mit Warteschlangen arbeitet, in die die Kooperationspartner Nachrichten eintragen und aus denen sie Nachrichten lesen. Die War-

teschlinge kann so konzipiert werden, dass jede Nachricht garantiert zugestellt und nur einmal verarbeitet wird. Eine Alternative wäre die Verwendung einer objektorientierten Middleware wie CORBA (vgl. [35]). Das ist dann sinnvoll, wenn komplexe Objekte zwischen den kooperierenden Workflows ausgetauscht werden sollen. Sollen intelligente Agenten eingesetzt werden (etwa für die Aushandlung von Verträgen) so muss dafür ebenfalls eine passende Middleware verwendet werden.

Ein weiterer Aspekt der Kommunikation ist das Datenformat, in dem die Nachrichten zwischen den kooperierenden Workflows ausgetauscht werden. Diese Nachrichten können kurz sein, z. B. bei einem Dienstaufwurf, sie können aber auch komplexe Daten enthalten, die bei der Rückgabe von Ergebnissen oder dem Versenden von Ereignissen erforderlich sind. Für solche komplexen Nachrichten kämen beispielsweise HTML- oder XML-Dokumente (HyperText Markup Language bzw. eXtensible Markup Language, vgl. [4]) in Betracht. Deren Verwendung hat den Vorteil, dass im Gegensatz zu EDI (Electronic Data Interchange) (vgl. [42]) keine teure Infrastruktur zur Verarbeitung nötig ist, so dass sich auch kleine und mittlere Unternehmen und Privatkunden ohne großen finanziellen und technischen Aufwand an der Kooperation beteiligen können. Allerdings müssen die beschreibenden Schemata der verwendeten XML-Dokumente allen Beteiligten bekannt sein. Da es für XML-Schemata im Gegensatz zu EDI diesbezüglich (noch) keine Standards gibt, müssen sich die Kooperationspartner vorher über die verwendeten Dokumente verständigen. Für die Zukunft wäre es daher sinnvoll, wenn für jedes Anwendungsgebiet, in dem Workflow-Kooperation auftreten kann (z. B. Handel, Medizin, Verwaltung), eine Menge von grundlegenden Schemata vereinbart werden würde, die öffentlich bekannt sind und die mehrere Hersteller unterstützen. Damit wäre eine Basiskommunikation gesichert. Alle zusätzlichen Dokumente, die für eine bestimmte Kooperation benötigt werden, können die Geschäftspartner individuell vereinbaren, indem sie z. B. bestehende Schemata erweitern oder neue hinzufügen. Ein erster Ansatz dazu ist das Biztalk-Framework (www.biztalk.org), in dem sich Hersteller wie Microsoft, IBM und SAP zusammengetan haben, um gemeinsame Schemata zu vereinbaren, die allen Kunden zur Verfügung stehen.

3 Bewertungskriterien

In diesem Abschnitt sollen eine Reihe von Bewertungskriterien in vier Kategorien definiert werden, nach denen anschließend die verschiedenen Forschungsansätze für die Kooperation von Workflows verglichen werden können. Die Kategorien umfassen den Workflow-Ansatz, die Kooperations- und Kommunikationsebene, die Fehlerbehandlung und das Anwendungsgebiet, für das der Ansatz entwickelt wurde.

3.1 Workflow-Ansatz

Hierunter fallen das verwendete *Workflow-Modell* und das *Workflow-Management-System*. Übliche Workflow-Modelle sind Petri-Netze, graphbasierte Modelle, State- und Activity-Charts oder auch proprietäre Formate von Workflow-Definitions-Tools.

Als Workflow-System kann entweder ein kommerzielles Workflow-System verwendet werden oder es werden kommerzielle Teilkomponenten etwa für die Workflow-Definition oder -Ausführung verwendet und durch Eigenentwicklungen ergänzt. Alternativ dazu kann auch das gesamte Workflow-System selbstentwickelt werden. Zudem ist zu untersuchen, ob der Ansatz an ein bestimmtes System gebunden oder ob er auf beliebige Workflow-Systeme aufgesetzt werden kann.

3.2 Kooperations- und Kommunikationsebene

Hier ist zu untersuchen welche der im Abschnitt 2 vorgestellten *Kooperationsformen* (synchrone oder asynchrone Kooperation, vgl. Abschnitt 2.1 und 2.2), welche *Kommunikationsform* (symmetrisch oder asymmetrisch) und welcher Kommunikationsmechanismus jeweils gewählt wurden und welche Datenformate unterstützt werden (vgl. Abschnitt 2.5).

Außerdem ist zu berücksichtigen welche Arten von *Kooperationsbedingungen* (vgl. Abschnitt 2.3) formuliert werden können. Sind nur temporale Bedingungen möglich oder können auch Qualitätsrestriktionen für die auszutauschenden Daten angegeben werden?

3.3 Fehlerbehandlung

Wesentlich für die Fehlerbehandlung sind die Möglichkeiten, die die lokalen Workflow-Systeme bieten. Hierfür gibt es mehrere Alternativen, die im Folgenden kurz skizziert werden.

Ist schon zur Definitionszeit bekannt, dass Ausnahmen auftreten können, ist die Angabe *alternativer Pfade* oder eine *Ausnahmebehandlung* in der Art höherer Programmiersprachen wie Java oder C++ möglich, damit zur Laufzeit möglichst automatisch und ohne signifikante Verzögerung auf Ausnahmen reagiert werden kann.

Spätes Binden ermöglicht es, in der Workflow-Definition Platzhalter an Stelle von Aktivitäten zu verwenden, die dann erst zur Laufzeit durch passende Aktivitäten oder Subworkflows ersetzt werden. Dieses Vorgehen bietet sich an, wenn erst zur Laufzeit bekannt ist, welche Aktivitäten auszuführen sind, und eine Modellierung zahlreicher konditionaler Pfade vermieden werden soll.

Reichen Ausnahmebehandlung und Auswahl von Aktivitäten zur Laufzeit zur Behandlung eines Fehlers nicht aus, so ist ein *dynamischer Umbau* der betroffenen Workflow-Instanz hilfreich. Dabei werden z. B. Aktivitäten gelöscht oder neu eingefügt. Dieser Umbau kann manuell durch einen Nutzer oder teilautomatisiert erfolgen, wobei der Nutzer dann nur noch eine abschließende Überarbeitung und Bestätigung der Änderungen durchführt.

Für die Fehlerbehandlung ist auch Transaktionsunterstützung wichtig, um beispielsweise Aktivitäten geordnet zurückzusetzen oder auch zu kompensieren, falls ein Rücksetzen nicht möglich ist (vgl. [15, 16, 1, 26]). Beim Vergleich der Forschungsansätze soll deshalb auch berücksichtigt werden, ob beispielsweise das Anhalten einer Workflow-Instanz und die Fortsetzung der Ausführung an der gleichen Stelle bzw. nach minimaler Rücksetzung (Forward Recovery, vgl. [9, 45]) oder eine Kompensation von Aktivitäten möglich ist.

Wird eine Workflow-Instanz durch Mechanismen der lokalen Fehlerbehandlung verändert, können auch kooperierende Workflows dadurch beeinflusst werden, wenn z. B. temporale Kooperationsbedingungen nicht eingehalten werden können, weil sich die Ausführung durch neu eingefügte Knoten verzögert. Es ist auch möglich, dass Daten nicht mehr in der vereinbarten Qualität erzeugt werden können, weil z. B. Knoten, die bestimmte Teile der Daten erzeugen, nach den lokalen Adaptationen nicht mehr ausgeführt werden. Bei einer asynchronen Kooperation (mit symmetrischer Kommunikation) kann es durch lokale Adaptationen bei der weiteren Ausführung des veränderten Workflows zu Deadlocks kommen, weil eine Aktivität gelöscht wurde, auf deren Ereignis eine Aktivität in einem kooperierenden Workflow wartet.

Ziel einer Workflow-übergreifenden Fehlerbehandlung ist es, schon bei Durchführung der lokalen Änderung zu überprüfen, ob sich während der weiteren Ausführung des Workflows Implikationen für zukünftige Kooperations Schritte ergeben (Verletzung von Kooperationsbedingungen, Deadlocks). Ist das der Fall, kann der kooperierende Workflow frühzeitig benachrichtigt werden, nicht erst dann, wenn ein betroffener Kooperationsschritt erreicht oder der Deadlock eingetreten ist. Somit hat der Kooperationspartner die Möglichkeit, die

veränderten Kooperationsbedingungen seinerseits zu berücksichtigen, indem z. B. Zeitlimits angepasst, Qualitätsanforderungen vermindert oder Schritte zur Vermeidung des Deadlocks eingeleitet werden.

3.4 Anwendungsgebiet

Der letzte Vergleichspunkt ist das Anwendungsgebiet, für das der Forschungsansatz entwickelt wurde. Das Hauptanwendungsgebiet für Workflow-Systeme und damit für kooperierende Workflows ist im Augenblick der E-Commerce. Aber auch im medizinischen Bereich ergeben sich Möglichkeiten für den Einsatz von Workflow-Systemen. Gerade dort können komplexe Abhängigkeiten zwischen Workflows auftreten und auch Abweichungen von der ursprünglichen Workflow-Definition sind häufig zu erwarten. Ein Beispiel ist die Durchführung von Therapiestudien zu Chemotherapien oder Radiotherapien in der Hämato-Onkologie. Hierbei arbeiten sehr viele Institutionen wie die behandelnde Station, das Labor oder das Studiensekretariat zur Koordination und Durchführung der Studien zusammen. Workflows, die dort zur Ausführung der Prozesse eingesetzt werden, müssen also häufig kooperieren.

Komplexe Ausnahmen treten beispielsweise dann auf, wenn ein Patient während der Durchführung der Chemotherapie eine Infektion bekommt, die mit zusätzlichen Medikamenten oder einem Verzögern der eigentlichen Therapie behandelt werden muss. Die entsprechende Workflow-Instanz muss also zur Laufzeit verändert werden. Das kann sich wiederum auf kooperierende Workflows in der Studienzentrale oder im Labor auswirken, da Meldungen aufgrund der Verzögerung nicht rechtzeitig erstellt oder eine Laboruntersuchung wegen zusätzlich verabreichter Medikamente nicht wie ursprünglich geplant durchgeführt werden kann, weil das Ergebnis verfälscht werden würde.

4 Kooperationsunterstützung in kommerziellen Systemen

In diesem Abschnitt wird an zwei Beispielen kurz untersucht, in welcher Form kommerzielle Systeme Unterstützung für kooperierende Workflows bieten.

IBM MQSeries Workflow (vgl. [24]) bietet keine Unterstützung für die Kooperation mehrerer Workflows. Die notwendigen Kooperationsmechanismen und -schritte müssen in den Anwendungen, die die Aktivitäten des Workflows durchführen, ausprogrammiert werden. Auch eine lokale dynamische Fehlerbehandlung zur Laufzeit wird nur unzureichend unterstützt.

Microsoft BizTalk Server (vgl. [31]) wurde für die Unterstützung des Dokumentenaustauschs in einem Geschäftsprozess innerhalb eines Unternehmens und zwischen mehreren Unternehmen entwickelt. Deshalb ist es möglich, Dokumente innerhalb eines Workflows an externe Organisationen zu schicken und Ergebnisdokumente von externen Organisationen anzufordern. Somit wird eine asynchrone Kooperation zwischen Workflows in verschiedenen Unternehmen ermöglicht. Allerdings müssen die Workflow-Systeme der Unternehmen homogen sein, d. h. alle Beteiligten müssen BizTalk Server als Workflow-System verwenden. Für das Verschicken der Dokumente können auch Auslieferungsgarantien vereinbart werden, die eine Form von Kooperationsbedingungen darstellen. Die ausgetauschten Dokumente sind XML-Dokumente, deren Schemata der Nutzer definieren kann. Für eine effiziente Kooperation sollten die XMLSchemata aber dem Sender und dem Empfänger bekannt sein. Microsoft BizTalk Server kann also für eine asynchrone, dokumentenbasierte Kooperation in homogenen Umgebungen eingesetzt werden.

5 Synchroner Kooperation über Dienste

Für die synchrone Kooperation über Dienste gibt es verschiedene Forschungsansätze. Vier von diesen sollen im Folgenden vorgestellt und anhand der im Abschnitt 3 entwickelten Kriterien verglichen werden. Es handelt sich um das Projekt ANAISOF (Universität Zürich, vgl. [37]), die COLLABORATION MANAGEMENT INFRASTRUCTURE (MCC vgl. [13, 14]), das CROSSFLOW-Projekt (u. a. IBM Zürich, GMD IPSI, Universität Twente, vgl. [17, 30]) und das WISE-Projekt (ETH Zürich, vgl. [27]).

5.1 ANAISOF

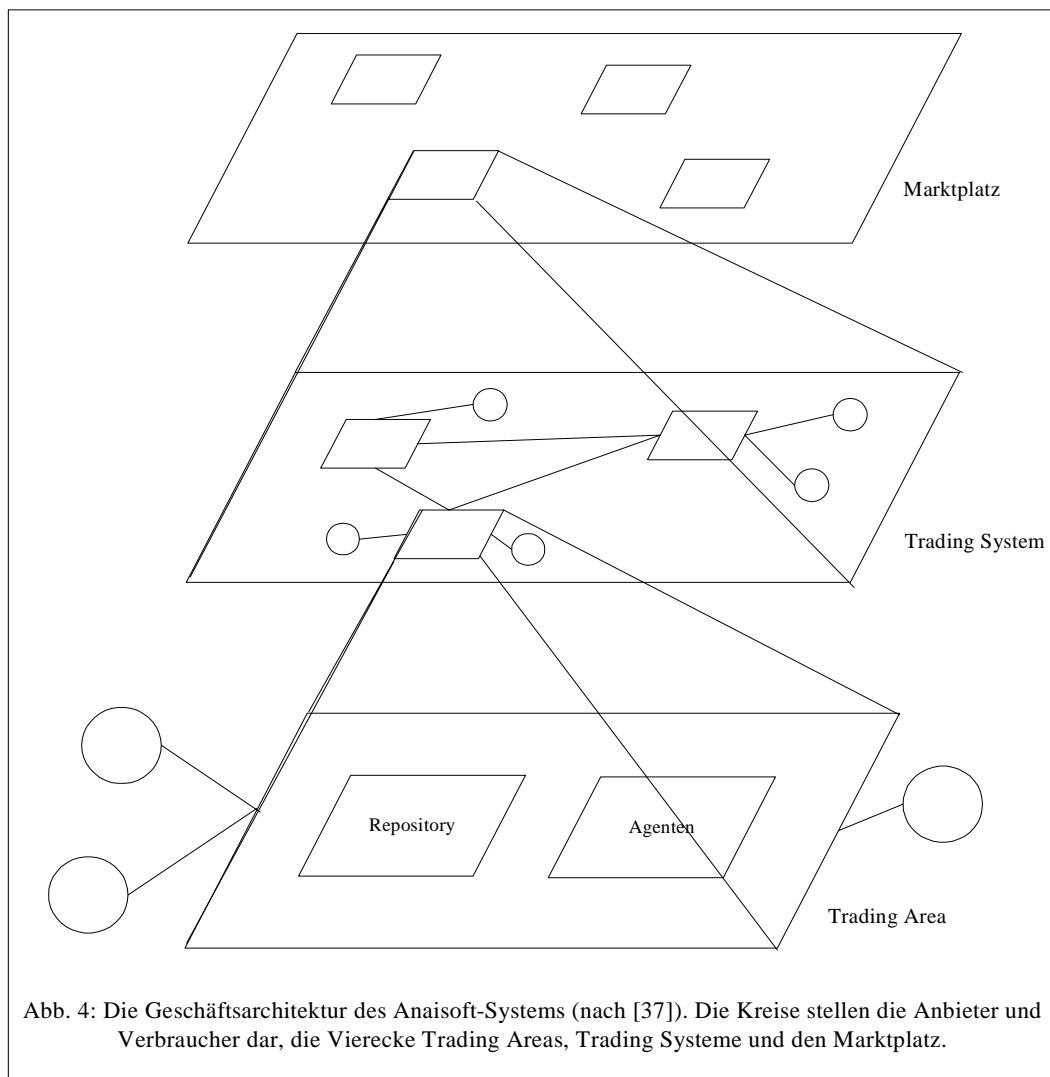
Im ANAISOF-Projekt (vgl. [37]) wird ein elektronischer Marktplatz entwickelt, auf dem Dienste in Form von Workflows über das Internet gehandelt werden können. Die beteiligten Unternehmen bieten Dienste an, die sie für andere Unternehmen ausführen können, und kaufen Dienste, die sie nicht selbst ausführen können aber zur Durchführung ihrer Anwendungsprozesse benötigen, bei anderen Unternehmen. Es entsteht also eine synchrone Kooperation zwischen den Workflows. Das Anwendungsgebiet ist der B2B-Bereich des E-Commerce.

Es wird dabei eine dreischichtige Geschäftsarchitektur verwendet (vgl. Abbildung 4): Die Klienten (Anbieter und Verbraucher) verbinden sich mit sogenannten *Trading Areas* (TA), die beispielsweise alle Klienten einer bestimmten Region zusammenfassen und alle Workflow-Definitionen, Anfragen und Gebote der beteiligten Unternehmen in einem Repository verwalten. Das Workflow-System eines Klienten kann nur an eine TA angeschlossen werden, wenn sein Workflow-Modell auf das im Repository der TA verwendete Modell abgebildet werden kann. Mehrere TAs können sich dann zu einem *Trading System* (TS) zusammenschließen und die Inhalte ihrer Repositories teilen. Ein Klient, der über eine bestimmte TA an das TS angeschlossen ist, sieht auch alle Workflow-Definitionen, Anfragen und Gebote der Unternehmen der anderen am TS beteiligten TAs. Der elektronische *Marktplatz* vereint dann verschiedene TS, in denen die Klienten handeln können. Dabei werden aber nicht automatisch die Repository-Inhalte einer TS für die übrigen im Marktplatz vertretenen TSs sichtbar. Um Workflow-Definitionen in und aus mehreren TSs zu handeln, müssen die Klienten sich mit mindestens einer TA je TS verbinden, an deren Workflow-Definitionen sie interessiert sind.

Als Datenaustauschformat zwischen den TAs eines TS werden XML-Dokumente verwendet. Das Workflow-Modell der einzelnen TAs muss also auf das gemeinsame Datenformat abbildbar sein, ansonsten kann eine TA sich nicht einer Föderation (also einem TS) anschließen. So wird die Definition eines globalen Workflow-Modells vermieden und die Klienten können ihr bisher verwendetes Modell beibehalten.

Der Handel von Workflows erfolgt über mobile Agenten, die z. B. für Angebot und Nachfrage, Auswahl eines Anbieters und die tatsächliche Ausführung des Workflows beim Anbieter verantwortlich sind. Verhandlungen über die Angebote und Kooperationsbedingungen sind nicht vorgesehen, es kann nur das beste Angebot ausgewählt werden.

Ändert ein Anbieter eine Workflow-Definition, die in laufenden oder bereits geplanten aber noch nicht gestarteten Workflow-Instanzen von Verbrauchern verwendet wird, so kann das zu Fehlern beim Verbraucher führen, wenn die alte Version nicht mehr verfügbar ist. Um das zu verhindern, wird eine erweiterte Versionsverwaltung eingesetzt (vgl. [39]), so dass gleichzeitig mehrere Versionen einer Workflow-Definition existieren. Ein Dienstanutzer muss seinen Workflow, in dem die Verwendung einer Dienstinanz nach der alten Definition vorgesehen ist, also nicht ändern, wenn die Definition durch den Anbieter verändert wird. Weitere Implikationen lokaler Änderungen wie die Verletzung von temporalen oder inhaltlichen Kooperationsbedingungen werden aber nicht berücksichtigt.



5.2 Collaboration Management Infrastructure

Die COLLABORATION MANAGEMENT INFRASTRUCTURE (CMI) (vgl. [13, 14]) ermöglicht die Realisierung der Kooperation zwischen mehreren Unternehmen auf der Prozessebene. Dazu werden sogenannte Multi-Enterprise-Prozesse (MEP) definiert, die aus einer Kombination von Single-Enterprise-Prozessen (SEP) bestehen. Die SEPs entsprechen Diensten, die die an dem entstehenden virtuellen Unternehmen beteiligten Partner zur Verfügung stellen. Es entsteht eine synchrone Kooperation zwischen dem MEP und den Unternehmen, die die SEPs ausführen, da der MEP wartet bis ein SEP vollständig ausgeführt ist, bevor die nächste Aktivität und damit der nächste SEP aktiviert wird. Das Anwendungsgebiet ist auch hier vorrangig der B2B-Bereich des E-Commerce.

Das *Collaboration Management Model* (CMM) besteht aus einem Kernmodell CORE und den Erweiterungsmodellen *Coordination Model* (CM), *Service Model* (SM), *Awareness Model* (AM) sowie anwendungsspezifischen Erweiterungen. CORE enthält Prozessmodellprimitive für Aktivitätszustände, Ressourcen und Rollendefinitionen. Das Coordination Model ermöglicht die Definition von Aktivitäten, Daten- und Kontrollfluss, das Service Model die Definition von Diensten und das Awareness Model die Überwachung von Ereignissen. Das CMM kann auf verschiedene Workflow-Systeme aufgesetzt werden, indem seine Elemente entweder auf

Primitive des Workflow-Systems abgebildet oder durch Prozesse des Workflow-Systems implementiert werden.

Die Semantik eines Dienstes wird in der *Dienstschnittstelle* in Form von Operationen, Zuständen und Ein- und Ausgabeparametern beschrieben. Zusätzliche Kooperationsbedingungen sind darin nicht vorgesehen. Die Dienstschnittstelle wird in *Dienstaktivitäten* implementiert, über die im Kontrollfluss des MEPs der Aufruf des eigentlichen Dienstes erfolgt. Für die Koordination von Diensten untereinander und die Synchronisation von Diensten mit dem restlichen Prozess werden erweiterte Kontrollfluss-Transitionen zur Verfügung gestellt, die auf die in der Dienstschnittstelle definierten Operationen und Zustände zugreifen können.

Ist bei der Definition des MEPs noch nicht bekannt, welche Aktivität oder welcher Dienst zur Laufzeit an einer bestimmten Stelle ausgeführt werden soll, kann ein Platzhalter für diese Aktivität verwendet werden, für den dann zur Laufzeit eine passende Aktivität ausgewählt wird. Ist auch unbekannt, wie oft die Aktivität durchgeführt werden soll, können sogenannte *windows of opportunity* angegeben werden, während deren Gültigkeit mehrere Instanzen der Aktivität gestartet werden können. Lokale Fehlerbehandlungen zur Laufzeit sind also nur über spätes Binden möglich und können nur dort den Workflow verändern, wo bereits ein Platzhalter vorgesehen ist. Die Auswirkungen solcher Änderungen auf kooperierende Workflows, die z. B. auf Ergebnisse warten, werden nicht berücksichtigt.

5.3 CrossFlow

Im CROSSFLOW-Projekt (vgl. [17, 30]) werden Dienste zur synchronen Kooperation zwischen Unternehmen verwendet, die Dienstleistungen in Form von Workflows anbieten und in Anspruch nehmen. Der Schwerpunkt liegt hier neben dem Einsatz von interagierenden Workflow-Systemen auf der Definition und Verwendung von Verträgen zur Spezifikation der angebotenen und in Anspruch genommenen Dienste. Das Anwendungsgebiet ist der B2B-Bereich des E-Commerce.

Die Ankündigung von Diensten und die Suche nach benötigten Diensten erfolgt über einen sogenannten *Dienstvermittler*, der vom Anbieter und vom Nutzer Vertragsvorlagen erhält, in denen der angebotene bzw. gesuchte Dienst beschrieben wird. Der Dienstvermittler schickt dem Dienstanbieter die Vertragsvorlagen der Anbieter, die den gewünschten Dienst anbieten. Der Nutzer wählt einen von diesen aus und erzeugt einen Vertrag, indem er der Dienstbeschreibung die von ihm gewünschten temporalen und inhaltlichen Kooperationsbedingungen in Form von Quality-of-Service-(QoS)Parametern hinzufügt. Der Vertrag wird dann an den Dienstanbieter geschickt, der ihn akzeptieren oder ablehnen kann.

Zur Ausführung des vereinbarten Dienstes wird eine Verbindung zwischen den Partnern aufgebaut, über die der Verbraucher die Ausführung des Dienstes initiiert und überwacht. Die Kommunikation erfolgt über Gateways, die dem jeweiligen Partner eine Schnittstelle zu den für ihn nach dem Vertrag zugänglichen Funktionen (beispielsweise für den Start oder die Unterbrechung eines Dienst-Workflows) bieten.

Es gibt zusätzlich eine Reihe von Dienstprogrammen, die u. a. die Einhaltung der Kooperationsvereinbarungen überwachen und die Workflow-Instanz dynamisch verändern, wenn Vereinbarungen beispielsweise aufgrund einer Verzögerung nicht eingehalten werden können (vgl. [23]). Die dynamische Veränderung besteht darin, aus einer Menge alternativer Pfade die auszuwählen, bei deren Ausführung die Vereinbarungen eingehalten werden können. Die alternativen Pfade sind Teil des Vertrages zwischen Dienstanbieter und Dienstnehmer.

Außerdem ermöglichen diese Dienstprogramme es dem Dienstanbieter, auch nach dem Start des Dienst-Workflows noch Einfluss auf dessen Ausführung zu nehmen. Über Funktionen wie *rollback*, *stop*, *continue*, *abort* oder *change case variable* kann der Zustand des Dienst-Workflows beeinflusst werden. Damit ist eine Änderung des Zustands des Dienst-Workflows im entfernten

Workflow-System des Partners beispielsweise nach einem Abbruch des Verbraucher-Workflows möglich.

[29] beschreibt eine Architektur für inter-organisatorische Workflows, die auch die durch ein unerwartetes Ende des Dienst-Workflows (zum Beispiel aufgrund einer ungültigen Lieferadresse beim Versandhandel) zusätzlich entstandenen Kosten berücksichtigt. Andere Abhängigkeiten, wie z. B. die Verletzung temporaler Kooperationsvereinbarungen durch das unerwartete Ende eines kooperierenden Workflows, werden aber nicht berücksichtigt.

5.4 WISE

Der Schwerpunkt von WISE (Workflow based Internet Services, vgl. [27]) liegt auf der Definition und Ausführung virtueller Geschäftsprozesse in virtuellen Unternehmen. Die Kooperation zwischen dem virtuellen und den realen Unternehmen ist synchron, da immer auf das Ende eines realen Workflows gewartet wird, bevor die nächste Aktivität im Workflow des virtuellen Unternehmens ausgeführt wird. Das Anwendungsgebiet ist auch hier wieder der B2B-E-Commerce.

Mittels einer ProzessdefinitionsKomponente wird aus den in einem Web-Katalog angebotenen Diensten der Workflow für den virtuellen Geschäftsprozess zusammengesetzt. Die Workflow-Definition wird in das Format der WISE-Engine übersetzt, die die Ausführung des Prozesses steuert und für jeden Dienst das lokale Workflow-System des realen Unternehmens aufruft, das dann den der Aktivität entsprechenden Workflow ausführt. Die lokalen Workflow-Systeme müssen die Atomarität der Aktivitätsausführung und eine erfolgreiche Durchführung von Kompensationen garantieren sowie von der WISE-Engine festgelegte Zugriffsreihenfolgen auf gemeinsame Datenquellen akzeptieren. Da nicht alle Workflow-Systeme diese Anforderungen erfüllen, wird jedes lokale Workflow-System mit einem *Program Execution Client* (PEC) gewrappt, der die Aktivitätsausführung im Auftrag der WISE-Engine inklusive dem Scheduling und der Überwachung der Kooperationsvereinbarungen (in Form von Garantien für die Ausführungsdauer oder die Qualität der Ergebnisse) übernimmt (vgl. [28]).

Treten während der Ausführung des virtuellen Workflows in externen Programmen oder innerhalb von WISE Fehler auf, wird eine Ausnahmebehandlung über Ausnahmeaktivitäten oder Subworkflows durchgeführt, die bei der Definition des Workflows festgelegt und einem Fehlertyp zugeordnet werden. Für alle so nicht erfassten Fehler könnte zum Beispiel eine Default-Ausnahmeaktivität definiert werden, um eine Behandlung aller vorkommenden Fehler zu garantieren.

Der Nutzer kann bei der Definition des virtuellen Workflows Atomaritäts-, Isolations- und Persistenzsphären bestimmen, die dann vom System auf ihre Plausibilität überprüft werden (z. B. hängt die Garantie der Atomarität eines Workflow-Abschnitts davon ab, ob die darin aufgerufenen Anwendungen diese garantieren können). Diese Sphären werden während der Ausführung des Workflows verwendet, um beispielsweise Fehlerfälle wie das Rücksetzen eines Dienstes oder Konfliktsituationen bei der parallelen Ausführung von Diensten (beispielsweise beim parallelen Zugriff auf gemeinsam genutzte Ressourcen) adäquat zu behandeln und Abhängigkeiten zwischen kooperierenden Workflows bei der lokalen Fehlerbehandlung zu berücksichtigen.

5.5 Zusammenfassender Vergleich

Tabelle 2 gibt einen Überblick über die vier in diesem Abschnitt vorgestellten Ansätze zur synchronen Kooperation und ihre Eigenschaften. Alle Ansätze wurden für den B2B-Bereich des E-Commerce entwickelt. Sie verwenden unterschiedliche Workflow-Modelle und -Systeme, die teils proprietär sind wie das Collaboration Management Model bei CMI oder die WISE-Engine.

Es werden aber auch kommerzielle Produkte wie IBM MQSeries Workflow (früher Flowmark) oder das Structware-Definitions-Tool eingesetzt.

Die Kooperation erfolgt bei allen synchron unter der Verwendung von Diensten. Es gibt bei allen Systemen eine Art der Vereinbarung (z. B. Verträge bei CROSSFLOW), dass der Dienst für die andere Organisation durchgeführt wird, aber explizite Kooperationsbedingungen werden nur von CROSSFLOW durch QoS-Parameter und von WISE durch Garantien für die Ausführungsdauer oder die Qualität unterstützt.

Alle vier Ansätze ermöglichen lokale dynamische Fehlerbehandlungen. ANAISOFT verwendet dazu eine erweiterte Versionsverwaltung, CMI spätes Binden, WISE eine Mechanismus zur Ausnahmebehandlung. In CROSSFLOW wird eine Workflow-Instanz bei Bedarf durch die Auswahl von Ausführungsalternativen verändert.

Workflow-übergreifende Fehlerbehandlungen werden außer von CMI (keine Aussage gefunden) von allen Ansätzen zumindest teilweise unterstützt. Die erweiterte Versionsverwaltung von ANAISOFT stellt sicher, dass kooperierende Workflows immer die für sie gültige Version des Dienst-Workflows erhalten. Weitere Benachrichtigungen z. B. über Verzögerungen erfolgen nicht.

In CROSSFLOW werden Verletzungen von Kooperationsvereinbarungen durch die Auswahl eines passenden Pfades aus einer Menge von Alternativen lokal behandelt, so dass der Dienstanutzer nicht benachrichtigt werden muss. Darüberhinaus ist über bestimmte Operationen eine Änderung des Zustands des Dienst-Workflows durch den Dienstanutzer möglich, wenn beispielsweise der Verbraucher-Workflow abgebrochen wird.

WISE bietet über die Definition von Atomaritäts-, Isolations- und Persistenzsphären eine Möglichkeit zur Koordination kooperierender Workflows auch im Fehlerfall. Ein Eingriff in die Workflows erfolgt dabei nur über die Programm Execution Clients nicht direkt durch die WISE-Engine.

	ANAISOFT (vgl. [37])	CMI (vgl. [13, 14])	CrossFlow (vgl. [17, 30])	WISE (vgl. [27])
Forschungsgruppe	Universität Zürich (Dittrich)	MCC (Georgakopoulos)	u. a. IBM Zürich (Ludwig)	ETH Zürich (Alonso)
Workflow-Modell	diverse, gemeinsames Nachrichtenformat	Collaboration Management Model	graphbasiert	Petri-Netze (Structware- Modell)
Workflow-System	diverse	u. a. IBM Flow- mark	IBM MQSeries Workflow	WISE-Engine (basierend auf OPERA-Engine)
Kooperationsform	synchron über Dienste	synchron über Dienste	synchron über Dienste	synchron über Dienste
Kooperationsbedingungen	-	-	temporal und inhaltlich	temporal und inhaltlich
Kommunikationsform / -mechanismus	asymmetrisch / Agenten mit passender Middleware	asymmetrisch / -	asymmetrisch / Gateways mit Java RMI	asymmetrisch / -
Datenaustauschformat	XML-Dokumente	festgelegt in Dienstschnittstelle	XML-basierte Verträge	nicht genannt
lokale dynamische Fehlerbehandlung	Versionsverwaltung bei Änderung der Workflow-Definitionen	spätes Binden	dynamische Auswahl eines Pfades aus einer Menge von Alternativen	Ausnahmebehandlung
Workflow-übergreifende Fehlerbehandlung		-	rollback, stop, continue, abort, change case variable	Atomaritäts-, Isolations-, Persistenzsphären
Anwendungsgebiet	E-Commerce	E-Commerce	E-Commerce	E-Commerce

Tabelle 2: Forschungsansätze zur synchronen Workflow-Kooperation über Dienste.

6 Asynchrone Kooperation über Ereignisse

Die zweite in diesem Artikel betrachtete Kooperationsform ist die asynchrone Kooperation über Ereignisse. Auch hierzu gibt es verschiedene Forschungsansätze, von denen hier wieder vier vorgestellt und anhand der Kriterien aus Abschnitt 3 verglichen werden. Die vier Ansätze sind OPERA (ETH Zürich, vgl. [19]), PANTA RHEI (Universität Klagenfurt, vgl. [18]), WIDE (u. a. Politecnico die Milano, Universität Twente, vgl. [6]) und AGENTWORK (Universität Leipzig, vgl. [34]).

6.1 Opera

Im OPERA-Kernel, auf dem die im Abschnitt 5.4 beschriebene WISE-Engine aufsetzt, erfolgt die Kooperation zwischen Workflows asynchron über Ereignisse (vgl. [19]). Der Ereignis-Manager ist Teil der Workflow-Engine. Das Anwendungsgebiet ist der E-Commerce, wobei der Ansatz ziemlich allgemein formuliert ist, so dass er auch in anderen Gebieten eingesetzt werden könnte. Kooperationsbedingungen können nicht angegeben werden.

OPERA unterstützt spätes Binden (vgl. Abschnitt 3.3) für eine lokale dynamische Ausnahmebehandlung. Dadurch können Subworkflows zur Laufzeit ausgetauscht werden, wenn der ursprünglich vorgesehene Subworkflow nicht mehr passend ist.

Wird bei einer lokalen Fehlerbehandlungen eine Aktivität abgebrochen, zurückgesetzt oder kompensiert, die mindestens ein Ereignis erzeugt hat, hat das möglicherweise Implikationen für kooperierende Workflows, die das Ereignis verarbeitet haben. Diese Implikationen werden folgendermaßen berücksichtigt: Alle Aktivitäten im gleichen und in kooperierenden Workflows, die dieses Ereignis verarbeitet haben, werden benachrichtigt. Die betroffenen Aktivitäten werden über eine persistente Tabelle ermittelt, in der OPERA für alle Ereignisse die Erzeuger-Verbraucher-Beziehungen speichert. Für diese Aktivitäten wird dann eine bei der Workflow-Definition festgelegte *Recovery-Prozedur* aufgerufen. Diese besteht aus zwei Teilen:

- einer *Ausnahmebehandlung*, um zum Beispiel Kompensationsschritte auszuführen, und
- einem sogenannten *Terminationsprotokoll*, dessen Verhalten durch den *Recovery-Modus* des Ereignisses festgelegt wird.

Der Recovery-Modus wird bei der Workflow-Definition für jedes Ereignis festgelegt, wobei folgende Möglichkeiten zur Auswahl stehen:

- IGNORE: Hier werden keine weiteren Aktionen durchgeführt, da ein Abbruch des betroffenen Workflows entweder zu teuer oder unmöglich ist oder aus der Anwendungssemantik klar ist, dass keine Recovery nötig ist und beispielsweise Kompensationen, die bei der Ausnahmebehandlung durchgeführt werden, ausreichend sind.
- ABORT: Der Workflow, in dem die Verbraucheraktivität liegt, wird abgebrochen, weil eine weitere Ausführung nach dem Ereignisrückruf nicht sinnvoll ist.
- PARTIAL ROLLBACK: Es werden alle Aktivitäten zurückgesetzt oder kompensiert, deren Ausführung kausal von dem zurückgerufenen Ereignis abhing. Danach wird die Ausführung der betroffenen Workflows am letzten konsistenten Zustand vor der Generierung des zurückgerufenen Ereignisses fortgesetzt.

Durch diese Recovery-Prozedur können fünf verschiedene *Recovery-Arten* realisiert werden, falls ein Ereignis wegen Abbruchs der erzeugenden Aktivität zurückgerufen werden muss. Im Einzelnen sind das:

- IGNORE: Das fehlerhafte Ereignis wird ignoriert.
- NO_ABORT: Durch die Ausnahmebehandlung werden alternative Aktivitäten durchgeführt.
- ABORT: Der betroffene Workflow wird abgebrochen.
- CONTINGENCY: Es werden alternative Aktivitäten anstelle der abgebrochenen Aktivität ausgeführt.
- ROLLBACK: Die Workflows werden in den Zustand vor der Generierung des Ereignisses zurückgesetzt und die Ausführung von dort fortgesetzt.

Mit diesem Mechanismus werden aber nur bereits generierte und verarbeitete Ereignisse berücksichtigt. Auswirkungen der lokalen Fehlerbehandlungen auf zukünftige Kooperationschritte (z. B. die verzögerte Generierung von Ereignissen) werden nicht betrachtet.

6.2 Panta Rhei

Im Workflow-System PANTA RHEI (vgl. [18]) ist die Kooperation zwischen Workflow-Systemen asynchron und basiert auf der *Formflow-Metapher*. Diese besagt, dass jegliche Kooperation über Formulare (z. B. HTML-Formulare bei der Kooperation über das Internet) erfolgt, d. h. der Eingang eines Formulars startet einen Workflow oder setzt ihn fort und das Versenden eines Formulars ist (Teil-)Ergebnis eines Workflows. Da das Versenden eines Formulars mit der Generierung eines Ereignisses und das Verarbeiten eines Formulars mit der Verarbeitung eines Ereignisses gleichgesetzt werden kann, kann dieses Vorgehen auch als asynchrone Kooperation über Ereignisse betrachtet werden. Kooperationsbedingungen sind nur als temporale Bedingungen in Form von Zeitlimits möglich.

Das Verschicken von Formularen wird über spezielle Aktivitäten realisiert, die die Formulare entweder an das Workflow-System oder an den E-Mail-Client des Empfängers (falls dieser kein Workflow-System hat) schicken. Die Formulare können HTML- oder XML-Dokumente sein und werden mittels HTTP-Protokoll verschickt. Das Hauptanwendungsgebiet ist der elektronische Handel, wobei hier auch der B2C-Bereich abgedeckt wird, da ein Kunde nicht unbedingt ein Workflow-System benötigt sondern nur einen Browser und ein E-Mail-Programm.

Die lokale Fehlerbehandlung wird durch die Auswahl alternativer Pfade oder durch Kompensationsaktivitäten ermöglicht (vgl. [10]). Die alternativen Pfade werden in der Workflow-Definition angegeben und zur Laufzeit dynamisch ausgewählt, wenn der ursprüngliche Pfad nicht erfolgreich ausgeführt werden kann. Die Kompensationsaktivität und der *Kompensationstyp* wird in der Aktivitäts-Definition festgelegt. Es werden vier Kompensationstypen unterschieden:

- NONE: Die Aktivität muss nicht kompensiert werden.
- UNDOABLE: Die Aktivität kann durch ihre Inverse vollständig kompensiert werden.
- COMPENSATABLE: Die Kompensation hat Nebeneffekte, z. B. die Berechnung einer Stornogebühr.
- CRITICAL: Die Aktivität kann nicht kompensiert werden (z. B. das Verschicken vertraulicher Informationen).

Kommt es durch lokale Fehlerbehandlungen zur Verletzung von Zeitlimits oder zur Nichtverfügbarkeit von Ressourcen, so kann das durch das Verschicken von Ausnahmeereignissen den kooperierenden Workflows angezeigt werden (vgl. [11]).

6.3 WIDE

In WIDE wird das ursprüngliche Workflow-Modell mit Ereignisknoten zum Versenden (*send-Knoten*) und Anfordern (*request-Knoten*) von Ereignissen erweitert, um eine asynchrone Kooperation mehrerer Workflows zu ermöglichen (vgl. [6]). Jedes beteiligte Workflow-System hat einen externen Ereignis-Manager, der die von seinen Workflows erzeugten Ereignisse verwaltet und Anfragen von Aktivitäten in lokalen Workflows und von Ereignis-Managern anderer Workflow-Systeme bearbeitet.

Das Anwendungsgebiet ist hier nicht festgelegt, WIDE wird sowohl in E-Commerce-Anwendungen als auch zur Unterstützung von Verwaltungsvorgängen in einem Krankenhaus einge-

setzt.

Ereignissen und Anfragen können Prioritäten zugewiesen werden (vgl. Abschnitt 2.2). Dadurch werden temporale Kooperationsbedingungen in Form von Restriktionen bzgl. der Auslieferung eines Ereignisses realisiert.

Die Kommunikation ist symmetrisch, es besteht also die Gefahr von Deadlocks. Diese können vor allem dann entstehen, wenn aufgrund lokaler Fehlerbehandlungen, die in WIDE durch ECA-Regeln realisiert werden (vgl. [8]), Ereignisse nicht mehr wie ursprünglich vorgesehen erzeugt werden können. Potentielle Deadlocks können mit Hilfe des integrierten Workflows erkannt werden, der entsteht, wenn alle kooperierenden Workflows in einem Gesamt-Workflow zusammengefasst werden. Werden bei der Analyse des integrierten Workflows potentielle Deadlocks erkannt, können z. B. alternative Pfade angegeben werden für den Fall, dass das benötigte Ereignis nach einer bestimmten Zeit noch nicht eingetroffen ist.

Können die Konstruktion und Analyse des integrierten Workflows auch zur Laufzeit durchgeführt werden, so bietet dieser Mechanismus eine Möglichkeit Implikationen lokaler Fehlerbehandlungen für kooperierende Workflows zu berücksichtigen.

6.4 AgentWork

Das Workflow-Modell des Projekts AGENTWORK (vgl. [34]) sieht Kommunikationsknoten vor, über die Objekte, die z. B. komplexe Daten enthalten, an kooperierende Workflows verschickt und von diesen empfangen werden können. Dies ermöglicht eine asynchrone Kooperation über Ereignisse, wobei der Ereignis-Manager Teil der Workflow-Engine ist, da diese für die Verteilung der Ereignisse an die entsprechenden Workflow-Instanzen sorgt. Das Projekt hat einen medizinischen Anwendungshintergrund, der Ansatz kann aber auch in anderen Anwendungsgebieten eingesetzt werden. Es sind sowohl temporale Kooperationsbedingungen in Form von Zeitlimits möglich, innerhalb derer ein Objekt verschickt werden muss oder erwartet wird, als auch inhaltliche Kooperationsbedingungen, z. B. über die Qualität eines Objekts. Die lokale Fehlerbehandlung erfolgt durch dynamische, regelbasierte Adaptationen des Kontroll- und Datenflusses. Dabei werden zum Beispiel Aktivitäten gelöscht, neu eingefügt oder durch andere Aktivitäten ersetzt. Direkt im Anschluss an eine lokale Fehlerbehandlung wird dann überprüft, ob noch alle Objekte wie vorgesehen, d. h. in der gewünschten Qualität und innerhalb der gewünschten Zeit, erzeugt werden und somit alle Kooperationsvereinbarungen im noch nicht ausgeführten Teil der Workflow-Instanz eingehalten werden können. Ist das nicht der Fall, wird das kooperierende Workflow-System benachrichtigt, welche Objekte betroffen sind und kann dann an den betroffenen Workflows entsprechende Änderungen vornehmen, wenn dafür Mechanismen zur Verfügung stehen.

AGENTWORK bietet also eine Workflow-übergreifende Fehlerbehandlung mit der Möglichkeit Implikationen für die Kooperation, die durch lokale Adaptationen entstehen, rechtzeitig zu entdecken und die kooperierenden Workflows zu informieren. Damit werden auch mögliche Deadlocks vermieden, die durch die symmetrische Kommunikation begünstigt werden.

	OPERA (vgl. [19])	PANTA RHEI (vgl. [18])	WIDE (vgl. [6])	AGENTWORK (vgl. [34])
Forschungsgruppe	ETH Zürich (Alonso)	Uni Klagenfurt (Eder)	u. a. Politecnico di Milano	Universität Leipzig
Workflow-Modell	OPERA graphical workflow language (OGWL)	Workflow Definition Language (WDL)	WIDE WPDL (Workflow Process Definition Language)	graphbasiert
Workflow-System	OPERA-Engine	Panta Rhei	WIDE-WfMS	AgentWork
Kooperationsform	asynchron über Ereignisse	asynchron über Ereignisse	asynchron über Ereignisse	asynchron über Ereignisse
Kooperationsbedingungen	-	temporal	temporal	temporal und inhaltlich
Kommunikationsform/-mechanismus	symmetrisch / Queues, Subscribe-Mechanismus	symmetrisch / HTTP	symmetrisch / CORBA	symmetrisch / CORBA
Datenaustauschformat	Ereignisse = (Name, Parameterliste)	HTML-/XML-Formulare	Ereignisname (Empfänger, Priorität, Parameter)	Objekte
lokale dynamische Fehlerbehandlung	spätes Binden	Definition alternativer Pfade, Kompensationsaktivitäten	regelbasiert (ECA-Mechanismus)	regelbasierte Adaptationen
Workflow-übergreifende Fehlerbehandlung	Ereignisrückruf mit Recovery-Prozedur	Ausnahmeereignisse	Analyse des integrierten Workflows	Benachrichtigung bei Verletzung von Kooperationsbedingungen oder Fehlen eines Objekts
Anwendungsgebiet	E-Commerce	E-Commerce	E-Commerce	Medizin

Tabelle 3: Forschungsansätze zur asynchronen Workflow-Kooperation über Ereignisse.

6.5 Zusammenfassender Vergleich

Tabelle 3 gibt nochmals einen Überblick über die in den vorhergehenden Abschnitten vorgestellten Ansätze. Diese verwenden alle proprietäre Workflow-Modelle und -Systeme. Außer AGENTWORK wurden sie alle für das Anwendungsgebiet des E-Commerce entwickelt, wobei PANTA RHEI auch im B2C-E-Commerce eingesetzt werden kann.

OPERA unterstützt keine Kooperationsbedingungen, bei PANTA RHEI und WIDE können temporale Kooperationsbedingungen angegeben werden. AGENTWORK unterstützt temporale und inhaltliche Kooperationsbedingungen.

Lokale dynamische Fehlerbehandlung wird von allen vier Systemen unterstützt, wobei sehr unterschiedliche Mechanismen verwendet werden. OPERA unterstützt spätes Binden. PANTA RHEI ermöglicht die Auswahl alternativer Pfade und die Verwendung von Kompensationsaktivitäten. WIDE und AGENTWORK verwenden eine regelbasierte Fehlerbehandlung, wobei AGENTWORK auch den dynamischen Umbau eines nicht mehr adäquaten Workflows ermöglicht.

Eine Workflow-übergreifende Fehlerbehandlung, d. h. die Berücksichtigung von Implikationen durch lokale dynamische Änderungen für kooperierende Workflows wird ebenfalls von allen Ansätzen unterstützt. Allerdings erfolgt bei OPERA und PANTA RHEI nur eine Berücksichtigung schon generierter Ereignisse. In WIDE erfolgt eine Analyse des integrierten Workflows zur Erkennung potentieller Deadlocks und in AGENTWORK wird nach lokalen Änderungen überprüft, ob noch alle Objekte erzeugt und alle Kooperationsbedingungen eingehalten werden können. Diese beiden Ansätze bieten also auch eine Berücksichtigung von Implikationen für zukünftige Kooperationschritte.

Da es sich bei allen in den Abschnitten 5 und 6 vorgestellten Systemen um Forschungsarbeiten handelt, kann in den meisten Fällen keine Aussage darüber gemacht werden, welche Teile davon tatsächlich implementiert sind und welche nur konzeptionell entwickelt wurden.

7 Zusammenfassung

In diesem Artikel wurde am Anfang motiviert, warum dynamisches Workflow-Management für kooperierende Workflows wichtig ist. Es wurden zwei grundlegende Kooperationsformen beschrieben, die synchrone und die asynchrone Kooperation, und der Begriff der Kooperationsbedingungen eingeführt, um temporale und inhaltliche Bedingungen für die Kooperation zu beschreiben. Anschließend wurden Bewertungskriterien für den Vergleich verschiedener Forschungsansätze zur Workflow-Kooperation aufgestellt. Diese umfassen neben dem Workflow-Modell und -System, der Kooperationsform und den Kooperationsbedingungen insbesondere die lokale und Workflow-übergreifende Fehlerbehandlung. Für die lokale Fehlerbehandlung wurden dabei folgende Mechanismen gefunden: Auswahl alternativer Pfade, Ausnahmebehandlung, spätes Binden zur Auswahl einer passenden Aktivität oder eines Subworkflows zur Laufzeit, dynamischer Umbau durch das Löschen und Einfügen von Aktivitäten. Die Workflow-übergreifende Fehlerbehandlung ist dafür verantwortlich, Implikationen lokaler Fehlerbehandlungen für kooperierende Workflows, z. B. ungültige Daten durch Rücksetzen oder Kompensation von Aktivitäten oder die Verletzung von Kooperationsbedingungen für zukünftige Kooperationschritte, zu erkennen und die betroffenen Workflows darüber zu informieren. Beim Vergleich der Forschungsansätze zur synchronen und asynchronen Kooperation zeigte sich, dass die meisten dieser Systeme Workflow-übergreifende Fehlerbehandlungen nur bzgl. bereits generierter Ereignisse oder bereits erbrachter Dienste berücksichtigen. Der Einfluss auf zukünftige Kooperationschritte wie die Verletzung von temporalen Kooperationsbedingungen durch Verzögerungen wird meistens nicht beachtet. Wird der Kooperationspartner aber frühzeitig über potentielle Verletzungen von Kooperationsvereinbarungen informiert, kann er rechtzeitig reagieren und z. B. seine Workflows entsprechend anpassen.

Weiter fiel auf, dass das am häufigsten betrachtete Anwendungsgebiet der E-Commerce ist, obwohl auch bei Verarbeitungsvorgängen im medizinischen Bereich oder in der öffentlichen Verwaltung Bedarf für den Einsatz von Workflow-Systemen vorhanden ist und gerade dort komplexe Abhängigkeiten auftreten.

Die betrachteten Beispiele zeigten, dass sowohl synchrone als auch asynchrone Kooperationsformen gebraucht werden je nach Anwendungsschwerpunkt und seinen Anforderungen. Für gut strukturierte Kooperations Szenarien, in denen die Aufgaben der einzelnen Organisationen klar definiert sind, mehrere Aktivitäten umfassen und auf Ergebnisse gewartet werden soll, bietet sich eine synchrone Kooperation an, während eine asynchrone Kooperation eine höhere Flexibilität für die Unterstützung komplexer Interaktionsmuster bietet. Kooperationsbedingungen

sind wie in Abschnitt 2.3 erläutert bei beiden Kooperationsformen möglich.

Aufbauend auf den beim Vergleich der Systeme gewonnen Erkenntnissen soll nun als nächstes, ausgehend von verschiedenen Kooperationsszenarien (z. B. aus der Medizin und dem E-Commerce), ein Modell für die Beschreibung einer Workflow-Kooperation gefunden werden, das unter anderem die Modellierung von Kooperationsbedingungen und eine Workflow-übergreifende Fehlerbehandlung unterstützt.

8 Literatur

- [1] Alonso, G.; Agrawal, D.; El Abbadi, A.; Kamath, M.; Günthör, R.; Mohan, M. (1996): Advanced Transaction Models in Workflow Contexts. Proceedings of the 12th International Conference on Data Engineering, New Orleans, Feb. 1996: S. 574-581.
- [2] Bartelt, A.; Lamersdorf, W. (2000): Agent-Oriented Concepts to Foster the Automation of E-Business. In Proceedings of the 11th International Workshop on Database and Expert Systems (DEXA 2000).
- [3] Borghoff, U.; Schlichter, J. H. (1995): Rechnergestützte Gruppenarbeit. Springer, Berlin et al.
- [4] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; Maler, E. (2000): Extensible Markup Language (XML) 1.0 Specification. October 2000, <http://www.w3.org/TR/REC-xml>.
- [5] Brenner W.; Zarnekow, R.; Wittig, H. (1998): Intelligente Softwareagenten: Grundlagen und Anwendungen. Springer, Berlin.
- [6] Casati, F. (1999): Semantic Interoperability in Interorganizational Workflows. In Ludwig, H. (ed.) (1999): Proceedings of WACC Workshop on Cross-Organizational Workflow Management and Co-Ordination, San Francisco, February 22, 1999.
- [7] Casati, F.; Discenza, A. (2000): Modeling and Managing Interactions among Business Processes. Technical Report HPL-2000-159, Hewlett-Packard-Laboratories.
- [8] Casati, F.; Pernici, B.; Pozzi, G.; Sánchez, G.; Vonk, J. (1999): Conceptual Workflow Model. In Grefen, P.; Pernici, B.; Sánchez G. (eds.) (1999): Database Support for Workflow Management. The WIDE Project. Kluwer Academic Publishers, Boston, S. 23-46.
- [9] Dayal, U.; Hsu, M.; Ladin, R. (1991): A Transactional Model for Long-running Activities. Proceedings of the 17th. Intl. Conference on Very Large Data Bases, Barcelona, Spain: S. 113-122.
- [10] Eder, J.; Groiss, H.; Liebhart, W. (1998): The Workflow Management System Panta Rhei. In Dogac, A.; Kalinichenko, L.; Tamer Ozsu, M.; Sheth, A. (eds.) (1998): Workflow Management Systems and Interoperability, NATO Advanced Study Institute, Springer: 129-144.
- [11] Eder, J.; Panagos, E. (1999): Towards Distributed Workflow Process Management. In Proc. of the WACC workshop on Cross-organizational workflows. San Francisco, CA, USA.
- [12] Georgakopoulos, D.; Hornick, M.; Sheth, A. (1995): An Overview of Workflow-Management: From Process Modelling to Infrastructure for Automation. Journal on Distributed and Parallel Database Systems 3 (2): 119-153.
- [13] Georgakopoulos, D.; Schuster, H.; Cichocki, A.; Baker, D. (1999): Managing Process and Service Fusion in Virtual Enterprises. In Information Systems, Special Issue on Information Systems Support for Electronic Commerce, Fall 1999.
- [14] Georgakopoulos, D.; Schuster, H.; Baker, D.; Cichocki, A. (2000): Managing Escalation of Collaboration Processes in Crisis Mitigation Situations. In Proceedings of ICDE 2000: 45-56.
- [15] Gray, J.; Reuter, A. (1993): Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers.
- [16] Grefen, P.; Vonk, J.; Boertjes, E.; Apers, P. (1999): Semantics and Architecture of Global Transaction Support in Workflow Environments. In Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS'99), Edinburgh, Scotland, September 1999. IEEE Computer Society Press.
- [17] Grefen, P.; Aberer, K.; Ludwig, H.; Hoffner, Y. (2001): CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. In IEEE Bulletin of the Technical Committee on Data Engineering, Vol. 24, No.1, March 2001.
- [18] Groiss, H.; Eder, J. (2000): Bringing Workflow Systems to the Web. <http://www.ifi.uni-klu.ac.at/ISYS/JE/Projects/Workflow/pro>.
- [19] Hagen, C.; Alonso, G. (1999): Beyond the Black Box: Event-based Inter-Process Communication in Process Support Systems. In: 19th International Conference on Distributed Computing Systems (ICDCS),

- Austin, Texas, USA, June 1999.
- [20] Hasenkamp, U.; Kirn, S.; Syring, M. (Eds.) (1994): CSCW - Computer Supported Cooperative Work. Addison-Wesley, Bonn et al.
 - [21] Heintz, P.; Horn, S.; Jablonski, S.; Neeb, J.; Stein, K.; Teschke, M. (1999): A Comprehensive Approach to Flexibility in Workflow Management Systems. WACC 1999, S. 79-88.
 - [22] Heinlein, C. (2000): Workflow and Process Synchronization with Interaction Expressions and Graphs. In: Ulmer Informatik-Berichte, Nr. 2000-11.
 - [23] Hoffner, Y. (1999): Architecture Description. Report D3.a zu ESPRIT-Projekt E/28635, Cross-Organisational Workflow, CrossFlow.
 - [24] IBM Company (1999): IBM MQSeries Workflow: Concepts and Architecture. Version 3.2.1. IBM Company.
 - [25] Jablonski, S.; Böhm, M.; Schulze, W. (Eds.) (1997): Workflow-Management: Entwicklung von Anwendungen und Systemen. dpunkt-Verlag, Heidelberg.
 - [26] Kamath, M.; Ramamritham, K. (1998): Failure Handling and Coordinated Execution of Concurrent Workflows. Proceedings of 14th International Conference on Data Engineering (ICDE'98), Orlando, Florida, February 1998: S. 334-341.
 - [27] Lazcano, A.; Alonso, G.; Schuldt, H.; Schuler, C. (2000): The WISE approach to Electronic Commerce. International Journal of Computer Systems Science & Engineering, special issue on Flexible Workflow Technology Driving the Networked Economy, Vol. 15, No. 5, September 2000.
 - [28] Lazcano, A.; Schuldt, H.; Alonso, G.; Schek, H. (2001): WISE: Process based E-Commerce. In IEEE Bulletin of the Technical Committee on Data Engineering, Vol. 24, No.1, March 2001.
 - [29] Ludwig, H. (1999): Termination Handling in Inter-Organisational Workflows - An Exception Management Approach. In: Antola, A. (Ed.): Proceedings of the Seventh Euromicro Workshop on Parallel and Distributed Processing (PDP '99), Funchal, February 1999. IEEE Computer Society, Los Alamitos, 1999: 122 - 129.
 - [30] Ludwig, H.; Hoffner, Y. (1999): Contract-based Cross-Organisational Workflows. The CrossFlow Project. In Proc. of the WACC workshop on Cross-Organizational Workflow Management and Co-Ordination, San Francisco, February 22, 1999.
 - [31] Mehta, B.; Levy, M.; Andrews, G.; Beckman, B.; Klein, J.; Mital, A. (2001): BizTalk Server 2000 Business Process Orchestration. In IEEE Bulletin of the Technical Committee on Data Engineering, Vol. 24, No.1, March 2001.
 - [32] Merz, M.; Liberman, B.; Lamersdorf, W. (1997): Using Mobile Agents to support Interorganizational Workflow Management. In: International Journal on Applied Artificial Intelligence, 11(6), September 1997, S. 551ff.
 - [33] Müller, R.; Rahm, E. (1999): Rule-Based Dynamic Modification of Workflows in a Medical Domain. German Database Conference (BTW), Springer 1999. <http://dol.uni-leipzig.de>
 - [34] Müller, R.; Rahm, E. (2000): Dealing with Logical Failures for Collaborating Workflows. In: Etzion, O.; Scheuermann, P. (eds.): Proceedings of Fifth International Conference on Cooperative Information Systems (CoopIS), Eilat, Israel, Sept. 2000. LNCS 1901, Springer, Berlin, 2000: 210-223.
 - [35] Object Management Group, Inc. (1997): CORBAservices: Common Object Services Specification. <http://www.omg.org>.
 - [36] Reichert, M.; Dadam, P. (1998): ADEPT_{FLEX} - Supporting Dynamic Changes of Workflows Without Loosing Control. Journal of Intelligent Information Systems 10 (1998), S. 93-129.
 - [37] Schönhoff, M.; Störmer, H. (2001): Trading Workflows Electronically: The ANAISOFT Architecture. In Proceedings der 9. Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW 2001), 7.-9. März 2001, Oldenburg.
 - [38] Shepherdson, J. W.; Thompson, S. G.; Odgers, B. R. (1999): Cross Organisational Workflow Co-ordinated by Software Agents. In Ludwig, H. (ed.) (1999): Proceedings of WACC Workshop on Cross-Organizational Workflow Management and Co-Ordination, San Francisco, February 22, 1999.
 - [39] Stricker, C.; Riboni, S.; Kradofer, M.; Taylor, J. (2000): Market-based Workflow Management for Supply Chains of Services. IEEE Hawaiian International Conference on System Sciences, Jan. 4-7, 2000.
 - [40] The Workflow Management Coalition (1996): Workflow Management Coalition Workflow Standard - Interoperability Abstract Specification. Version 1.0. Document Number WFMC-TC-1012, October 1996.
 - [41] Workflow Management Coalition (1999): Basic Terminology & Glossary. Workflow Management Coalition, <http://www.wfmc.org>.
 - [42] UN/EDIFACT Working Group: United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport. <http://www.unece.org/trade/untdid>.
 - [43] Vossen, G.; Becker, J. (Eds.) (1996): Geschäftsprozessmodellierung und Workflow-Management: Modelle,

Methoden, Werkzeuge. Thomson, Bonn.

- [44] Woolridge, M.; Jennings, N. R. (1995): Intelligent Agents: Theory and Practice. In *The Knowledge Engineering Review*, 10(2), S. 115-152.
- [45] Worah, D.; Sheth, A. (1997): Transactions in Transactional Workflows. In Jajodia, S.; Kerschberg, L. (Eds.) (1997): *Advanced Transaction Models and Architectures*. Kluwer, Boston.