Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner

Interner Bericht

Anpassungsfähigkeit in Workflow-Systemen

Modellierung und Beschreibung anpassungsfähiger Workflows

Status: **Final**
Vom: **30.6.1995**
Seiten: **50**

Von: Dipl.-Inform. Reiner Siebert
Abteilung Anwendersoftware

IPVR
Breitwiesenstraße 20-22
D-70565 Stuttgart
Telefon: D-711/ 7816-431
E-Mail: Reiner.Siebert@informatik.uni-stuttgart.de

Inhaltsverzeichnis

Inhalt und Aufbau des Berichts

Der vorliegende interne Bericht, entstanden am Institut für Parallele und Verteilte Höchstleistungsrechner (Abteilung Anwendersoftware) der Universität Stuttgart beschäftigt sich mit der Modellierung und Beschreibung von Workflows. Die vorläufigen Ergebnisse stammen aus Arbeiten im Zusammenhang mit dem Verbundprojekt PoliFlow, das vom BMBF gefördert wird.

Einleitend werden grundlegende Begriffe und Kenntnisse der Prozeßmodellierung erörtert und Besonderheiten von Prozeßmodellen in verschiedenen Anwendungsbereichen herausgestellt. Die Software-Entwicklung und das Geschäftsprozeßmanagement werden als Anwendungsbereiche für Prozeßmodelle betrachtet. Zielsetzung und Eigenheiten dieser Gebiete werden mit denjenigen der Workflow-Management-Systeme (WFMS) verglichen.

Die Kapitel zwei und drei befassen sich mit den Eigenschaften eines Prozeßmodells für WFMS und den Inhalten einer allgemeinen Workflow-Beschreibung. Dabei werden die Notwendigkeit mehrerer modularer Aspekte (Sichten) des Modells begründet und verschiedene Sichten einer Workflow-Beschreibung vorgestellt.

Im vierten Kapitel werden nichtfunktionale Anforderungen an interne und externe Workflow-Beschreibungen aufgestellt und in tabellarischer Form – als Anforderungskatalog – zusammengestellt. Die vorhergehenden Kapitel bilden hierfür die wesentliche Grundlage.

Das fünfte Kapitel beschäftigt sich mit verschiedenen Arten von Prozeßbeschreibungen bzw. der hierfür verwendeten Formalismen. Neben netzorientierten Beschreibungsformalismen werden prozedurale und regelbasierte Beschreibungen betrachtet und jeweils einige Beispiele vorgestellt. Dabei wird deren Eignung zur Beschreibung von Workflows untersucht und ihre Eigenschaften mit den oben ermittelten Anforderungen verglichen.

Im sechsten Kapitel werden die Ergebnisse der Untersuchungen abschließend noch einmal zusammengefaßt und einige laufende Projekte mit ihren derzeitigen Workflow-Beschreibungen vorgestellt.

1. Einleitung

Workflow-Management-Systeme dienen grundsätzlich der Steuerung und Kontrolle von Prozessen in unterschiedlichen Anwendungsgebieten. Somit stellt die Workflow-Beschreibung eine Modellierung dieser Prozesse dar, welche die für die Steuerung und Kontrolle wesentlichen Aspekte der Vorgänge verdeutlicht.

Nach einer kurzen Einführung in die Prozeßmodellierung werden Erfahrungen aus verschiedenen Anwendungsgebieten zusammengefaßt und deren Unterschiede in der Zielsetzung und Modellbildung diskutiert. Im Anschluß daran wird kurz die spezielle Situation bei der Workflow-Modellierung skizziert, und es werden Unterschiede zu den anderen Gebieten der Prozeßmodellierung aufgezeigt. In den weiteren Unterkapiteln werden einige Begriffe aus dem Bereich des Workflow-Managements diskutiert.

1.1 Prozeßmodellierung

Jedes Modell stellt eine abstrakte Repräsentation der realen Welt bzw. eines Ausschnitts dieser realen Welt dar. Bei der Modellbildung werden nicht alle Details der Realität abgebildet; vielmehr versucht man, die Komplexität durch Abstraktion zu reduzieren. Dabei werden irrelevante Details eliminiert und bestimmte Charakteristika durch die Wahl des Modells hervorgehoben.
Die Modellbildung ist somit ein stark individueller Vorgang und wird zum einen von der Aufgabe bzw. der Verwendung des Modells geprägt; zum anderen ist sie stark von der subjektiven Vorstellung abhängig, die der Erzeuger des Modells von der realen Welt besitzt.

In welchem Sinne der Begriff Prozeßmodell verwendet wird, wird durch eine Definition aus [CuKeOv92], S. 76 verdeutlicht: “A process model is an abstract description of an actual or proposed process that represents selected process elements that are considered important to the purpose of the model and can be enacted by a human or machine.”
Diese Definition spiegelt im wesentlichen die Sachverhalte der Abstraktion und Selektion wider und betont die Abhängigkeit des Modells nicht nur von dem realen Vorgang, sondern auch von der Aufgabe und dem Einsatz des Modells. Darüber hinaus wird die Möglichkeit zur Ausführung des Prozeßmodells gefordert, was auch für Workflow-Beschreibungen relevant ist.

In der Regel werden zwei Aspekte von Prozeßmodellen gegeneinander abgegrenzte und separat untersucht: Die Notationen (was wird wie beschrieben?) und die Werkzeuge (wie und wofür wird die Beschreibung eingesetzt?). Da die Aufgabe der Kontrolle und Steuerung der Prozesse bei Workflow-Systemen eine so dominante Rolle spielt und der Einsatz des Modells auch die Notationen in gewisser Weise beeinflußt, werden beide Aspekte im folgenden gemeinsam betrachtet.

Betrachtet man verschiedene Ansätze innerhalb der Prozeßmodellierung, kann man diese in zwei Gruppen einteilen (vgl. [Longch93]):

Beim beschreibenden (deskriptiven) Ansatz werden existierende Gegebenheiten und Aspekte der Prozesse untersucht und beschrieben. Dieser Ansatz ist eher gegenwartsbezogen (wie wird ein Prozeß derzeit ausgeführt) oder sogar vergangenheitsbezogen (wie wurde er ausgeführt). Er eignet sich daher besonders zur einfachen, informalen oder halbformalen Beschreibung von Prozessen, die zum Verständnis, zur Verständigung oder zur Standardisierung beitragen kann. Eine weitere Einsatzmöglichkeit für deskriptive Modelle ist die Analyse bestehender oder erwünschter Prozesse anhand mehr oder weniger formaler Methoden, durch welche die Prozesse simuliert, verglichen, bewertet oder verbessert werden können.

Weitere Zielgebiete sind die Anleitung und die Ausführung der Prozeßbeschreibungen. Bei der Anleitung wird die Durchführung der Prozesse durch das Bereitstellen von Informationen unterstützt (z.B. aktueller Zustand, nachfolgende Prozeßschritte). Bei der Ausführung hingegen soll die Beschreibung maschinell interpretierbar sein und die Durchführung direkt unterstützen. Diese kann sowohl maschinell (durch Programme/Maschinen) als auch manuell (durch Personen) erfolgen.
Für die beiden letzten Zielgebiete eignet sich auch der zweite Ansatz innerhalb der Prozeßmodellierung, der vorschreibende (präskriptive) Ansatz. Dieser ist eher zukunftsorientiert; es wird beschrieben bzw. vorgeschrieben, wie ein Prozeß durchgeführt werden soll. Wird diese Vorschrift negativ formuliert, im Sinne von Verboten, dann spricht man von einem proskriptiven Ansatz.

Abhängig vom Modell kann ein Prozeß sowohl auf verschiedenen Abstraktionsgraden als auch durch unterschiedliche Sichten auf den Prozeß beschrieben werden. Weit verbreitet sind hier die folgenden vier Aspekte:

- Funktionale Aspekte - was wird ausgeführt?
- Verhaltensbezogene Aspekte - wann und unter welchen Bedingungen?
- Organisatorische Aspekte - wo und bei wem wird ausgeführt?
- Informationsbezogene Aspekte - welche Informationen/Produkte sind relevant?

1.2 Anwendungsgebiete der Prozeßmodellierung

Zwei Anwendungsgebiete, in denen Prozeßmodelle eine wichtige Rolle spielen sind das Geschäftsprozeßmanagement und die Software-Entwicklung. Diese werden zuerst betrachtet, bevor die Rolle der Prozeßmodellierung für WFMS diskutiert wird.

Geschäftsprozeßmodellierung

Methoden und Tools der Geschäftsprozeßmodellierung (Business Process Modeling, BPM) werden in unterschiedlichen Geschäftsbereichen eingesetzt. Dazu gehören vor allem die Produktion, Entwicklung und Verwaltung. Dabei ist der Trend von der statischen, strukturorientierten Sichtweise zu einer dynamischen, verhaltensorientierten Sichtweise zu beobachten [FeSiEl93].
Bei der strukturorientierten Sicht stehen die funktionalen Einheiten und die Aufbaustruktur im Vordergrund, bei der das Unternehmen in Untereinheiten aufgegliedert wird. Bei der verhaltensorientierten Sicht hingegen ist eine starke Prozeßorientierung festzustellen, wobei die Ablaufstruktur in den Vordergrund rückt.

Neben den oben genannten Zielbereichen, welche die Simulation, Optimierung und Vorgangssteuerung einschließen, sind im Bereich des BPM vor allem folgende Aufgaben zu erwähnen:

- Modellierung der Unternehmensprozesse, wobei die Prozesse von der strategischen bis zur operativen Ebene verfeinert werden und in ständigen Überarbeitungen verbessert und optimiert werden können.
- Modellierung des Zielsystems, das auf den Entwicklungsprozeß abgebildet werden kann.
- Besondere Berücksichtigung der Leistungserstellung im Zusammenhang mit dem Faktor Zeit, wodurch kürzere Reaktionszeiten und somit Wettbewerbsvorteile erreicht werden können.
- Erstellung ganzheitlicher Informationssysteme, für die das BPM eine Grundlage darstellt.
- Simulation, Analyse
- Reengineering - kontinuierliche Verbesserung der Geschäftsprozesse

Diskussion bestehender BPM-Methoden und -Tools finden sich in der Fachliteratur, u.a. in [KopMay95], [FeSiEl93], [Leinga95], bzw. in den dort angegebenen Referenzen. An dieser Stelle seien lediglich einige bekannte Verfahren mit den wesentlichen Konzepten aufgelistet:

- Business Systems Planning (BSP) - basierend auf Geschäftsprozessen (Aktivitäten und Entscheidungen), Datenklassen und deren Beziehungen.
- Information Engineering (IE) - basierend auf Datenobjekten und Geschäftsprozessen, die über Ereignisse gekoppelt sind.
- Vorgangskettenanalyse - Automatisierung von an die Aufbauorganisation gekoppelten Vorgangsketten; integrierte Daten- und Funktionssicht.
- Semantisches Objektmodell (SOM), hierbei handelt es sich um eine objektorientierte Methode, bei der geschäftliche Transaktionen, deren Sachziele und die Koordination beschrieben werden.
- Netzmodelle - Modellierung von Aktivitäten und Zuständen mit deren Übergängen basierend auf Petrinetzen.

Software Engineering

Ein weiteres Anwendungsgebiet, in dem die Prozeßmodellierung eine wichtige Rolle spielt, ist das Software Engineering bzw. die damit verbundene Modellierung von Software-Entwicklungsprozessen. Besonders in der Forschung konnte man hier in den letzten Jahren eine rege Tätigkeit beobachten.

Neben der Planung der Software-Entwicklungsprozesse, liegt das Nutzenpotential in diesem Anwendungsgebiet vor allem im Erfassen der Prozesse zur Verständigung über Abläufe, in der Unterstützung der Kommunikation zwischen Subprozessen und in der ständigen Verbesserung der Prozesse. Weitere wesentliche Aufgaben sind die Messung und Auswertung relevanter Prozeßdaten (z.B. durch Metriken) sowie die Anleitung und Ausführung der Prozesse.

Die Software-Entwicklungsprozesse bzw. deren Modellierung zeichnen sich durch einige spezifische Merkmale aus, weshalb häufig deskriptive Modelle, teilweise aber auch präskriptive (oder proskriptive) Modelle verwendet werden. Durch die große Dynamik und Unbestimmtheit der Prozesse basieren viele Modelle auf in- oder halbformalen Beschreibungssprachen mit textuellen Notationen. Zur Verdeutlichung seien hier einige Merkmale aufgeführt (vgl. [KelRom91]):

- Die allgemeine Vorgehensweise in Software-Entwicklungsprozessen ist selbst noch Gegenstand der Forschung und stark abhängig von der Art der Projekte. Dadurch sind zum Teil unzureichende Kenntnisse über die Prozesse selbst vorhanden, was eine Prozeßbeschreibung erschwert.
- Innerhalb der Prozesse existieren eine Vielzahl an Aktivitäten, die eine kreative Problemlösung erfordern und deren Ergebnisse zum Teil schwer vorhersehbar sind. Gleichzeitig kann jedoch das weitere Vorgehen stark von diesen Ergebnisse abhängig sein.
- Wichtig ist die Unterstützung kreativer Gruppenarbeit, weshalb dem Datenfluß und der Produkt-/Dokumentenverwaltung eine große Bedeutung zukommt.
- Verschiedene Aufgaben können maschinelle oder manuelle Bearbeitung und die Integration von Tools/Dokumenten erfordern, was zu Ressourcenkonflikten führen kann.
- Temporale Aspekte, sowohl in der Art von Versionenverwaltung, als auch durch die explizite Berücksichtigung von Zeitpunkten und Zeitdauer sind zu integrieren.
- Durch die erwünschte Prozeßverbesserung sind Messungen und Auswertungen der projektrelevanten Informationen von großer Bedeutung. Dies ist durch die unscharfe Vorstellung des Prozesses und die Art der zu erstellenden Produkte (Software, Dokumente) jedoch häufig problematisch (Metriken).

Mit aus diesen Gründen werden häufig regel- oder wissensbasierte Sprachen zur Beschreibung der Softwareprozesse eingesetzt. Darüber hinaus finden sich aber auch einige andere Ansätze, die in unterschiedlichen Verfahren und Tools zum Einsatz kommen. Die folgende Übersicht soll lediglich einen Überblick über relevante Ansätze geben; darüber hinaus sei auf die Fachliteratur verwiesen ([CuKeOv92], [KelRom91], [Longch93], [LotRom93], [Finkel94]).

- Modelle basierend auf der Auswertung von Regeln und Bedingungen
  (OPIUM, AP5, POS, MERLIN)
- Modelle basierend auf Petrinetz-ähnlichen Beschreibungen und Zustandsübergängen
  (SPADE, PMDB+/SARA, Statemate)
- Natürlichsprachige Modelle
  (WSH, KyotoDB)
- Programmiersprachen-ähnliche prozedurale Modelle (APPL/A)

1.3 Workflow-Modellierung

Workflow-Systeme werden in der Regel nicht für eine spezielle Anwendungsumgebung entwickelt, sondern sollen als Informationssysteme in unterschiedlichen Bereichen eingesetzt werden können. Somit sollten die Grundbestandteile eines WFMS geeignet sein, Vorgänge in unterschiedlichen Anwendungsbereichen zu kontrollieren und zu steuern. Dies setzt wiederum voraus, daß diese unterschiedlichen Vorgänge in einem gemeinsamen Modell beschrieben werden können.

In einem Prozeßmodell zur Beschreibung von Workflows sollten also grundsätzlich auch die oben genannten Geschäfts- und Software-Entwicklungsprozesse dargestellt werden können. Auch die Aufgaben, die WFMS erfüllen sollen, ähneln den zuvor genannten in weiten Teilen. So müssen die entsprechenden Prozesse definiert werden, wodurch die Grundlage für Standardisierungen, Kommunikation und Schulung, Simulationen, Analysen und spätere Prozeßverbesserungen (Reengineering) gelegt werden. Die definierten Prozesse sollen dann durchgeführt werden, wobei ebenfalls wieder funktionale, verhaltensbezogene, organisatorische und informationsbezogene Aspekte zu berücksichtigen sind [CuKeOv92].

Nahezu alle angesprochenen Prozeßmodelle beschreiben Aktivitäten, Objekte und Rollen [KelRom91]. Trotz dieser gemeinsamen Aufgaben und Funktionen haben diese in den verschiedenen Prozeßmodellen doch einen unterschiedlichen Stellenwert.

Im letzten Unterkapitel stellten wir fest, daß im Software Engineering bereits die Modellierung der gewünschten Prozesse ein Problem darstellt. Dadurch besteht eine Hauptaufgabe bereits in der Beschreibung geeigneter Prozesse, wozu häufig deskriptive Modelle verwendet werden. Im Workflow-Management (WFM) dagegen kommt dieser Problematik keine große Bedeutung zu. In der Regel geht man davon aus, daß die durchzuführenden Prozesse und deren Eigenschaften bekannt und geeignet sind. Ähnliches gilt für die Modellierung eines Zielsystems im BPM.
Auch sind die Inhalte der Aktivitäten für Workflow-Systeme von sekundärem Interesse. Aufgabe der WFMS ist es, verschiedene Aktivitäten zu koordinieren und deren Ausführung im

Rahmen eines Geschäftsprozesses zu steuern. Weiteres Wissen über diese Aktivitäten ist nur notwendig um z.B. den Bearbeitungszustand zu erfassen oder die notwendigen Ressourcen für die Bearbeitung bereitzustellen.

Den Aufgaben, die Systeme im Zusammenhang mit der Prozeßmodellierung zu erfüllen haben, kann in verschiedenen Anwendungsgebieten also eine unterschiedliche Bedeutung zukommen. Zu den Schwerpunkten der WFMS gehören [Jablon95a]:

- Die automatisierte Steuerung und Kontrolle der Prozesse; die Ausführbarkeit der Workflow-Beschreibung ist überhaupt *der* wesentliche Faktor, weshalb auch überwiegend präskriptive Modelle zum Einsatz kommen.
- Unterstützung der Koordination und Kommunikation zusammenhängender Teilaufgaben
- die Gewährleistung der Sicherheit und Konsistenz zu jedem Zeitpunkt der Ausführung
- höchstmögliche Flexibilität und Anpassungsfähigkeit der zu steuernden Vorgänge
- die Integration von Tools zur rechnergestützten und vollautomatischen Bearbeitung bestimmter Teilaufgaben und für Telekooperationsdienste
- Interoperabilität und weit verteilte Ausführung der Workflows

In der Vergangenheit wurden Workflow-Systeme überwiegend in den Zusammenhang mit stark strukturierten Prozessen gebracht. Je stärker die Struktur eines Prozesses vorgegeben ist und je früher die Prozeßparameter festgelegt werden können, um so einfacher läßt sich die Durchführung der Prozesse automatisieren. Andererseits wird durch die Beschränkung auf die Unterstützung dieser strukturierten Workflows das Anwendungsfeld stark eingeschränkt. Mittlerweile gewinnen Ad Hoc Workflows und die Anpassungsfähigkeit bereits definierter Workflows zunehmend an Bedeutung. So sollen auch Prozesse gesteuert werden, deren Durchführung nicht vollständig oder gar nicht vordefiniert werden kann [Duitsh94].

In Abschnitt 1.1 wurden vier wesentliche Aspekte eines Prozesses vorgestellt (Funktionalität, Verhalten, Information und Organisation). Generell liegt jedoch weder die Anzahl der notwendigen Sichten fest, noch sind die Inhalte dieser Sichten zwingend vorgegeben. Dennoch werden die Eigenschaften eines Workflow-Modells stark von der Wahl und Gestaltung dieser Sichten beeinflußt. Eine wesentliche Voraussetzung für verschiedene Qualitätsfaktoren des Modells ist nach [Jablon95a] die Modularität dieser Aspekte. Jablonski stellt fünf Anforderungen an die Modularität eines Workflow-Modells vor, die hier noch durch die modulare Erweiterbarkeit ergänzt werden.

- Modulare Dekomposition -
  Teilaspekte des Workflow-Modells können getrennt untersucht und beschrieben werden.
- Modulare Komposition -
  Teilaspekte können beliebig kombiniert werden, um Zusammenhänge auszudrücken.
- Modulare Verständlichkeit -
  Teilaspekte stellen abgeschlossene Einheiten dar, die unabhängig voneinander verständlich sind.

- Modulare Kontinuität -
  Prozeßänderungen bezüglich eines Aspektes wirken sich lediglich in Änderungen innerhalb dieses Teilaspektes der Workflow-Beschreibung aus.
- Modulare Fehlerbegrenzung -
  Fehlerhafte Prozeßpezifikationen innerhalb eines Aspektes können nur zu unerwünschtem Verhalten bezüglich dieses Aspektes führen.
- Modulare Erweiterbarkeit -
  Das Modell kann durch Hinzunahme weiterer Aspekte bzw. Teilaspekte ergänzt werden.

Vom Einsatz leistungsfähiger WFMS verspricht man sich die Steigerung der Effektivität und Produktivität (Unterstützung kreativer und Automatisierung einfacher Nebentätigkeiten). Neben quantitativen Kriterien und Kostenreduzierung läßt sich auch die Prozeßqualität (durch geringere Durchlaufzeiten, höhere Flexibilität, Reaktionsmöglichkeit etc.) verbessern. Dies schließt auch Qualitätskriterien wie Informationstransparenz und Konsistenz der Prozeßdurchführung ein. Durch die Integration von Tools sowie Telekommunikations- und -kooperationsdienste lassen sich Prozesse zum Teil grundlegend verbessern.
Weitere wesentliche Vorteile lassen sich aus den Informationen erzielen, die mit Hilfe eines WFMS über die Prozesse gewonnen werden können. Dies bezieht sich sowohl auf die Laufzeit (hohe Auskunftsbereitschaft, Rückfragemöglichkeiten, Entscheidungsunterstützung etc.) als auch auf die Auswertung der beendeten Prozesse (Management-Informationssystem, Reengineering).

1.4 Begriffe

In der Fachliteratur finden sich immer häufiger individuelle Definitionen, die
- verschiedene Aspekte des Workflows (z.B. die Datenfluß, Kontrollfluß der Prozesse)
- verschiedene Phasen des Workflow Managements (Definition, Ausführung, Verwaltung) oder
- verschiedene Komponenten (Organisationsservice, Kommunikationsservice, Engine)
der WFMS besonders hervorheben.
In diesem Bericht erfolgt die Verwendung von Begriffen in Übereinstimmung mit dem Glossar [WFMC94] der Workflow Management Coalition (WfMC), aus dem zwei der folgenden Definitionen entnommen wurden. Darüber hinaus wurden aus der Vielzahl weiterer Definitionen eine abstrakte und eine detailliertere Variante ausgewählt:

- Workflow Management System: “A system that completely defines, manages and executes workflow processes through the execution of software whose order of execution is driven by a computer representation of the workflow process logic.” [WFMC94]
- Workflow / Workflow Process: “The computerized facilitation or automated component of a process. (A Workflow is that part of a process that is automatable using workflow software products).” [WFMC94]

- “Workflow Management is the computer assisted management of business processes through the execution of software whose order of execution is controlled by a computerized representation of the business processes“ [JosWWW95].
- “Workflow management software is a proactive computer system which manages the flow of work among participants, according to a defined procedure consisting of a number of tasks. It coordinates user and system participants, together with the appropriate data resources, which may be accessible directly by the system or off-line, to achive defined objectives by set deadlines. The coordination involves passing tasks from participant to participant in correct sequence, ensuring that all fulfil their required contributions, taking default actions when necessary.” [Jablon95b], S. 14

2. Basisaspekte einer Workflow-Beschreibung

Mit einer Workflow-Beschreibungssprache soll also der Ablauf unterschiedlichster Prozesse beschrieben werden, wobei auf eine modularisierte Darstellung der einzelnen Aspekte besonderen Wert gelegt wird. In der Literatur findet man verschiedene Möglichkeiten, diese Aspekte zu gliedern [Longch93], [MarRom89], [Jablon95a], die sich zum Teil an dem speziellen Anwendungsfeld der Prozesse orientieren, zum Teil aber auch allgemeine Lösungsansätze darstellen.
Ein allgemein üblicher Ansatz ist die Trennung der funktionalen, der verhaltensbezogenen, der informationalen und der organisatorischen Aspekte [CuKeOv92].
Wie in zahlreichen Prozeßmodellen in der Vergangenheit, bilden diese vier Basisaspekte auch den Grundstock der Workflow-Modelle, wenngleich den verhaltensorientierten Aspekten eine besondere Bedeutung zukommt.

2.1 Funktionale Aspekte

Die funktionalen Aspekte eines Workflows beschreiben, was innerhalb dieses Workflows ausgeführt werden soll bzw. aus welchen Bestandteilen sich der Workflow konstituiert.

Ein Workflow kann sich aus mehreren Subworkflows und Aktivitäten zusammensetzen. Subworkflows können wiederum verfeinert und bis zu Aktivitäten aufgelöst werden. Aktivitäten besitzen aus Sicht des Prozesses keine weitere interne Struktur. Der eigentlich produktive Teil eines Prozesses liegt somit in der Ausführung der einzelnen Aktivitäten, während diese durch Workflows und Subworkflows gesteuert und kontrolliert werden.
Das Prinzip der Dekomposition wäre bei einer Auflösung der Workflows bis auf Aktivitätenebene technisch gesehen nicht notwendig. Die Verfeinerung verbessert jedoch die Definition und Strukturierung von Workflows auf unterschiedlichen Abstraktionsniveaus und ermöglicht eine übersichtliche Darstellung der Workflows. Darüber hinaus kann sie in Verbindung mit Gültigkeitsbereichen von Workflow-Parametern und mit der Vererbung von Eigenschaften (wie Rechten) bei der Modellierung und Definition (Wiederverwendung) sinnvoll eingesetzt werden.

Ein weiterer Bestandteil des funktionalen Aspektes ist die Definition von Workflow-Parametern und Bedingungen, wie Plausibilitäts- und Integritätsbedingungen für Ein- bzw. Ausgabewerte oder die Synchronisation mit externen Ereignissen. Auch können zu den einzelnen Aktivitäten Vor- oder Nachbereitungsroutinen definiert werden.

Die Definition eines Workflows und die Deklaration seiner Subworkflows bestimmen somit den funktionalen Aspekt einer Workflow-Beschreibung [Jablon95a]. Sie beinhaltet in der Regel:

- Name des Workflows (Eindeutige Bezeichnung),
- Beschreibung des Workflows (für die betreffende Aktoren verständlich und ausreichend),
- Konsistenzbedingungen,
- Vor- oder Nachbereitungsroutinen
- Untergliederung in Subworkflows bzw. Aktivitäten
- Definition relevanter Objekte der anderen Teilaspekte
  z.B. welche Eigenschaften sind zur Bearbeitung eines Workflows erforderlich

2.2 Verhaltensbezogene Aspekte

Die verhaltensbezogenen Aspekte eines Workflows bestimmen wann und unter welchen Bedingungen Workflows und Aktivitäten ausgeführt werden bzw. in welchen Beziehungen diese miteinander stehen.
Da die Steuerung und Kontrolle der einzelnen Aktivitäten die eigentliche Aufgabe von WFMS ist, kommt den verhaltensbezogenen Aspekten eine besondere Bedeutung zu. Sie bilden die Kernfunktionalität eines WFMS, weshalb auf Mächtigkeit, Flexibilität und Übersichtlichkeit dieser Beschreibungsaspekte besonderer Wert zu legen ist.

Elementare und zusammengesetzte Kontrollstrukturen

Die üblichen, aus den Programmiersprachen bekannten Kontrollkonstrukte, die auch sonst bei der Prozeßmodellierung eingesetzt werden, sind mit Sicherheit auch für die Workflow-Beschreibung erforderlich:

- serielle Ausführung (Sequenz),
- alternative Ausführung (bedingte Verzweigung),
- parallele Ausführung (unbedingte Verzweigung),
- Schleifen (for, while, repeat until) in sequentieller und paralleler Ausführung

Darüber hinaus können zusammengesetzte oder spezielle Konstrukte die Definition erheblich vereinfachen und erhöhen die Verständlichkeit komplexer Workflows. Beispiele hierfür sind:

- Option - eine Aktivität kann, muß aber nicht ausgeführt werden.
- Reihe - die Aktivitäten einer Reihe können in beliebiger Reihenfolge (aber sequentiell) ausgeführt werden.
- m aus n - aus einer Auswahl von n können m beliebige ausgewählt werden.

- die bedingte Auswahl (Einschränkung der unbedingten Verzweigung) - aus einer Auswahl von Aktivitäten wird eine bedingte Auswahl getroffen.

Deskriptive Kontrollstrukturen

Eine weitere Klasse von Kontrollkonstrukten bilden deskriptive Strukturen, die nicht nur durch eine kompakte Beschreibung auffallen, sondern auch die Ausdrucksmächtigkeit der möglichen Kontrollstrukturen erweitern.
Die oben genannten elementaren Konstrukte besitzen die übliche Semantik, bei den neuen Konstrukten wird die entsprechende Semantik mit aufgeführt. Die Syntax wurde [Sch_ea94] entnommen.

- ->(A;B) ... Sequenz
- if (cond(); A; B) ... bedingte Verzweigung
- ||(A;B) ... unbedingte Verzweigung
- not A ... Ausschluß -
  d.h. die Durchführung von Workflow A wird explizit verboten
  (zu einem bestimmten Zeitpunkt).
- <(A;B) ... Limitierung (Deadline) - A wird limitiert durch B
  d.h. A darf nicht mehr ausgeführt werden, falls die Ausführung von B bereits begonnen hat. Ansonsten sind die Ausführungen von A und B optional.
- >(B;A) ... Verzögerung (Delay) - B wird durch A verzögert
  d.h. die Ausführung von B wird solange verzögert, bis die Ausführung von A beendet ist oder garantiert ist, daß A nicht mehr ausgeführt wird. Die Ausführungen von A und B sind wieder optional.
- ⇒(A;B) ... Existenzabhängigkeit (Existance) - A erfordert B
  d.h. A darf nur ausgeführt werden, wenn sicher ist, daß auch B ausgeführt wird. Die Ausführungsreihenfolge ist dabei nicht relevant.

Ereignisse

Neben den internen Strukturen zur Beschreibung und Definition des Kontrollflusses hängt das Verhalten eines Workflows häufig auch vom Eintreten bestimmter Ereignisse ab. Diese können sich unterschiedlich auf das Verhalten der Workflows auswirken.
Mit der Verwendung von Informationen über Ereignisse in Kontrollstrukturen können zum Beispiel bedingte Verzweigungen aufgrund des Eintretens bestimmter Ereignisse gesteuert werden. Eine weitere Möglichkeit ist die Synchronisierung von Workflows durch Ereignisse. So kann z.B. die Ausführung einer Aktivität bis zum Eintreten eines Ereignisses verzögert werden. In beiden Fällen wirken sich die Ereignisse nur zum Startzeitpunkt eines Workflows aus, was nicht zwangsläufig so sein muß.
Unabhängig vom Zeitpunkt, an dem das Ereignis eintritt, können an dieses Eintreten Aktionen gekoppelt werden; z.B. kann durch ein externes Ereignis der Start eines vordefinierten (Sub-)

Workflows ausgelöst werden. Hierfür ist es sinnvoll, die Ereignisse mit bestimmten Informationen zu versehen, die zum Beispiel zur Initialisierung des Workflows verwendet werden können.

Ereignisse sind atomare Erscheinungen; ihr Auftreten steht im Zusammenhang mit einem genau definierbaren Zeitpunkt [CharMis94]. Zur Definition und Steuerung von Workflows sind folgende Arten von Ereignissen relevant:

- Interne Ereignisse
  Zur Realisierung werden häufig interne Ereignisse verwendet, aus denen sich in der Regel Aussagen über den Verlauf interner Subworkflows oder Aktivitäten ableiten lassen. Dies kann sich auf den Start bzw. das Ende einer Aktivität (gestartet, beendet) oder auf den Erfolg der Ausführung beziehen (erfolgreich beendet, gescheitert). Internen Ereignissen kommt besonders bei der Konsistenzsicherung und Fehlerbehandlung eine große Bedeutung zu.
- Externe Ereignisse
  Mit Hilfe externer Ereignisse können äußere Einflüsse auf einen Workflows modelliert und beschrieben werden. Ein internes Ereignis eines Workflows A stellt für einen anderen Workflow B ein externes Ereignis dar, mit dem dieser evtl. synchronisiert werden soll (z.B. das Ende eines anderen Workflows oder die Erreichung eines bestimmten Zwischenstandes).
  Eine weitere Anwendungsmöglichkeit für externe Workflows stellen benutzerdefinierte Ereignisse dar, mit denen Erscheinungen der realen Welt und deren Auswirkungen für die Durchführung eines Workflows beschrieben werden können. Diese Ereignisse werden häufig als auslösendes Moment für Workflows oder Aktivitäten verwendet (z.B. soll das externe Ereignis “Reklamation” entsprechende Vorgänge innerhalb einer Auftragsbearbeitung anstoßen).
- Zeitbezogene Ereignisse
  Häufig unterliegen Workflows auch einem zeitlichen Bezug und müssen mit Terminen oder Zeiten synchronisiert werden. Man kann absolute, relative und periodische Ereignisse unterscheiden.
  Bestimmte Termine, an denen Aktivitäten koordiniert werden sollen (z.B. eine Anmeldefrist) können als absolute Ereignisse durch die Angabe des Datums und der Uhrzeit beschrieben werden. Bestimmte Fristen können dagegen mit relativen Ereignissen durch die Angabe eines bestimmten Zeitpunktes (der z.B. auch durch das Eintreten eines Ereignisses definiert werden kann) und einer zugehörigen Verzögerung modelliert werden. Mit Hilfe der dritten Kategorie, den periodischen Ereignissen sollen Aktivitäten mit mehreren, periodisch wiederkehrenden Zeitpunkten synchronisiert werden. Hiermit können z.B. Aktivitäten synchronisiert werden, die nur an bestimmten Terminen (z.B. nachts oder zum Monatsbeginn) gestartet werden dürfen.

Für eine flexible Steuerung der Workflows ist darüber hinaus wichtig, daß diese einfachen Ereignisse kombiniert und somit komplexe zusammengesetzte Ereignisse definiert und verarbeitet werden können. Beispiele für solche Ereignis-Operatoren sind die Sequenz, Disjunktion, Konjunktion, periodischer und unperiodischer Operator (vgl. [CharMis94]).

Bedingungen und Prädikate

Bedingungen oder Prädikaten kommen bei der Steuerung der Workflows eine große Bedeutung zu. Sie werden im Zusammenhang mit Kontrollstrukturen, Datenstrukturen und Ereignissen verwendet und tragen maßgeblich zur Flexibilität bei.
In Verbindung mit Kontrollstrukturen steuern sie Verzweigungen und dienen als Abbruchkriterien von Schleifen. In zusammengesetzten Kontrollstrukturen können Bedingungen auch zur Auswahl der Kontrollstrukturen selbst eingesetzt werden. So könnte man zum Beispiel eine Verzweigung definieren, deren Äste abhängig von einer Bedingung alternativ oder parallel ausgeführt werden. Auch der Einfluß von Ereignissen auf einen Workflow könnte in Abhängigkeit von Bedingungen definiert werden.

Innerhalb dieser Bedingungen sollten Informationen bzgl.

- Workflow-Kontrolldaten,
- Workflow-relevanten Applikationsdaten und
- Ereignissen und Zuständen der Workflows
- vom Benutzer definierte Sachverhalte (z.B. über Makros)

verarbeitet werden können.

2.3 Informationale Aspekte

Die informationalen Aspekte eines Workflows beschreiben Datenobjekte die innerhalb eines Workflows relevant sind. Grundsätzlich müssen zwei verschiedene Arten von Informationen innerhalb eines WFMS unterschieden werden - Kontrolldaten und Anwendungsdaten [WFMC94].

Workflow-Kontrolldaten

Die Workflow-Kontrolldaten dienen nur der Steuerung und Kontrolle des Workflows. Es handelt sich dabei um interne Daten, mit denen der Anwender keinen direkten Kontakt hat. Die Kontrolldaten werden vom Workflow-System (WFS) verwaltet und befinden sich somit eindeutig in der Kontrollsphäre des WFMS. Neben der Definition der Typen und Werte im Rahmen der Workflow-Definition, muß der Datenfluß zwischen Subworkflows (bzw. zwischen einem Workflow und seinen Subworkflows) spezifiziert werden.

Workflow-Anwendungsdaten

Bei den Workflow-Anwendungsdaten (Produktionsdaten [Jablon95a]) handelt es sich um Daten, die zur Ausführung einzelner Aktivitäten oder Workflows notwendig sind bzw. dort erzeugt werden; häufig werden sie mit Hilfe mehr oder weniger integrierter Anwendungsprogramme erzeugt und liegen somit zunächst außerhalb der Kontrollsphäre des WFMS. Bezüglich der Anwendungsdaten sind vor allem vier Aspekte von Interesse:

- Aufbau und Struktur der Daten:
  Welche Struktur besitzen die Datenobjekte, bzw. wie sind komplexe Informationsobjekte aufgebaut?
- Erzeugen und Verbrauchen von Daten:
  In welchen Aktivitäten werden Daten für spätere Aktivitäten erzeugt (Schreibzugriff)?
- Verwendung von Daten:
  Welche Daten werden zur Ausführung von Aktivitäten benötigt (Lesen/Schreiben)?
- Zugriffsrechte auf die Daten:
  Wer soll im Rahmen der Workflow-Ausführung welche Rechte an diesen Daten erhalten?

Workflow-relevante Anwendungsdaten

Die Workflow-relevanten Anwendungsdaten nehmen als dritte Art der Workflow-Daten eine Zwischenstellung ein. Es handelt sich um Teile von Anwendungsdaten oder Informationen, die aus Anwendungsdaten ermittelt werden, die aber zur Kontrolle des Workflows erforderlich sind. In der Regel werden sie dem WFMS zugeführt und dort zusätzlich verwaltet; dies kann wahlweise durch die Übergabe als Wert- oder Referenzparameter erfolgen.

Für alle Arten der Workflow-Daten können Integritätsbedingungen definiert werden, die z.B. bei der Übergabe oder beim Start eines Workflows überprüft werden. Gegebenenfalls ist auch eine Vor- oder Nachbehandlung (z.B. Konvertierung) erforderlich, falls Daten zwischen Workflows ausgetauscht werden sollen. Dies erfolgt in der Regel durch Spezifikation von Vor- oder Nachbereitungsroutinen in der Workflow-Definition (vgl. auch Kapitel 2.1).

Das WFMS kann eine unterschiedlich starke Kontrolle auf die Anwendungsdaten ausüben. Zum einen besteht die Möglichkeit, die Daten vollständig der Kontrolle und Verantwortung der Bearbeiter bzw. der Applikationen zu überlassen. Die Anwendungsdaten sollen in der Regel nicht explizit von Workflow zu Workflow weitergeleitet werden; dennoch sprechen auch Gründe dafür, dem WFMS einen Teil der Kontrolle zu übertragen. Dadurch könnten Abhängigkeiten, die sich durch das Erzeugen und Verbrauchen von Daten ergeben oder die Verfügbarkeit von notwendigen Daten in Kontrollflußentscheidungen miteinbezogen werden. Die Zugriffsrechte für Anwendungsdaten könnten exklusiv vom WFMS verwaltet werden.
In diesem Fall müßten jedoch die Datenverwendung und der Datenfluß innerhalb der Workflows definiert werden und der Zugriff auf die Anwendungsdaten über das WFMS erfolgen bzw. diesem mitgeteilt werden.

2.4 Organisatorische Aspekte

Die organisatorischen Aspekte einer Workflow-Definition beschreiben wo bzw. von wem Workflows durchgeführt werden. Je detaillierter die organisatorischen Aspekte eines Workflows beschrieben werden, desto flexibler kann die Ausführung des Workflows gestaltet werden. Dabei ist auch innerhalb der organisatorischen Aspekte auf eine gute Modularisierung zu achten.

Die Fragestellung, wem ein Workflow bzw. eine Aktivität übergeben wird, betrifft folgende Teilaspekte:

- Wer soll die Aktivitäten ausführen?
  Da mit WFMS neben der reinen Steuerung auch die Automatisierung bestimmter Aktivitäten angestrebt wird, sollen diese Personen oder Agenten zugewiesen werden können. Hierfür müssen die notwendigen Informationen verfügbar sein ("organisatorische Population" [Jablon95a], wie z.B. Name, Terminal des Bearbeiters, Pfad eines Programms, Eigenschaften der Aktoren etc.
- Wer kommt für die Ausführung der Aktivitäten in Frage?
  Der Bearbeiter einer Aktivität könnte im Zusammenhang mit der Definition der Aktivität fix vorgegeben sein, was jedoch einen starren, ungeeigneten Lösungsansatz darstellt. Im Gegensatz dazu ermöglicht die Beschreibung der organisatorischen Struktur die Auswahl einer Menge von möglichen Aktoren. Dazu gehören die Definition von Gruppen und Rollen, Abteilungsstrukturen, Projektstrukturen und Stellvertreterregelungen.
- Nach welchen Kriterien werden die Aktoren als Bearbeiter ausgewählt?
  Auch die etwas flexiblere Lösung über ein Rollenkonzept stellt noch einen relativ statischen Lösungsansatz dar. Um die Auswahl der Aktoren dynamisch zu gestalten, sollten verschiedene Zuteilungsstrategien definiert werden, nach denen die Aktoren ausgewählt werden können. Durch diese Angabe können dann dynamisch die jeweils geeignetsten Bearbeiter ermittelt werden. Möglich wäre zum Beispiel die Zuordnung nach Eigenschaften (der durch seine Fähigkeiten geeignetste Bearbeiter wird ausgewählt), nach Kapazitäten (der Bearbeiter, der derzeit die größten Kapazitäten frei hat wird ausgewählt), nach Kostengründen oder beliebige Mischformen.
  Auch müssen Regeln für die Synchronisation der Vergabe definiert werden, falls Aktivitäten an Gruppen zur Auswahl vergeben werden sollen.

Aus funktionaler Sicht sollten grundsätzlich die folgenden Zuordnungen möglich sein:
- direkte Zuordnung - durch Bestimmung eines Aktors,
- indirekte Zuordnung - durch Bestimmung einer Rolle und einer Zuteilungsstrategie,
- Zuteilung an eine Einzelperson - zur individuellen Bearbeitung,
- Zuteilung an eine Gruppe - zur alternativen Bearbeitung,
- Zuteilung mit oder ohne die Möglichkeit der Ablehnung durch die Aktoren.

3. Spezielle Aspekte einer Workflow-Beschreibung

Neben den oben genannten Basisaspekten sind aus verschiedenen Anwendungsgebieten (vgl. auch Kapitel 1) bereits weitere Sichten etabliert, wie die operationalen (Anleitung und Durchführung) und die kausalen Aspekte (Geschäftsprozeßmodellierung), oder sie wurden ausführlich in der wissenschaftlichen Literatur diskutiert (transaktionalen Aspekte).
Neben diesen häufig beschriebenen Aspekten, wurden drei weitere Bereiche identifiziert, die speziell zur Beschreibung und Durchführung von Workflows erforderlich sind. Die reine Steuerung der Prozesse stellt sicherlich die Hauptaufgabe eines WFMS dar; darüber hinaus birgt der Einsatz dieser Systeme weitere Nutzenpotentiale wie z.B. die Versorgung der Mitarbeiter und des Managements mit Vorgangsinformationen. Auch die Anpassungsfähigkeit der Workflows und Aspekte wie Datenschutz und -sicherheit in weit verteilten WFMS werden in diesem Kapitel als eigene Sichten auf Workflows diskutiert.

3.1 Operationale Aspekte

Die operationalen Aspekte einer Workflow-Beschreibung bestimmen wie ein Workflow (bzw. eine Aktivität) ausgeführt werden soll. Dabei muß zum einen berücksichtigt werden, wie stark die Durchführung der Prozesse durch das WFMS gesteuert wird; zum anderen muß auf die Ausführung der einzelnen Aktivitäten eingegangen werden, innerhalb derer die eigentliche, produktive Arbeit des Prozesses verrichtet wird.

Wie bereits in der Einleitung angesprochen wurde, kann die Durchführung der Prozesse auf unterschiedliche Art und Weise erfolgen. Es können im wesentlichen drei Grade der Prozeßsteuerung identifiziert werden:

- Information oder Empfehlung

Hierbei werden die Bearbeiter lediglich über mögliche Alternativen informiert und die Bearbeitung durch die Weitergabe relevanter Informationen unterstützt. Dem Bearbeiter bleibt sowohl die Auswahl, als auch die Durchführung der Prozesse selbst überlassen.

- Anleitung oder Führung

Im Gegensatz zur Empfehlung werden die Informationen über die Prozeßvorschrift und die Durchführung der Prozesse in die Steuerung der Workflows integriert. Das bedeutet, daß der Bearbeiter zwar selbst über die Art und Weise bestimmen kann, wie einzelne Workflows durchgeführt werden können, die darüberliegende Prozeßsteuerung erfolgt jedoch überwiegend durch das WFMS.

• Steuerung und Kontrolle

In diesem Grad übernimmt das WFMS nicht nur die volle Steuerung der Prozesse, sondern kontrolliert auch die Ausführung der einzelnen Workflows und Aktivitäten. Die Vorgehensweise innerhalb des Prozesses ist genau bestimmt und unterliegt der Kontrollsphäre des WFMS. Die Ausführung der entsprechenden Aktivitäten obliegt zwar nach wie vor einem Bearbeiter, sie kann jedoch vom WFMS überwacht und geprüft werden (z.B. durch die Prüfung auf Vorhandensein oder Konsistenz der Ergebnisse).

In allen drei Fällen besitzt das WFMS keine direkte Kontrolle über die Ausführung der Aktivitäten. Diese erfolgt entweder manuell durch einen Bearbeiter oder automatisch durch eine Applikation. Die notwendigen Informationen zur Ausführung werden dem Aktoren durch das WFMS in Form einer Beschreibung übermittelt. Sie kann Informationen über Daten, Vor- und Nachbedingungen, zu verwendende Tools aber auch Workflow-Kontrolldaten beinhalten.

3.2 Kausale Aspekte

Die Fragestellung, warum Workflows existieren bzw. wozu sie beitragen, konstituiert die kausalen Aspekte einer Workflow-Beschreibung. Kausale Informationen können ebenfalls zur Steuerung von Workflows verwendet werden. Aufgrund dieser Informationen können laufende Workflows u.U. abgebrochen oder der Ablauf existierender Workflows beeinflußt werden.

Abhängige Workflows

Häufig werden Workflows durch Ereignisse angestoßen (vgl. Abschnitt 2.2) oder in Abhängigkeit von Bedingungen ausgeführt. In diesem Fall kann es erwünscht sein, den Workflow bei Eintreten weiterer Ereignisse oder bei einer Veränderung dieser Bedingungen zu beenden oder zu verändern. Ein Workflow, der die Erstellung eines Kostenvoranschlages repräsentiert, kann zum Beispiel durch eine Stornierung abgebrochen werden (externes Ereignis). In einem anderen Beispiel kann die Bearbeitung einer Bestellung von der Zahlungsfähigkeit des Kunden abhängig gemacht worden sein. Erlischt nun diese Zahlungsfähigkeit, könnte die Auslieferung verzögert oder der gesamte Vorgang abgebrochen werden.

Auch die Abhängigkeit eines Workflows von einer Unternehmensregel ist vorstellbar, wenn zum Beispiel ein Subworkflow “Unterzeichnung durch zweiten Vorgesetzten” deshalb eingefügt wird, weil generell alle Dienstreisen mit dem Flugzeug von zwei Vorgesetzten abgezeichnet werden müssen. Ändert sich diese Regelung, ist auch die Ausführung des Subworkflows nicht mehr erforderlich.

Kontrollfluß-Steuerung aufgrund kausaler Aspekte

Durch die Spezifikation weiterer kausaler Aspekte und die Integration dieser Aspekte in den Kontrollfluß kann eine flexible Steuerung unterstützt werden. Nach der Definition eines Workflow-Zielsystems könnten z.B. die Teilziele einer Aktivität in Bedingungen von Kontrollstrukturen eingesetzt werden.

Kausale Abhängigkeiten wirken sich nicht nur zu vorher bestimmbaren Zeitpunkten auf einen Workflow aus (z.B. bei der Aktivierung eines Subworkflows), sie können sich auch zu beliebigen Zeitpunkten ändern und damit Reaktionen auslösen. Daher ist es sinnvoll, diese Abhängigkeiten entweder in regelmäßigen Abständen zu überprüfen oder die Kontrollpunkte durch den Benutzer definieren zu lassen.
Die Spezifikation kausaler Aspekte erfordert also nicht nur die Definition von Vorbedingungen, sondern auch von Bedingungen, die während der gesamten Durchführung erfüllt sein müssen.

3.3 Transaktionale Aspekte

Die transaktionalen Aspekte eines Workflows befassen sich mit der Konsistenz, der Persistenz und dem Verhalten in Ausnahme- oder Fehlerfällen.

Da Workflows u.U. lang dauernde Prozesse darstellen können, stoßen die traditionellen Datenbank-Mechanismen an ihre Grenzen. Mit diesem Problem beschäftigen sich derzeit verschiedene Forschungsprojekte im Rahmen der erweiterten Transaktionskonzepte. Ob diese Konzepte für den Einsatz in WFMS geeignet sind, wird innerhalb der Arbeitspakete IPVR.CAM untersucht. Daher sollen an dieser Stelle nur einige Ziele und Aufgaben erwähnt werden, welche als transaktionale Aspekte von Workflows von Bedeutung sind.

Die Ausführung einzelner Teilschritte eines Workflows in einem weit verteilten System muß soweit geschützt werden, daß die gewünschte Ausführungssemantik des Workflows garantiert werden kann. Teilweise erfordert dies Maßnahmen von der Systemebene bis zur Anwendungsebene (z.B. Definition bestimmter transaktionaler Charakteristika) des Workflows.

- Konsistenzerhaltung
  Durch die Einzelschritte eines Workflows sollte die Ausführung jeweils von einem konsistenten Zustand in den anderen versetzt werden.
- Persistenz von Zwischenergebnissen
  Wenn auch teilweise Zwischenergebnisse wieder verworfen werden müssen oder Teilschritte rückgängig gemacht werden sollen (z.B. Fehlerbehandlung), so kann es dennoch erforderlich sein, die Dauerhaftigkeit bestimmter Zwischenergebnisse oder Zwischenzustände zu gewährleisten.
- Verhalten im Fehlerfall
  Sollte die Durchführung einer Aktivität oder eines Workflows scheitern, können verschiedene Strategien der Fehlerbehandlung gewählt werden. Diese können vom WFMS aufgrund der Fehlersemantik oder der Workflow-Parameter veranlaßt werden oder durch den Benut-

zer definiert worden sein. Mögliche Alternativen sind:
- Rollback (passives Zurücksetzen auf einen definierten Ausgangspunkt),
- Undo (aktives Aufheben/Kompensieren der bereits durchgeführten Aktionen),
- Redo (erneutes Durchführen der gescheiterten Aktionen),
- Roll-Forward (alternatives Fortsetzen der gescheiterten Aktionen),
- Kompensation (ersatzweise Durchführung anderer Aktionen)

oder auch Kombinationen dieser Möglichkeiten.

3.4 Anpassungsaspekte

Bisher wurden Vorgangssteuerungssysteme überwiegend in Bereichen eingesetzt, in denen die zu steuernden Vorgänge klar definiert und sehr gut strukturiert waren. Die Gegebenheiten in vielen Prozeßumgebungen erfordern jedoch sehr flexible und anpassungsfähige Systeme, die auch weniger strukturierte oder nicht im voraus definierte Prozesse unterstützen können.

Eine Maßnahme hierfür ist, die Workflows anpassungsfähig zu gestalten. Bezüglich dieser Anpassungsaspekte sollte aus einer Workflow-Beschreibung hervorgehen,
- welche Workflow-Parameter geändert werden dürfen,
- von wem diese geändert werden dürfen,
- zu welchem Zeitpunkt dies geschehen kann und ggfs.
- unter welchen Bedingungen diese Änderungen durchgeführt werden dürfen.

Im Arbeitspaket IPVR.AWS.2 wird speziell auf diese Aspekte eingegangen und untersucht, welche Anforderungen an adaptive WFMS zu stellen sind. In diesem Abschnitt sollen zunächst einige Anpassungsaspekte vorgestellt werden, soweit sich diese auf die Beschreibung der Workflows auswirken. Für die Beschreibung sind im wesentlichen drei Arten von Anpassungen relevant:

• Änderungen einzelner Workflow-Parameter zur Laufzeit
  Unter Umständen soll bei der Durchführung des Workflows von der ursprünglichen Definition abgewichen werden. Dies bedeutet, daß zur Laufzeit einzelne Änderungen der Workflows vorgenommen werden. Die Änderungen können sich auf verschiedene Aspekte des Workflows beziehen:
  - Veränderungen der Kontroll- oder Anwendungsdaten
  - Zuweisung an einen anderen (speziellen) Bearbeiter
  - Verändern der Prozeßumgebung (Anwendungsprogramme, Zugriffsrechte etc.)
• Änderungen der Workflow-Struktur
  Während die Änderung der einzelnen Parameter meist nur eine geringe und überschaubare Auswirkung hat, können Änderungen der Workflow-Struktur (verhaltensorientierte Aspekte) den weiteren Ablauf nachhaltig beeinflussen. Teilweise sollen auch hier nur einzelne Parameter wie die Definition eines Ereignisses oder die Vorbedingung eines Sub-

workflows geändert werden. Es sind jedoch auch größere Änderungen denkbar wie:
- Änderungen der Kontrollstruktur eines Subworkflows (z.B. Reihenfolge einer Sequenz)
- Übergehen oder Nichtausführen eines Subworkflows
- Ersetzten eines Subworkflows durch einen anderen

- Erweiterungen eines Workflows um Subworkflows
  Während der Bearbeitung eines Workflows können weitere Maßnahmen offensichtlich werden, die sich aus der Bearbeitung ergeben (z.B. die Rückversicherung bei einem Vorgesetzten - als Einfügen einer weiteren Aktivität). Werden diese Maßnahmen außerhalb der Kontrollsphäre des WFMS durchgeführt, besteht keinerlei Zusammenhang zu diesem. Wird hingegen ein zusätzlicher Subworkflow in die Beschreibung eingefügt, werden die Maßnahmen voll in die weitere Bearbeitung integriert.
  Wie auch bei den anderen Änderungen kann dies teilweise erwünscht sein oder bestimmten Personen erlaubt sein; andere Workflows (oder Teile davon) hingegen dürfen nicht geändert werden, um eine bestimmte Ausführung zu garantieren. Welche Änderungen möglich sind und welche nicht, muß der Beschreibung der Anpassungsaspekte entnommen werden können.

Da alle drei Teilaspekte Rechte auf bestimmte Teile des Workflows beschreiben, könnten z.B. Metarechte eines Workflows definiert werden, die über die Änderungsmöglichkeiten einzelner Anwender Aussagen zulassen. Über solche Klassen könnten auch die Rechte zur Definition oder zum Start von Workflows geregelt werden.

3.5 Management-Aspekte

Die Informationsfunktion speziell des Managements stellt eine wichtige Aufgabe dar, die durch WFMS in hohem Maße unterstützt werden kann. Bisher sind meist isolierte Projektmanagement-Systeme oder Management-Informationssysteme im Einsatz, deren Einsatz sehr aufwendig ist. Ein Großteil der relevanten Informationen kann aus einem WFMS gewonnen und zur Verfügung gestellt werden.
Durch die Beschreibung verschiedener Teilaspekte kann diese Sicht in Workflows integriert werden.

Protokollierung bzw. Historie

In der Regel wird die Historie von Prozeßausführungen im Fehlerfall für Recovery-Maßnahmen verwendet. In WFMS kann diese Historie jedoch auch sinnvoll zur Steuerung der Workflows verwendet werden, falls die vergangenheitsbasierten Daten in die Steuerung integriert werden (Bsp. Rückfrage - vgl. Arbeitspaket IPVR.MWM.5). Weitere Verwendungsmöglichkeiten sind der Aufbau eines Vorgangsinformationssystems *VIS* (aktuelle Vorgangsinformationen zur Laufzeit) oder eines Management-Informationssystems *MIS* für die langfristige Verarbeitung von Prozeßinformationen (Erhebung, Analyse, Reengineering).

Dabei kann ein Teil der Vorgangsinformationen standardmäßig vom System protokolliert werden, um diese für die oben genannten Zwecke bereitzustellen. Darüber hinaus soll der Anwender weitere Informationen bestimmen können, die in der Historie erfaßt und zu einem späteren Zeitpunkt als Workflow-Kontrolldaten wieder abgerufen werden können. Die Workflow-Beschreibung muß also beinhalten,

- Welche Informationen protokolliert werden sollen,
- Wofür dies geschieht (Steuerung, VIS, MIS),
- Wie diese Informationen ermittelt werden können und
- Wo bzw. wie sie protokolliert werden.

Zeitliche Aspekte

Zeitliche Aspekte sind nicht nur zur Steuerung (Timeouts für Subworkflows oder Aktivitäten), sondern auch zur Kontrolle (Informationen über Verzögerungen etc.) von großer praktischer Bedeutung für die Durchführung von Workflows.

Für die Steuerung ist es notwendig, daß die Beschreibung Informationen über geplante Durchführungszeiten von Workflows (Analyse, Simulation) und über maximal erlaubte Durchführungsdauer (Timeouts) erlaubt.
Mit diesen Informationen können zum einen Erkenntnisse über die zu erwartenden Durchführungszeiten bzw. etwaige Schwachstellen gewonnen werden. Zum anderen können Workflows nach Ablauf der Timeouts abgebrochen und entsprechende Fehlerbehandlungsmaßnahmen eingeleitet werden.

Für die Kontrolle der Durchführungszeiten müssen
- der späteste Anfangstermin,
- die kritische Durchführungsdauer (evtl. mehrstufig) und
- durchzuführende Aktionen (bei Überschreitung) beschrieben werden.
So kann z.B. definiert werden, daß eine Benachrichtigung an den Projektleiter geschickt werden soll, falls eine Aktivität nach einer bestimmten Zeit noch nicht beendet wurde oder falls ein Subworkflow bis zu einem bestimmten Termin nicht begonnen wurde.

Die Beschreibung der Termine sollte sowohl absolut als auch relativ (zu einem anderen definierten Zeitpunkt) möglich sein. Die Aktionen könnten mit bestimmten Bedingungen verknüpft werden, welche das Auslösen oder die Auswahl der Aktionen bestimmen.

Controlling

Ein effektives Controlling kann neben der Kontrolle zeitlicher Aspekte die Berücksichtigung weiterer Geschehnisse innerhalb eines Workflows erfordern.

Auch in diesem Fall wird wohl die Benachrichtigung eines Rollenträgers eine der wichtigsten Aktionen sein, obgleich auch andere Reaktionen vorstellbar sind.

Ursachen, aufgrund derer bestimmte Aktionen ausgelöst werden sollen, können sein:

- Eintreten externer Ereignisse von besonderer Bedeutung
- Eintreten interner Ereignisse
  z.B. Start, erfolgreiche Beendigung oder Abbruch (erfolglos) eines Workflows
- Workflow-Parameter erfüllen bestimmte vordefinierte Bedingungen
  z.B. falls die Bearbeitung eines Auftrages mit besonders hoher Priorität gestartet wird.

Bei einem umfangreichen Einsatz dieses Aspektes kann es vorteilhaft sein, diese nicht einzeln, sondern über ein bestimmtes Schema ausführen zu lassen. So könnten z.B. Workflows oder Ereignisse in Kategorien eingeteilt werden, wobei das Verhalten während der Durchführung durch diese Kategorien bestimmt würde.

3.6 Sicherheitsaspekte

In dieser Sicht werden die Aspekte Datenschutz, Datensicherheit und die lokale Verteilung von Daten und Workflows beschrieben.
Einerseits sollen die einzelnen Teile eines Workflows auf verschiedenen Rechnern verteilt ausgeführt werden können, andererseits besteht gerade bei sehr sensitiven Prozessen ein großes Bedürfnis nach Datenschutz und Datensicherheit. Dieser Konflikt in der Zielsetzung tritt selten innerhalb eines gemeinsamen Workflows auf; ein flexibles WFMS sollte jedoch in der Lage sein, beiden Anforderungen getrennt gerecht zu werden.

- Sicherheit

Sowohl die Ausführung von Aktivitäten, Subworkflows oder Workflows als auch die Kommunikation im Rahmen der Workflows kann durch unterschiedliche Maßnahmen gesichert werden. Über ein abgestuftes Konzept könnten die Workflow je nach Einteilung durch unterschiedliche Transaktionen Verfahren gesichert werden. Auch die Kommunikation kann je nach Sicherheitsstufe durch unterschiedliche Protokolle oder Verfahren geschützt werden.

- Datenschutz

Ähnlich wie bei der Sicherheit können auch die Workflow-Daten (Kontroll- und/oder Applikationsdaten) in bestimmte Klassen eingeteilt werden. Abhängig von dieser Einteilung können die Daten z.B. verschlüsselt gespeichert oder übertragen werden.

- explizite Verteilung

In der Regel ist die Transparenz der Verteilung eine wesentliche Anforderung an WFMS (vgl. Arbeitsbereich TE). Der Benutzer sollte sich nicht mit der expliziten Verteilung beschäftigen müssen, was sich sowohl auf die verteilte Ausführung der Workflows (mehrere verteilte Workflow-Engines) als auch auf die Verteilung der Workflow-Daten bezieht.

Doch gerade Sicherheitsinteressen können ein direktes Eingreifen auf die lokale Verteilung erfordern. Bestimmte Workflows oder Daten sollen z.B. nur auf einem speziellen Rechner ausgeführt bzw. gehalten werden. Die Verteilung dieses Workflows und der notwendigen Daten auf einen anderen Rechner muß also unterbunden werden. Die dafür notwendigen Informationen müssen in der Workflow-Beschreibung enthalten sein.

Die verschiedenen Teilaspekte müssen nicht unbedingt getrennt, sondern können auch in einem gemeinsamen Modell beschrieben werden.

4. Anforderungen an eine allgemeine Workflow-Beschreibung

In diesem vierten Kapitel werden Anforderungen an eine Sprache zur Beschreibung von Workflows vorgestellt.
Im ersten Unterkapitel werden die wesentlichen nichtfunktionalen Anforderungen beschrieben, wobei diese in allgemeine Anforderungen und spezielle Anforderungen an eine interne bzw. externe Repräsentation der Beschreibung gegliedert sind.
Anschließend werden die funktionalen Anforderungen an die Beschreibung präsentiert. Diese wurden aus den in Kapitel 2 und Kapitel 3 beschriebenen Aspekten abgeleitet; zur besseren Übersicht wird eine kurze, tabellarische Darstellungsform gewählt. Da sich insbesondere aus der Beschreibung der verhaltensbezogenen Aspekte wichtige Anforderungen ergeben, werden diese im zweiten Unterkapitel gesondert behandelt, während die weiteren Aspekte im letzten Unterkapitel zusammengefaßt sind.

Als Ausgangspunkt für die Analyse diente ein umfangreiches Literaturstudium, wobei Beschreibungen und Produkte verschiedener Anwendungsgebiete berücksichtigt wurden. Die Anforderungen wurden also nicht aus einem konkreten Umfeld oder Szenario abgeleitet, sondern so allgemein wie möglich gehalten, um eine gute Grundlage für mögliche Spezialisierungen zu bilden. Sie können zum Teil konfliktär sein und deshalb oder aus Gründen der Komplexität für einen späteren Entwurf wieder eingeschränkt werden. Anforderungen, die sich aus dem speziellen Anwendungsumfeld der öffentlichen Verwaltung ergeben, werden bei der Bearbeitung des Arbeitspaketes IPVR.AWS.2 berücksichtigt.

Die Anforderungen werden nach dem Grad ihrer Bedeutung für ein flexibles WFMS klassifiziert und den Kategorien MUST, SHOULD und WANT zugeordnet.

Die Gesamtheit der Informationen, die zur Ausführung eines Workflows notwendig ist, läßt sich in zwei Arten unterteilen. Im allgemeinen Teil der Beschreibung werden Informationen dargestellt, die für alle Workflows gültig sind und während der Durchführung extrahiert werden können. Organisatorisches Wissen zum Beispiel kann allgemeingültig definiert und von allen Workflows verwendet werden (Rollen). Im Gegensatz dazu sind im spezifischen Teil der Beschreibung diejenigen Informationen enthalten, die von Workflow zu Workflow unterschiedlich sein können und sollen (z.B. Spezifikation einer Rolle, die zur Bearbeitung erforderlich ist). Nach der Diskussion aller Aspekte im vorigen Kapitel, soll der Schwerpunkt jetzt auf der Workflow-spezifischen Beschreibung liegen, die natürlich von der Struktur und den Inhalten der allgemeinen Beschreibung abhängt.

4.1 Nichtfunktionale Anforderungen

4.1.1 Allgemeine

Vollständigkeit (NF1) *MUST*

Die Beschreibung eines Workflows sollte zusammen mit den allgemeinen Beschreibungsteilen (z.B. Organisationsmodell) alle Informationen enthalten, die zur Ausführung und Kontrolle des Workflows erforderlich sind. Eine Ausnahme bilden Workflows, bei denen bestimmte Informationen erst zur Laufzeit ergänzt werden können (automatisch oder durch Benutzereingabe). Umfang und Genauigkeit der Beschreibung sind u.a. von der gewünschten Funktionalität des WFMS stark abhängig.

Mächtigkeit (NF2) *SHOULD*

Die WF-Definition sollte mächtig genug sein, um eine Vielzahl unterschiedlicher und komplexer Vorgänge beschreiben zu können und nicht auf die Beschreibung eines speziellen Typs von Prozessen beschränkt sein (vgl. funktionale Anforderungen).

Eindeutigkeit und Konsistenz (NF3) *SHOULD*

Auch wenn die Beschreibung aus einzelnen, sich ergänzenden Komponenten besteht, muß sie in ihrer Gesamtheit eindeutig und konsistent sein.

Verfeinerung (NF4) *SHOULD*

Workflows sollen auf unterschiedlichen Abstraktionsebenen beschrieben werden können (um z.B. zur Laufzeit verfeinert oder ergänzt zu werden). Es müssen sowohl High-Level-Prozesse (zur individuellen Ausführung) als auch sehr detaillierte, kleinere Prozesse (evtl. zur automatischen Ausführung) beschrieben werden können.

Änderbarkeit (NF5) *SHOULD*

Die Workflow-Beschreibung soll zur Laufzeit änderbar sein. Dies beinhaltet Änderungen und Erweiterungen von Seiten des Systems und durch den Benutzer.

Anpassungsfähigkeit (NF6) *WANT*

Zur Steuerung komplexer Vorgänge ist u.a. erforderlich, daß die Prozeßbeschreibung flexibel und anpassungsfähig gestaltet werden kann. Die hierfür notwendigen Methoden (Vererbung, Parametrisierung etc.) sollten daher in der Workflow-Beschreibung berücksichtigt werden.

Vergangenheitsbezug (NF7) *WANT*

Neben den aktuellen Informationen zur Steuerung des weiteren Ablaufes sollen auch Informationen über den bereits durchgeführten Teil des Workflows in der Beschreibung enthalten sein. Damit ist zu jedem Zeitpunkt der gesamte Prozeß darstellbar.

4.1.2 Definition und externe Repräsentation

Überführbarkeit (NF8) *MUST*

Die WF-Definition sollte so beschaffen sein, daß sie entweder direkt ausführbar ist oder automatisch in eine interne Beschreibung überführt werden kann. Dadurch entsteht eine interne Repräsentation, die von der Workflow-Engine ausgeführt werden kann.

Verständlichkeit (NF9) *SHOULD*

Die Beschreibungssprache zur Definition eines Workflows mit allen notwendigen Zusatzinformationen sollte so einfach wie möglich sein. Die Definition kann erleichtert werden durch:

- Möglichkeit zur graphischen Definition mit übersichtlicher Darstellung
- WF-Definition in einzelnen unabhängigen Sichten (vgl. modulare Aspekte).
  Hilfreich sind eine gute Benutzerführung, Hilfen und automatische Prüfungen (Konsistenz, Plausibilität, Vollständigkeit).
- Eignung für verschiedene Benutzergruppen. Insbesondere sollten (zumindest einfache) Anpassungen auch zur Laufzeit vom Anwender vorgenommen werden können.

Modularität (NF10) *SHOULD*

Die Definition und die externe Repräsentation sollte in modulare Komponenten unterteilt werden können, die jeweils spezielle Aspekte der Workflows widerspiegeln (vgl. auch Kapitel 1.3).

Prüfbarkeit (NF11) *SHOULD*

Auch die externe Repräsentation sollte, die für die Definition wichtigsten Prüfungen (z.B. Konsistenz oder Plausibilität) erlauben.

Benutzerfreundlichkeit - Wiederverwendung (NF12) *WANT*

Da die Definition von komplexen Prozessen sehr aufwendig sein kann, sollte auf Wiederverwendung geachtet werden. Über die Definition von Aktivitäten- und Prozeßklassen in einer Bibliothek könnte die Definition neuer Aktivitäten und Vorgänge stark erleichtert werden. Der Einsatz von Vererbung von Prozeßattributen kann hilfreich sein.

Benutzerfreundlichkeit - Default-Werte (NF13) *WANT*

Die Nutzung geeigneter Default-Wertebelegungen bei der WF-Definition stellt eine weitere Vereinfachung dar und bietet die Möglichkeit einfache Prozesse auch mit geringem Aufwand definieren zu können.

Benutzerfreundlichkeit - Wartbarkeit (NF14) *WANT*

Struktur, Aufbau und Inhalt der Prozeßbeschreibungen soll so gestaltet sein, daß Verwaltung und Wartung der Workflow-Beschreibungen unterstützt werden (vgl. Modularisierung der Beschreibung).

4.1.3 Interne Repräsentation

Ausführbarkeit (NF15) *MUST*
Die interne Repräsentation der Workflow-Beschreibung muß (nach der Initialisierung) automatisch ausgeführt werden können. Dies kann durch eine Interpretation der Beschreibung oder durch Compilation und Ausführung erfolgen.
Für eine Interpretation sprechen eine bessere Anpassungsfähigkeit und eine höhere Transparenz (Datenschutz/Datensicherheit) bei der Portierung von Subprozessen. Argumente, die häufig für eine Übersetzung sprechen, wie Performance und kontextabhängige Prüfungen, scheinen bei WFMS weniger relevant zu sein. In speziellen Teilen kann es jedoch Ausnahmen geben, wie bei der Verarbeitung komplexer, zusammengesetzter Ereignisse oder bei kontextabhängigen Trigger-Regeln.

Darstellbarkeit (NF16) *MUST*
Von der internen Repräsentation muß eine Abbildung auf die externe Repräsentation möglich sein, die alle relevanten Details (verlustfrei) enthält.

Analysierbarkeit und Meßbarkeit (NF17) *SHOULD*
Die Beschreibung sollte hinreichend formal sein, um notwendige Analysen (Deadlocks, Erreichbarkeit etc.) der Prozesse zu erlauben und erforderliche Prozeßattribute messen zu können.

4.2 Funktionale Anforderungen zur Beschreibung der verhaltensorientierten Aspekte

F1: *Darstellung elementarer Kontrollkonstrukte* *MUST*

- F1.1: Vorgänger-Nachfolger-Beziehung zwischen Workflows
- F1.2: Start bzw. Abbruch eines Workflows
- F1.3: Optionale Ausführung eines Workflows
- F1.4: Synchronisation von Workflows mit Ereignissen

F2: *Steuerung des Kontrollflusses durch* *MUST*

- F2.1: einfache Kontrollstrukturen, z.B. Sequenz, bedingte und unbedingte Verzweigung, sequentielle und parallele Schleifen. (M)
- F2.2: kompakte Kontrollstrukturen
  Reihe, m aus n, bedingte Auswahl (S)
- F2.3: deskriptive Kontrollstrukturen
  z.B. Limitierung, Verzögerung, Existenzabhängigkeit, Ausschluß (S)

F3: *Darstellung unterschiedlicher Ereignisse* *MUST*

- F3.1: Interne Ereignisse (M)
  z.B. Start, Ende, Abbruch, Unterbrechung eines Workflows
- F3.2: externe (benutzerdefinierte) Ereignisse (M)
- F3.3: zeitbezogene Ereignisse (M)
  z.B. absolute (Termine), relative und periodische Ereignisse
- F3.4: einfache und zusammengesetzte Ereignisse (S)

F4: *Verhalten bei Eintreten eines Ereignisses* *MUST*

- F4.1: Kopplung der Aktion an bestimmte Bedingungen (M)
- F4.2: Beschreibung der auszuführenden Aktion (S)

F5: *Auslösen externer Ereignisse* *SHOULD*

F6: *Einsatz von Bedingungen:* *MUST*

- F6.1: in Kontrollstrukturen (Verzweigung, Invarianten etc.)
- F6.2: über Workflow-Daten, Ereignisse

4.3 Funktionale Anforderungen zur Beschreibung der weiteren Aspekte

F7: *Dekomposition von Workflows (elementaren Subworkflows/Aktivitäten)* *SHOULD*

F8: *Spezifikation verschiedener Workflow-Parameter* *MUST*
Name, Beschreibung, Vor- und Nachbereitungsprozeduren etc.

F9: *Spezifikation von Verbindungen zu anderen Sichten* *MUST*

F10: *Darstellung der Kontrolldaten* *MUST*

- F10.1: Typ, Wert, Konsistenzbedingungen
- F10.2: Datenfluß zwischen Subworkflows (Art der Übergabe)

F11: *Darstellung der Anwendungsdaten* *MUST*

- F11.1: Referenzen
- F11.2: Ort der Entstehung
- F11.3: Aufbau und Struktur
- F11.4: Ort (Workflows) der Verwendung
- F11.5: Art der Verwendung (lesend, schreibend)
- F11.6: Darstellung der Zugriffsrechte

F12: Spezifikation organisatorischer Attribute *MUST*

F12.1: Aktoren (Personen oder Tools) und
Verantwortliche (automatisch oder per Definition) (M)

F12.2: Aktorenarten (S)
Einzelpersonen, Gruppen: gemeinsam, zur Auswahl

F12.3: Zuteilungsarten: mit/ohne Ablehnungsmöglichkeit (S)

F12.4: Zuteilungsstrategie (S)

F13: Spezifikation der Bearbeitungsart (automatisch/manuell) *SHOULD*

F14: Darstellung kausaler Abhängigkeiten eines Workflows von *SHOULD*

F14.1: Bedingungen

F14.2: Ereignissen

F14.3: organisatorischen Regeln

F15: Spezifikation des Verhaltens in Fehlerfällen *SHOULD*

F15.1: Beschreibung möglicher Fehlerfälle

F15.2: Beschreibung der entsprechender Aktionen

F16: Spezifikation persistenter Zwischenergebnisse *SHOULD*
oder der Art des transaktionalen Schutzes einer Workflow-Ausführung

F17: Verändern der Workflow-Beschreibung *SHOULD*

F17.1: Spezifikation der änderbaren bzw. nicht änderbaren Beschreibungsmerkmale

F17.2: Spezifikation von Bedingungen für diese Änderungen
(basierend auf Ereignissen, Parametern etc.)

F17.3: Spezifikation von Zugriffsrechten für die Änderungen

F18: Ergänzen der Workflow-Beschreibung *SHOULD*

F18.1: Sichtbarkeit der Workflow-Parameter, die verwendet werden können

F18.2: Vererbung von Parametern an die neu definierten Subworkflows

F19: Spezifikation des Protokolls *WANT*

F19.1: Standard-Protokoll (Was wird protokolliert?)

F19.2: Spezielles-Protokoll: Zeitpunkt, Inhalt und Zielort der Protokollierung

F20: Spezifikation zeitlicher Attribute *WANT*

F20.1: erwartete Durchführungsdauer

F20.2: Zeitliche Restriktionen für die Durchführung:
kritische/maximale Durchführungsdauer - spätester Anfangstermin

F20.3: Aktionen bei Nichteinhaltung

F21: Spezifikation von Controlling-Maßnahmen *WANT*

F21.1: Spezifikation kritischer Geschehnisse

F21.2: Spezifikation der auszulösenden Aktionen

F22: Spezifikation der Sicherheitsstufe zur Ausführung eines Workflows *WANT*

F23: Spezifikation der Schutzmaßnahmen für Workflow-Daten *WANT*

F24: Spezifikation von lokalen Restriktionen *WANT*

F24.1: für die lokale Ausführung von Workflows

F24.2: für die lokale Sicherung der Workflow-Daten

5. Bekannte Formalismen zur Workflow-Beschreibung

In diesem fünften Kapitel sollen nun einige Beschreibungsformalismen untersucht werden. Dabei werden jeweils Stärken und Schwächen untersucht bzw. überprüft, wie sich dieses Formalismen zur Beschreibung von Workflows eignen.
Nachdem im ersten Unterkapitel eine Übersicht bestehender Formalismen gegeben wird, werden drei Klassen weiter diskutiert: netzbasierte, programmiersprachenähnliche (prozedurale/ regelbasierte) und ereignisorientierte Beschreibungen.
Abschließend werden dann die Eigenschaften dieser Formalismen mit den oben ermittelten Anforderungen verglichen und bewertet.

5.1 Übersicht und Klassifikation von Prozeßmodellen

In den letzten Jahren entstand eine Vielzahl von Prozeßmodellen, die unterschiedliche Anforderungen zu erfüllen haben (vgl. auch Kapitel 1.2). In diesem Unterkapitel sollen diese nach gemeinsamen Ansätzen und Ausrichtungen zusammengefaßt und klassifiziert werden.

Ein Merkmal nach dem klassifiziert werden kann, ist die vorrangige Orientierung der Modelle, d.h. welche Objekte im Vordergrund des Interesses stehen. Hierbei können drei Arten von Modellen unterschieden werden: daten-, aktivitäten- und ereignisorientierte Modelle.
Bei den datenorientierten Modellen stehen die Daten und Informationen im Vordergrund. Sie werden an bestimmten Stellen im Prozeß erzeugt und verwendet, wobei das Eintreffen oder die Verfügbarkeit der Daten den Prozeßablauf steuert. Bei den aktivitätenorientierten Modellen bilden die Aktivitäten die zentrale Sicht der Prozesse. Notwendige Daten werden den einzelnen Aktivitäten zugeordnet und die Koordination der einzelnen Aktivitäten wird als wesentlicher Bestandteil des Modells durch den Kontrollfluß beschrieben. Bei der dritten Art stehen Ereignisse im Mittelpunkt des Modells, deren Auftreten dann bestimmte Aktionen auslöst und Aktivitäten synchronisiert.

Wie bereits erwähnt wurde, bietet sich zur Beschreibung der Workflows vor allem der aktivitätenorientierte Ansatz an, da die Steuerung und Kontrolle der einzelnen Aktivitäten die Hauptaufgabe von WFMS darstellt. Hierbei können zwei Arten der Ablaufkontrolle unterschieden werden: die präskriptive und die deskriptive Ablaufkontrolle [Jablon95b].
“Wenn sie die alternative Ausführungsreihenfolgen nur über die Auswertung von vorgegebenen Ablaufbedingungen zuläßt, spricht man von präskriptiver Ablaufkontrolle; wenn sie eine Menge äquivalenter Ausführungsreihenfolgen unterstützt, wird sie deskriptiv genannt”.

Weiter können Prozeßmodelle nach der Art der verwendeten Beschreibungsformalismen klassifiziert werden. In [CuKeOv92] werden die Beschreibungen eingeteilt in:
Functional Models, Plan Based Models, Petri-Net Models, Quantitative Models und Programming Models.
Eine ähnliche Unterscheidung der verwendeten Beschreibungsformalismen findet sich in [PenShu91]. Hier werden unterschieden:
- visual expression-oriented formalisms (SADT),
- graph/net analysis-oriented formalisms (Petri Nets, Zustandsübergang-Diagramme),
- process programming enforcement-oriented languages (APPL/A) und
- rule-based enforcement-oriented languages.

Diesen Ansätzen folgend werden in den weiteren Unterkapiteln einige Formalismen bzw. Modellklassen vorgestellt. Dabei werden sie kurz eingeführt und ihre wesentlichen Eigenschaften hinsichtlich der Eignung zur Beschreibung von Workflows diskutiert.
Bei der Klassifikation stand nicht die Vollständigkeit bzgl. einsetzbarer Formalismen im Vordergrund. Vielmehr sollten einige wesentliche Ansätze, die sich deutlich voneinander abgrenzen, unterschieden und diskutiert werden:

- netzbasierte Modelle
- prozedurale (programmiersprachenähnliche) Modelle
- regelbasierte (programmiersprachenähnliche) Modelle
- Ereignis-bzw. Trigger-orientierte Modelle

5.2 Netzbasierte Beschreibungen

Unter die graphen- und netzbasierten Modelle fallen einige wichtige Basisformalismen aber auch eine Vielzahl von Abwandlungen, Erweiterungen und speziell entwickelte Modelle.

Häufig werden sie eingesetzt, um dynamisches, nebenläufiges Verhalten zu beschreiben und zu analysieren. In den Basisformalismen (z.B. Petrinetzen) stehen meist nur eine sehr geringe Anzahl syntaktischer Elemente zur Verfügung, was den Sprachenumfang zwar sehr übersichtlich hält, die Darstellung komplexer Sachverhalte jedoch häufig aufwendig und unübersichtlich werden läßt. Daraufhin wurden verschiedene spezielle Netze (höhere Petrinetze) entwikkelt, welche in der Regel eine kompaktere Darstellungsform ermöglichen, ohne die Mächtigkeit der darstellbaren Vorgänge jedoch wesentlich zu erweitern (Prädikat-Transitionsnetze, gefärbte, zeitbezogene, stochastische Petrinetze). Da auch diese Standardnetze nicht immer geeignet sind, spezielle Gegebenheiten zu beschreiben, wurden sie in verschiedenen Projekten und Produkten spezialisiert und erweitert (RFA-Netze, Funsoft-Netze etc.).

Netze werden in verschiedenen Systemen der Softwareentwicklung, Geschäftsprozeßmodellierung und verschiedenen Vorgangssteuerungssystemen eingesetzt. Sie eignen sich besonders

zur Beschreibung von Koordination und Synchronisation von Aufgaben mit kontrollierter Vergabe von Ressourcen.
Durch ihre formale Darstellung lassen sie sich gut übersetzen bzw. maschinell ausführen, ermöglichen umfangreiche Analyse- und Simulationsmöglichkeiten und eignen sich daher gut für die interne Repräsentation einer Prozeßbeschreibung. Darüber hinaus lassen sich Netzbasierte Beschreibungen sehr gut graphisch darstellen und weisen damit eine hohe Pragmatik aus, die zur Verständlichkeit der Beschreibung erheblich beiträgt. Sie können daher auch zur externen Repräsentation der Beschreibung verwendet werden.

5.2.1 Petrinetze

Als einfaches Grund-Petrinetz kann das Stellen-/Transitions-Netz (S/T- Netz) bezeichnet werden. Es dient als Basis, auf der die höheren Netze aufbauen und wird durch einen 5-Tupel
N = (S, T; F, K, W) definiert. Berücksichtigt man darüber hinaus die Startmarkierung M_0, dann erhält man ein S/T-System als 6-Tupel. S/T-Netze sind wie die meisten Petrinetze bipartite Graphen mit Stellen S, Transitionen T, einer Flußrelation F, einer Kapazitätsfunktion K und einer Gewichtsfunktion W. Das bedeutet, daß die Stellen gemäß der Kapazitätsfunktion mehrere Marken enthalten können und Mehrfachkanten durch die Kantengewichte dargestellt werden.
Zum weiteren Studium der Petrinetze sei auf die entsprechende Fachliteratur verwiesen [BrReRo87], [Röbbec93], [Starke80], [Starke90]. Hier soll nur soweit auf die Theorie eingegangen werden, daß die unterschiedlichen Netze, deren Eigenschaften und Modellierungsmächtigkeit (bezogen auf Workflows) diskutiert werden können.

Wie bereits erwähnt, spielen die Analyse- und Berechenbarkeitseigenschaften der Petrinetze eine wichtige Rolle. Zu den elementaren Berechnungen auf Petrinetzen gehören:
- erreichbare Markierungen,
- aktivierte, aktivierbare, lebendige und tote Transitionen
- lebendige, tote und deadlock-freie Netze und
- Invarianten.

Aufgrund der Einfachheit der S/T-Netze kann die Modellierung von Prozessen mit S/T-Netzen zu sehr komplexen und unübersichtlichen Modellen führen. Um Sachverhalte kompakter darzustellen bzw. bestimmte Eigenschaften überhaupt erst modellierbar zu machen (Zeit), wurden daher Erweiterungen wie Verbotskanten, Netze mit integrierten Zeitbegriffen oder Transitionsprioritäten vorgenommen. Zwei dieser höheren Petrinetze werden im nächsten Abschnitt beschrieben.

Die Beschreibung von Vorgängen mit petrinetzähnlichen Modellen erfolgt in der Regel auf folgende Art und Weise:
- Aktivitäten werden auf Transitionen abgebildet,
- Zustände werden auf Stellen abgebildet und
- Marken beschreiben je nach Modell den Daten- oder Kontrollfluß des Vorgangs.

5.2.2 Höhere Petrinetze

Die höheren Petrinetze bauen auf den S/T-Netzen auf, wobei Zusatzinformationen über die Stellen, Marken oder Transitionen modelliert werden. Damit lassen sich Stellen oder Transitionen zusammenfassen, was zu der erwünschten kompakteren Darstellung führt. Grundsätzlich kann man folgende Eigenschaften höherer Netze unterscheiden:
- Darstellbarkeit verschiedener Marken (CP-Netze)
- Beschriftung der Netzelemente, z.B. Schaltbedingungen für Transitionen (Pr/T-Netze)
- Bewertung von Netzelementen, z.B. Prioritäten für Transitionen

Von den höheren Petrinetzen sind die gefärbten Netze (CP-Netze) und die Prädikat-/Transitions-Netze am weitesten verbreitet; auf sie soll im weiteren näher eingegangen werden. Sie unterscheiden sich somit hauptsächlich in ihrer Darstellungsform bzw. durch ihre unterschiedlich starke Eignung zur Modellierung bestimmter Eigenschaften.

Gefärbte Petrinetze

Gefärbte Petrinetze unterscheiden verschiedenartige Marken (Farben) in Stellen und Transitionen. Die Kanten zwischen Stellen und Transitionen können mit Ausdrücken beschriftet werden, die über die Schaltregel der Transitionen Auskunft geben. Dadurch wird vorgegeben, wieviele und welche Marken beim Feuern einer Transition von einer Stelle abgezogen bzw. einer Stelle zugeführt werden. Das relativ komplexe S/T-Netz zur Beschreibung der "speisenden Philosophen" ist weitgehend bekannt. Durch die Beschreibung mit einem CP-Netz vereinfacht sich diese Darstellung erheblich.

Prädikat-/Transitions-Netze

Prädikat-/Transitions-Netze unterscheiden nicht nur verschiedenartige Marken, sondern geben diesen Marken auch noch eine Struktur. Auf dieser Struktur kann man nun Funktionen bilden, die wiederum zur Steuerung der Transitionen genutzt werden können. Wann eine Transition feuert, wird durch ein Prädikat bestimmt, das als Wächter der Transition dient. Durch diese Verwendung von Variablen lassen sich viele Probleme übersichtlich modellieren, allerdings erschwert dies auch zum Teil die Analyse der Netze (z.B. Invariantenberechnung [Claus94]).
Pr/T-Netze haben Vorteile gegenüber CP-Netzen,
- wenn viele Variablen an einer Kante stehen
- bei stetigen oder unendlichen Sorten von Marken
- bei komplexen Wächtern an Transitionen (Funktionen der Transitionen)

Pr/T-Netze eignen sich durch die Berücksichtigung von Variablen und Prädikaten besonders gut zur Beschreibung von Vorgängen; Regeln zur Steuerung des Kontrollflusses lassen sich dadurch direkt in die Netze integrieren und erhöhen dadurch die Verständlichkeit der Beschreibung.

5.2.3 Spezielle Netze

Trotz dieser Erweiterungen durch die höheren Petrinetze ist die Modellierung bestimmter Eigenschaften oder Zusammenhänge von Prozessen häufig schwierig, weshalb eine Vielzahl spezieller Netze entworfen wurde. Zu den bekannteren Netzen sind die Role/Function/Action Nets (RFA-Netze) und die Funsoft-Netze zu zählen; letztere werden auch im Workflow-Bereich eingesetzt. Die Eigenschaften, Stärken und Schwächen dieser beiden Netztypen sollen im folgenden diskutiert werden.

RFA-Netze

RFA-Netze wurden entwickelt, um die Zusammenarbeit von Mensch und Maschinen bei der Vorgangsbearbeitung zu beschreiben. Dabei sollten die Vorgänge in unterschiedlichen Detaillierungsgraden beschrieben werden können.

Elementare Operationen werden in RFA-Netzen als Transitionen dargestellt, während es zwei Arten von Stellen gibt (Bearbeitungszustände und Objekte). Durch die Verbindung der Stellen und der Transitionen mit Pfeilen (ebenfalls mehrere Arten) werden Kontroll- und Datenfluß dargestellt. Aktionsnetze können auf unterschiedliche Arten verfeinert bzw. vergröbert werden (Stellen, Transitionen, Teilnetze), wodurch die Darstellung auf verschiedenen Ebenen ermöglicht wird [Oberqu87].
Durch Komposition lassen sich abgeschlossene Einheiten von Operationen und Zuständen bilden, die zusammen mit den jeweiligen lokalen Objekten die Funktionsnetze (eines einzelnen Aktors) konstituieren. Mit Hilfe dieser Funktionen werden dann letztendlich die Rollennetze dargestellt, wobei eine Rolle verschiedene Funktionen beinhalten und sich aus mehreren Unterrollen zusammensetzen kann.

RFA-Netze wurden entwickelt, um anwendungsorientierte Beschreibungen von Arbeitsumgebungen zu erzeugen, welche die Vorgänge aus der Sicht der einzelnen Bearbeiter beschreiben.
Der Zusammenhang dieser Einzelsichten ist über die feinste Netzebene - die Aktionsnetze - gegeben, wodurch RFA-Netze auch als interne Repräsentationen eingesetzt werden können, um die Systemsicht der Vorgänge zu beschreiben.
Da dies jedoch das Hauptziel der internen Repräsentation von Workflow-Beschreibungen ist, sollte der Schwerpunkt stärker auf der Beschreibung dieser Zusammenhänge liegen.

Funsoft-Netze

Funsoft-Netze [GruJeg92] wurden bereits in verschiedenen wissenschaftlichen und industriellen Projekten eingesetzt. Sie sind ebenfalls eine Abwandlung von Pr/T-Netzen, wobei elementare und zusammengesetzte Aktivitäten durch Transitionen dargestellt werden. Die Transitionen werden durch Eingangs- und Ausgangsregeln beschrieben, welche das Verhalten der Transition beim Feuern festlegen. Stellen repräsentieren Ansammlungen von Objekten, wobei es sich sowohl um komplexe (Dokumente) als auch um Workflow-relevante Anwendungsdaten (Ergebnis einer Entscheidung) handeln kann. Der Wertebereich der Stellen wird wie bei höheren Petrinetzen definiert.

Ähnlich wie bei RFA-Netzen werden verschiedene Typen von Kanten (Flußrelation) unterschieden. Die Datenflußkanten drücken lesenden (In) oder schreibenden (Out) Zugriff aus, wobei der Wert der entsprechenden Daten das Verhalten der Transition bestimmt. Im Gegensatz dazu sind die Werte der Kontrollflußkanten nicht von Bedeutung, sie steuern die Transitionen lediglich durch ihre Existenz.

5.3 Programmiersprachenähnliche Beschreibungen

In diesem Unterkapitel sollen programmiersprachenähnliche Beschreibungen vorgestellt und untersucht werden. Dabei werden prozedurale und regelbasierte Sprachen unterschieden; auf spezielle objektorientierte Ansätze wird hier nicht näher eingegangen (der Bericht wird evtl. zu einem späteren Zeitpunkt dahingehend erweitert).
Da die Möglichkeiten der prozeduralen Beschreibung doch sehr begrenzt sind wird der Schwerpunkt auf die regelbasierten und ereignisorientierten Beschreibungssprachen gesetzt.

5.3.1 Prozedurale Beschreibungsprachen

Eine Abstammungslinie der WFMS sind die operativen Anwendungssysteme. Der Kontrollfluß wurde in diesen Systemen häufig mit Hilfe der bekannten prozeduralen Kontrollkonstrukte definiert, durch die einzelnen Prozeduren gesteuert wurden. Somit lag es nahe, auch zur Kontrolle und Steuerung der Workflows eine prozedurale Beschreibung zu verwenden.

Zur Beschreibung einfacher Prozesse, die keinen besonderen Ansprüche an die Kontrollkonstrukte stellen, können prozedurale Sprachen gut verwendet werden. Sind die Vorgänge auch noch gut strukturiert und zum Zeitpunkt der Definition vollständig bekannt, eignet sich die prozedurale Beschreibung mit ihren präskriptiven Eigenschaften gut und ist auch leicht ausführbar.
Allerdings stoßen die Sprachen an ihre Grenzen, wenn es darum geht deskriptive Kontrollkonstrukte zu integrieren. Dies kann dann sinnvoll und notwendig sein, wenn das Verhalten von Prozessen, deren genauer Ablauf zum Zeitpunkt der Definition noch nicht bekannt ist, beschrieben werden soll. Auch zur Beschreibung sehr komplexe Prozesse, bei denen viele Abhängigkeiten zwischen den Subprozessen/Aktivitäten bestehen, können deskriptive Kontrollkonstrukte sehr hilfreich sein.

Darüber hinaus sind die Integration von transaktionalen Erweiterungen und die Anpassungsfähigkeit weitere wichtige Anforderungen an WFMS, die ebenfalls nicht für eine prozedurale Beschreibung sprechen.

5.3.2 Regelbasierte und ereignisorientierte Beschreibungen

Aus der Diskussion prozeduraler Beschreibungen lassen sich bereits einige Aussagen über die Eignung regelbasierter Beschreibungen ableiten. Bei Prozessen mit komplexen Abhängigkeiten der Aktivitäten eignen sich besonders regelbasierte Systeme. Workflows lassen sich u.U. nicht ausschließlich durch einfache präskriptive Kontrollkonstrukte steuern, da ihr Verhalten stark abhängig von verschiedenen externen und internen Einflüssen sein kann.
Aus diesen Gründen basieren viele Workflow-Beschreibungen auf regelbasierten, ereignisorientierten Beschreibungen, die die Ausführung eines Workflows durch interne und externe Abhängigkeiten (dependency rules) bestimmen.

Diese Regeln dienen also dazu, das Verhalten des Workflows zu bestimmen, indem sie Einschränkungen oder Bedingungen spezifizieren, wann bestimmte Aktivitäten gestartet, abgebrochen oder beendet werden (bzw. wann nicht); sie drücken somit gegenseitige Abhängigkeiten aus, die sich u.a. auf die Zustände der Aktivitäten beziehen. Das bedeutet jedoch, daß Kenntnisse über die einzelnen Zustände erforderlich sind, weshalb Zustandsänderungen häufig als interne Ereignisse dargestellt und propagiert werden. Darüber hinaus beeinflussen auch externe (benutzerdefinierte) Ereignisse den Ablauf der Workflows.
Die Regeln werden also so definiert, daß sie durch Aussagen über Ereignisse die Abhängigkeiten der Aktivitäten ausdrücken.

ECA-Rules

Die meisten dieser regelbasierten Sprachen bauen auf ECA-Rules (Event-Condition-Action-Rules) auf, wobei die Ereignisse die eigentlich steuernden Elemente der Beschreibungssprache darstellen. ECA-Rules bestehen aus drei Komponenten: Ereignissen (Events), Bedingungen (Conditions) und Aktionen (Actions). Tritt ein spezifiziertes Ereignis ein und sind bestimmte Bedingungen erfüllt, wird durch die ECA-Regel eine bestimmte Aktion angestoßen bzw. ausgeführt (z.B. ON <Ereignis>, IF <Condition>, DO <Action>).

Der Spezifikation der Ereignisse kommt dabei eine entscheidende Rolle zu. Ein Beispiel für eine Sprache zur Spezifikation von Ereignissen ist SNOOP [CharMis94]. SNOOP wurde zwar zur Beschreibung von Ereignissen in aktiven Datenbanken entworfen, die Konzepte können jedoch auf Workflow-relevante Ereignisse übertragen werden; folgende Unterscheidungen von Ereignissen erscheinen für WFMS sinnvoll:

- Aktivitäts- bzw. Datenorientierte Ereignisse
- Temporale Ereignisse (absolute, relative und periodische)
- Externe, vom Benutzer definierte Ereignisse und interne, vom System ausgelöste Ereignisse
- Einfache und zusammengesetzte Ereignisse
  Zur Bildung zusammengesetzter Ereignisse werden in SNOOP bsw. die Operatoren Disjunktion, Sequenz, Konjunktion, sowie der periodische und der aperiodische Operator verwendet.

Ansätze für Regelbasierte Systeme

Im folgenden sollen drei Ansätze für regelbasierte Systeme aus dem Bereich der erweiterten transaktionalen Konzepte vorgestellt werden.

• ACTA [Elmaga92]
Der Ansatz geht davon aus, daß Abläufe von Vorgängen durch Historien der aufgetretenen Ereignisse und Prädikate auf diesen Ereignissen gekennzeichnet sind. Beispielsweise kann eine Folgeaktivität B erst dann gestartet werden, wenn das Ereignis “committed (A)” der Vorgängeraktivität A in der Historie enthalten ist. ACTA beruht auf der Prädikatenlogik und verwendet zwei Arten von Ereignissen: Transaktionsereignisse und Objektereignisse. Zur Spezifikation werden Eigenschaften der Historie beschrieben, wobei die logische Implikation als Operator und zwei Basisprädikate eine besondere Rolle spielen. Mit dem Prädikat ($e \in H$) bzw. ($e \rightarrow e'$) wird ausgedrückt, daß ein Ereignis e in der Historie enthalten sein muß bzw. vor dem Ereignis e’ enthalten sein muß.
Mit diesen Mitteln lassen sich eine Anzahl verschiedener Abhängigkeiten ausdrücken, mit denen Abläufe von Vorgängen beschrieben werden können (Abort-Abhängigkeit, Commit-Abhängigkeit etc.).

• Einen weiteren Ansatz stellte Klein [Klein91] vor, bei dem die Regeln ebenfalls auf zweierlei Arten von Abhängigkeiten aufbauen. Die Ordnungsabhängigkeiten erfordern ähnlich wie bei dem oben vorgestellten ACTA-Prädikat eine bestimmte Reihenfolge der Ereignisse in der Historie; die zweite Art - die Existenzabhängigkeiten - schreiben vor, daß beim Eintreten eines Ereignisses e_1 gefordert wird, daß auch das Eintreten des Ereignisses e_2 gewährleistet ist (gleichgültig ob vor oder nach dem Auftreten von e_1).
Auf diesem Ansatz baut Jablonski auf und definiert eine Reihe komplexer Kontrollkonstrukte zur Spezifikation von Workflows, wobei er ebenfalls die zwei generellen Bedingungstypen, Zeitbedingungen (z.B. Limitierung) und Existenzbedingungen (z.B. Existenzabhängigkeit) verwendet, die “... sowohl hinreichend als auch notwendig sind um Protokolle zur Synchronisation von Workflows zu beschreiben” [Jablon95b].

• Auch der Ansatz von Günthör [Günthö93] stammt aus dem Bereich der erweiterten Transaktionsmodelle und baut auf den Überlegungen Kleins auf. Das Verhalten der Vorgänge wird ebenfalls deskriptiv über Abhängigkeitsregeln spezifiziert, wobei neben transaktionalen Ereignissen auch nicht-transaktionale Ereignisse (z.B. zeitbezogene) verwendet werden können. Aus diesen Regeln werden dann Formeln temporärer Logik erzeugt und ein Abhängigkeitsgraph erstellt, mit dessen Hilfe die Ausführung der Vorgänge kontrolliert werden kann. Durch die Abbildung auf die temporäre Logik können temporäre Abhängigkeiten durch spezielle Operatoren einfach dargestellt werden. So verwendet Günthör u.a. Operatoren wie:
- *sometimes* ... eine Regel wird in der Zukunft erfüllt werden,
- *always* ... eine Regel ist immer erfüllt,
- *never* ... eine Regel wird nie erfüllt werden,
- *before* ... wenn B irgendwann erfüllt wird, dann wird A vorher erfüllt werden.

5.4 Vergleich mit den Anforderungen

In der folgenden Übersicht wird dargestellt, inwieweit die verschiedenen Formalismen die in Kapitel 4 aufgestellten Anforderungen erfüllen. Da es dabei um die generelle Eignung der verschiedenen Klassen zur Workflow-Beschreibung geht, werden diese in ihrer Gesamtheit betrachtet und nicht die unterschiedlichen Systeme verglichen.
Jede Anforderung wird dabei mit einer Note (1-5) bewertet, die aussagt wie gut diese Anforderung erfüllt werden kann. Bei den nichtfunktionalen Anforderungen wurden nur diejenigen bewertet, bei denen die Bewertung (sehr gut, gut, befriedigend, mangelhaft, ungenügend) durch die verwendeten Formalismen beeinflußt wird. Bei den funktionalen Anforderungen gibt die Bewertung Aufschluß darüber, wie gut sich die entsprechenden Sachverhalte beschreiben lassen:
1 ... Formalismus ist sehr gut zur Darstellung geeignet
2 ... Sachverhalt kann dargestellt werden
3 ... Sachverhalt kann durch (mögliche) Erweiterungen dargestellt werden
4 ... Sachverhalt kann schlecht dargestellt werden
5 ... Sachverhalt kann nicht dargestellt werden.

Zur Beschreibung einiger Sachverhalte sind lediglich Verweise oder Zuordnungen erforderlich. Hierfür können verschiedene einfache Darstellungsformen gewählt werden, die unter dem Sammelbegriff “deklarative Formalismen” beurteilt werden.

Nr.	Nichtfunktionale Anforderungen	proze-dural	netz basiert	regel basiert
NF2	Mächtigkeit	4	3	2
NF5	Änderbarkeit	4	2	3
NF7	Vergangenheitsbezug	2	3	3
NF8	Überführbarkeit	1	2	2
NF9	Verständlichkeit	2	2	4
NF11	Prüfbarkeit	3	2	2
NF15	Ausführbarkeit	1	2	2
NF16	Darstellbarkeit	4	2	4
NF17	Analysierbarkeit und Meßbarkeit	2	2	3

Nr.	Funktionale Anforderungen	proze-dural	netz basiert	regel basiert	dekla-rativ
F1	Elementare Kontrollkonstrukte	2	2	2	-
F1.2	Start bzw. Abbruch	2	2	2	-
F1.3	Optionale Ausführung	2	2	2	-
F1.4	Synchronisation mit Ereignissen	3	1	1	-
F2.1	einfache Kontrollstrukturen	1	1	1	-
F2.2	kompakte Kontrollstrukturen	4	3	2	-
F2.3	deskriptive Kontrollstrukturen	5	4	2	-
F3	Darstellung von Ereignissen	3	2	2	-
F4	Verhalten auf Ereignisse	3	2	1	-
F5	Auslösen externer Ereignisse	2	2	2	-
F6	Einsatz von Bedingungen	1	2	2	-
F7	Dekomposition von Workflows	2	3	4	-
F8	Spezifikation von Parametern	2	3	3	+
F9	Verbindungen zu anderen Sichten	4	3	3	+
F10	Spezifikation der Kontrolldaten	2	3	3	+
F11	Spezifikation der Anwendungsdaten	2	3	3	+
F12	Spezifikation organisator. Aspekte	3	3	3	+
F13	Spezifikation der Bearbeitungsart	3	3	3	+
F14	Darstellung kausaler Abhängigkeiten	4	2	1	+
F15	Verhaltens in Fehlerfällen	3	3	2	+
F16	Persistente Zwischenergebnisse	3	3	3	+
F17	Änderungen der Beschreibung	3	3	3	+
F18	Ergänzen der Workflow-Beschreibung	1	1	3	-
F19	Spezifikation des Protokolls	3	3	3	+
F20	Spezifikation zeitlicher Attribute	2	2	2	+
F21	Spez. von Controlling-Maßnahmen	3	3	3	+
F22	Sicherheitsstufen/Schutzmaßnahmen	3	3	3	+
F24	Spezifikation lokaler Restriktionen	4	3	3	+

6. Zusammenfassung und Ausblick

Abschließend sollen noch kurz einige bestehende Systeme und die dort verwendeten Notationen betrachtet werden. Wie nach der Analyse bereits zu erwarten war, werden in aktuellen Forschungsprojekten vor allem netzbasierte (auf Pr/T-Netzen aufbauende) Notationen und regelbasierte Beschreibungen eingesetzt:

- Am Institut für Informatik der Universität Klagenfurt wurde das Workflow Activity Model (WAMO) und eine eigene Definitionssprache WADL (Workflow Activity Definition Language) entwickelt. Die Beschreibung ist stark aktivitätsorientiert und erlaubt eine programmiersprachenähnliche Definition von Workflows. Die Daten- und Kontrollflußsemantik von WADL kann formal auf der Basis von Abhängigkeitsregeln (in Anlehnung an ACTA) definiert werden. Die Realisierung erfolgt ebenfalls regelbasiert, auf der Basis aktiver Datenbanken [EdeLie95].
- Berthold Reinwald entwickelte an der Universität Erlangen-Nürnberg das System ActMan (Activity Management), das ebenfalls auf der Dekomposition von Aktivitäten beruht. Wesentlicher Bestandteil der Beschreibung sind Aktivitätennetze mit Warteschlangen und Abhängigkeiten. Die Ausführung der Aktivitäten erfolgt durch Übersetzung und Aktivierung von Triggern in Form von ECA-Rules [ReiWed92], [Reinwa93].
- An der Universität Twente entwickelte Stef Joosten ein Trigger-Modell zur Definition von Workflows. Diese können später in Petrinetze transformiert und ausgeführt [Jooste94] werden.
- Am IPVR der Universität Stuttgart wird im Rahmen des Projekts APRICOTS (A PRototype Implementation of a COnTract System) ebenfalls ein Prototyp entwickelt, mit dem Basismechanismen für WFMS umgesetzt werden. Dabei wird zur Beschreibung transaktional geschützter Vorgänge, ein erweitertes Pr/T-Netz verwendet [ReuSch95].

Betrachtet man nun rückblickend die Beschreibungsformalismen, so stellt man fest, daß sie Stärken und Schwächen in verschiedenen Bereichen haben.

Die Stärken der netzbasierten Systeme liegen in ihrer Verständlichkeit und ihren Analysemöglichkeiten. Der Ablauf eines Vorgangs und die damit verbundenen Entscheidungen sind einfach nachzuvollziehen und eignen sich somit auch für die externe Repräsentation. Präskriptive Kontrollkonstrukte können gut beschrieben werden, wohingegen die Darstellung deskriptiver Konstrukte Schwierigkeiten bereitet. Die Erweiterungen der höheren Netze bieten die Möglichkeiten, auch komplexe Prozesse, durch die Spezifikation von Variablen und Prädikaten, übersichtlich darzustellen. Wesentlich hierfür ist jedoch die Dekomposition in Teilnetze.

Die regelbasierten Systeme hingegen zeichnen sich durch ihre Ausdrucksmächtigkeit aus, auch kompakte und deskriptive Kontrollstrukturen einfach beschreiben zu können. Weitere

Stärken sind die Orientierung an Ereignissen und die Darstellung komplexer Abhängigkeiten. Die Schwächen der regelbasierten Notationen liegen in der geringeren Verständlichkeit und Übersichtlichkeit der Darstellung. So kann es relativ schwierig sein die Durchführung eines Workflows und die damit verbundenen Teilentscheidungen nachzuvollziehen und in einer externen Repräsentation darzustellen.

Prozedurale Beschreibungen sind zwar einfach auszuführen und leicht nachzuvollziehen, in der Darstellung vielfältiger Kontrollstrukturen und Abhängigkeiten stoßen sie jedoch ebenfalls an ihre Grenzen. Auch bzgl. der Darstellbarkeit und der Änderbarkeit weisen sie Nachteile gegenüber den anderen Formalismen auf.

Nachdem in den ersten Kapiteln die Notwendigkeit einer modularen Beschreibung nachgewiesen wurde, liegt der Schluß nahe, eine Workflow-Beschreibung zu wählen, deren Modularisierung den oben genannten Anforderungen entspricht. In diesem Falle könnten die einzelnen Aspekte durchaus in unterschiedlichen Notationen beschrieben werden, wobei die jeweiligen Stärken der Formalismen genutzt werden sollten.
Neben der Analyse von Anpassungsfähigkeit in Workflows soll dieser Ansatz in der weiteren Arbeit verfolgt werden.

7. Literaturverweise

[BrReRo87] Brauer, W.; Reisig W.; Rozenberg G. : Petri Nets: Applications and Relationship to other Models of Concurrence: Lecture Notes in Computer Science, Proceedings of an Advanced Course II - Bad Honnef, 1986. Springer: Berlin 1987.

[ChaMis94] Chakravarthy, S.; Mishra, D. : Snoop: An Expressive Event Specification Language for Active Databases. In Data & Knowledge Engineering 14 (1994), Nr. 1, pages 1-26.

[Claus94] Claus, Volker : Theorie paralleler Prozesse: Petrinetze I. Internes Skript zur Vorlesung an der Universität Stuttgart. Stuttgart, 1994.

[CuKeOv92] Curtis, Bill; Keller, Marc I.; Over, Jim : Process Modeling. In Communications of the ACM (1992), Band 35, Nr. 9, S. 75-90.

[Duitsh94] Duitshof, Matthijs : Workflow Automation in Three Administrative Organisations: Highlighting Business Process Modeling. Master thesis at University of Twente (Department of Computer Science - Information Systems). Enschede, 1994.

[EdeLie95] Eder, Johann; Liebhart Walter : A Transaction-Oriented Workflow Activity Model. In Proc. 3rd Int. Conf. of Cooperative Information Systems (CoopIS). IEEE, 1995.

[Elmaga92] Elmagarmid, Ahmed K. : Database Transaction Models for Advanced Applications. Morgan Kaufmann, 1992.

[FiKrNu94]]Finkelstein, Anthony; Kramer, Jeff; Nuseibeh, Bashar (Hrsg.) : Software Process Modelling and Technolgy. John Wiley & Sons, 1994

[FeSiEl93] Ferstl, Otto K.; Sinz, Elmar J. : Geschäftsprozeßmodellierung. In Wirtschaftsinformatik 35 (1993) 6, S. 589-592.

[Günthö93] Günthör, Roger : Extended Transaction Processing Based on Dependency Rules. In Proc. RIDE (1993) - Research Issues in Data Engineering and Interoperability in Multidatabase Systems.

[Jablon95a] Jablonski, Stefan : Workflow-Management-Systeme: Modellierung und Architektur. Thomson's Aktuelle Tutorien, Band 9. Berlin: Thomson, 1995.

[Jablon95b] Jablonski, Stefan : Workflow-Management-Systeme: Motivation, Modellierung, Architektur. In Informatik Spektrum 18 (1995), pages 13-24.

[GruJeg92] Gruhn, Volker; Jegelka, Rüdiger : An Evaluation of FUNSOFT Nets. In Proc. EWSPT 1992.

[Jooste94] Joosten, Stef : Trigger Modelling for Workflow Analysis, In G. Chroust, A. Benczur, Proceedings CON '94 : Workflow Management, Challenges, Paradigms and Products, pp. 236-247. Vienna: Oldenbourg, 1994.

[JosWWW95] Joosten, Stef : Definition of the Term Workflow Management. http://wwwis.cs.utwente.nl: 8080/~joosten/workflow.html#Def

[KelRom91] Kellner, Marc I.; Rombach, Dieter H. : Comparison of Software Process Descriptions - Session Summary. In Proc. 6th Int. Software Process Workshop (ISPW), pp. 7-18. New York, IEEE, 1991.

[Klein91] Klein, Johannes : Advanced Rule Driven Transaction Management. In 36th IEEE Computer Society Int. Conference (CompCon) Spring 1991, Digest of Papers, S. 562-567, 1991.

[KopMay95] Koppenberger, Dietmar; Mayer, Renate : Geschäftsprozeßmanagement : Untersuchung existierender Systeme. Interner Bericht (PoliFlow) am Institut für Arbeitswissenschaft und Technologiemanagement. Universität Stuttgart, März 1995.

[Leinga95] Leingang, Eduard : Workflow in der Entwicklung von Fahrsicherheitssystemen. Diplomarbeit Nr. 1231 am IPVR der Universität Stuttgart, 1995.

[Loncha93] Lonchamp, J. : A Structured Conceptual and Terminological Framework for Software Process Engineering. In Proc. 2nd Int. Conf. on the Software Process (ICSP-2), S. 41-53.. Los Alamitos: IEEE, 1993. S. 41-53.

[LotRom93] Lott, C.M.; Rombach, Dieter H. : Measurement-based guidance of software projects using explicit project plans. In nformation and Software Technology, 35 (1993), Nr. 6/7, S. 407-419.

[MarRom89] Marc, Leo; Rombach, Dieter H. : Software Process and Product Specifications: A Basis for Generating Customized SE Information Bases. In Proc. 22nd Hawaii Int. Conf. on System Science (HICSS-22), .S. 165-174. IEEE, 1989.

[Oberqu87] Oberquelle, Horst : Human Machine Interaction and Role / Function / Action Nets. In Brauer, W.; Reisig W.; Rozenberg G. : Petri Nets: Applications and Relationship to Other Models of Concurrence: Lecture Notes in Computer Science, Proceedings of an Advanced Course II - Bad Honnef, 1986. Springer: Berlin 1987.

[PenShu91] Penado, Maria H.; Shu, Christine : Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work. In Software Engineering Journal (UK), 6 (1991), S. 259-274.

[Reinwa93] Reinwald, Berthold : Workflow-Management in verteilten Systemen. Entwurf und Betrieb geregelter arbeitsteiliger Anwendungssysteme. Teubner. Texte zur Informatik : Stuttgart, 1993.

[ReiWed92] Reinwald, Berthold; Wedekind, Hartmut : Integrierte Aktivitäten- und Datenverwaltung zur systemgestützten Kontroll- und Datenflußsteuerung. In Informatik - Forschung und Entwicklung, 7 (1992), Heft 2, S. 73-82.

[ReuSch95] Reuter, Andreas; Schwenkreis, Friedemann : ConTracts - A Low-Level Mechanism for Building General-Purpose Workflow Management-Systems. In IEEE Bull. on the Tech. Com. on Data Engineering, 18 (1995), Nr. 1, S. 4-10.

[Röbbec93] Röbbecke, Michael : Petri-Netze in der Wissensverarbeitung. Informatik-Bericht 93/8 der TU Clausthal.

[Sch_ea94] Schuster H.; Jablonski S.; Kirsche T.; Bussler C. : A Client/Server Architecture for Distributed Workflow Management Systems. In Proceedings of the 3rd International Conference on Parallel and Distributed Information Systems. : Austin-Texas. September 1994.

[Starke80] Starke, Peter H. : Petri-Netze : Grundlagen, Anwendungen, Theorie. Berlin : Dt. Verlag der Wissenschaften, 1980.

[Starke90] Starke, Peter H. : Analyse von Petri-Netz-Modellen. Stuttgart: Teubner, 1990

[WFMC94] WFMC (Hrsg.) : Glossary. A Workflow Management Coalition Specification. WFMC: Brüssel, November 1994.