

Arbeitsberichte des Instituts für Wirtschaftsinformatik

Herausgeber: Prof. Dr. J. Becker, Prof. Dr. H. L. Grob,
Prof. Dr. U. Müller-Funk, Prof. Dr. G. Vossen

Arbeitsbericht Nr. 48

**Der Lösungsbeitrag von Metadatenmodellen
beim Vergleich von Workflowmanagementsystemen**

Michael Rosemann, Michael zur Mühlen

Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster,
Grevener Str. 91, 48159 Münster, Tel. (0251) 83-9751, Fax (0251) 83-9754

Juni 1996

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Zusammenfassung	3
1 Grundlagen des Workflowmanagement	3
1.1 Merkmale von Workflowmanagementsystemen	3
1.2 Marktüberblick	6
2 Definition und Intention der Metamodellierung	7
2.1 Merkmale und Ausprägungen von Metamodellen	7
2.2 Metamodelle für Workflowmanagementsysteme	8
2.3 Beschreibung der verwendeten Notation	9
3 Vergleich zweier Metadatenmodelle	10
3.1 Konfliktidentifikation und -auflösung	10
3.2 Generelle Ansatzpunkte eines Metadatenmodellvergleichs	10
3.3 Spezifikation der betrachteten Systeme	12
3.4 Vergleich der Organisationssicht	13
3.4.1 Die Organisationssicht von WorkParty	13
3.4.2 Die Organisationssicht von FlowMark	16
3.4.3 Vergleich	18
3.5 Vergleich der Prozeßsicht	20
3.5.1 Die Prozeßsicht von WorkParty	20
3.5.2 Die Prozeßsicht von FlowMark	21
3.5.3 Vergleich	26
4 Résumé und Forschungspotential	28
Literatur	30

Zusammenfassung

Die DV-gestützte Prozeßabwicklung mittels Workflowmanagementsystemen erlaubt die Auslagerung der Ablaufsteuerung aus Anwendungssystemen. Mittlerweile ist eine Vielzahl an Systemen mit unterschiedlicher Entwicklungsgeschichte (Dokumentenmanagement, Vorgangsteuerung etc.) verfügbar. Damit stellt sich das Problem der Auswahl eines bedarfsgerechten Workflowmanagementsystems. Im Rahmen dieses Arbeitsberichts soll der diesbezügliche Lösungsbeitrag des Vergleichs der Metadatenmodelle von Workflowmanagementsystemen evaluiert werden. Nach allgemeinen Ausführungen zur Metamodellierung werden hierzu exemplarisch die Metadatenmodelle für die Workflowmanagementsysteme WorkParty (SNI) und FlowMark (IBM) skizziert und verglichen.

1 Grundlagen des Workflowmanagement

1.1 Merkmale von Workflowmanagementsystemen

Vergleichbar mit der Auslagerung der Datenhaltung auf Datenbankmanagementsysteme kann der mit Workflowmanagementsystemen möglichen *Auslagerung der Ablaufsteuerung* (Abbildung 1) eine wesentliche Bedeutung für die grundsätzliche Architektur zukünftiger Informationssysteme zugesprochen werden.

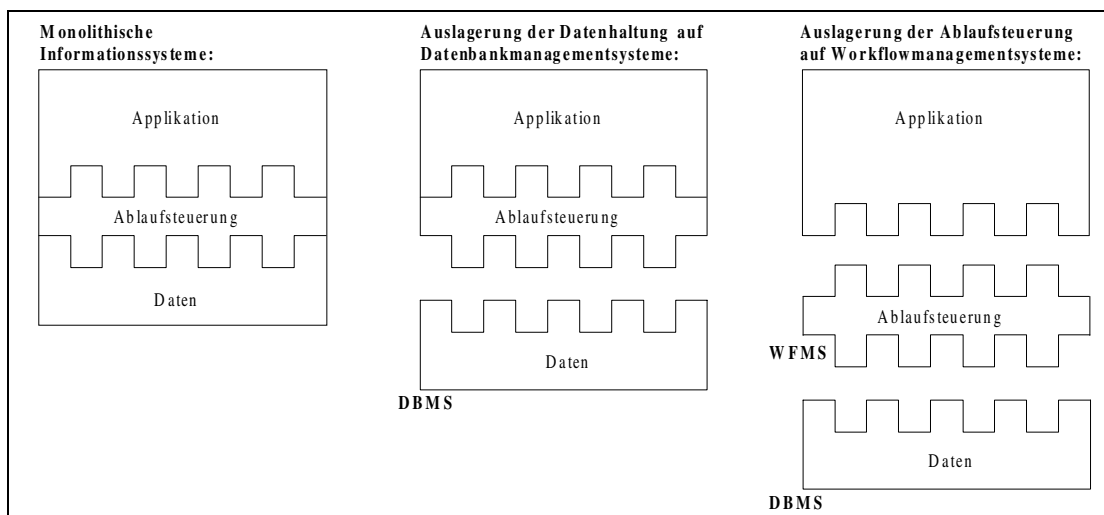


Abb. 1: Prognostizierte Entwicklung des Aufbaus von Informationssystemen

Ein *Workflowmanagementsystem* wird von JOOSTEN - in Anlehnung an die Definition der Workflow Management Coalition (WfMC)¹ - definiert als

„[...] a system that defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the business process logic.“²

*Workflows*³ wiederum sind Prozesse - zeitlich-sachlogische Abfolgen von Funktionen, die zur Bearbeitung eines betriebswirtschaftlich relevanten Objekts notwendig sind -, deren Funktionsübergänge automatisiert sind, d. h. Prozesse, deren Ablauflogik sich innerhalb der Kontrollsphäre eines Informationssystems befindet.⁴ Jedem Workflow liegt ein um die besonderen Anforderungen der informationstechnischen Realisierung⁵ angereichertes Prozeßmodell, das *Workflowmodell*, zugrunde.⁶ Workflowmodelle werden im Regelfall durch gerichtete Graphen beschrieben, deren Knoten (elementare bzw. komposite) Funktionen repräsentieren. Aus Sicht des Workflowmanagementsystems sind elementare Funktionen gekapselt, d. h. das Workflowmanagementsystem besitzt auf die automatisierte oder nicht automatisierte Funktionsdurchführung nur eine Außensicht.⁷

Die Architektur von Workflowmanagementsystemen konstituiert sich gemeinhin durch eine Zweiteilung in eine Build- und in eine Runtime-Komponente.⁸ Zur Buildtime (Modellierungskomponente) gehören im wesentlichen die Workflowmodellierung, die Spezifikation der Datenstrukturen und der aufzurufenden Applikationen, die Animation und die Simulation. Ferner ist die Beschreibung der Aufbauorganisation (Organisationseinheiten, Rollen, Stellen, Kompetenzen, Mitarbeiter u. ä.) zur Buildtime zu zählen. Mit der Transformation der Modelle in instanziierbare Workflowmodelle erfolgt der Übergang von der Build- in die Runtime. Zu letzterer zählen insbesondere die Instanziierung der Workflows, die Rollenauflösung⁹ und die Verteilung

1 Vgl. Workflow Management Coalition (1994), S. 39.

2 Joosten (1995), S. 4.

3 Zur Vielfältigkeit der Interpretation des Workflowbegriffs vgl. Georgakopoulos, Hornick, Sheth (1995), S. 122f.

4 Nach Jablonski, Stein (1995), S. 95f., stellen Workflowmanagementsysteme allgemein „eine Ausführungsinfrastruktur für Geschäftsprozesse zur Verfügung“, die zur weitgehenden Prozeßautomatisierung beitragen soll. A. M. Heilmann (1994), S. 9: „Workflow setzt Computereinsatz nicht zwingend voraus.“

5 Beispielsweise die Spezifikation der Datenstrukturen oder die Lokalisierung der Applikationen.

6 Zum Zusammenhang von Prozeß- und Workflowmodellierung vgl. u. a. Amberg (1996).

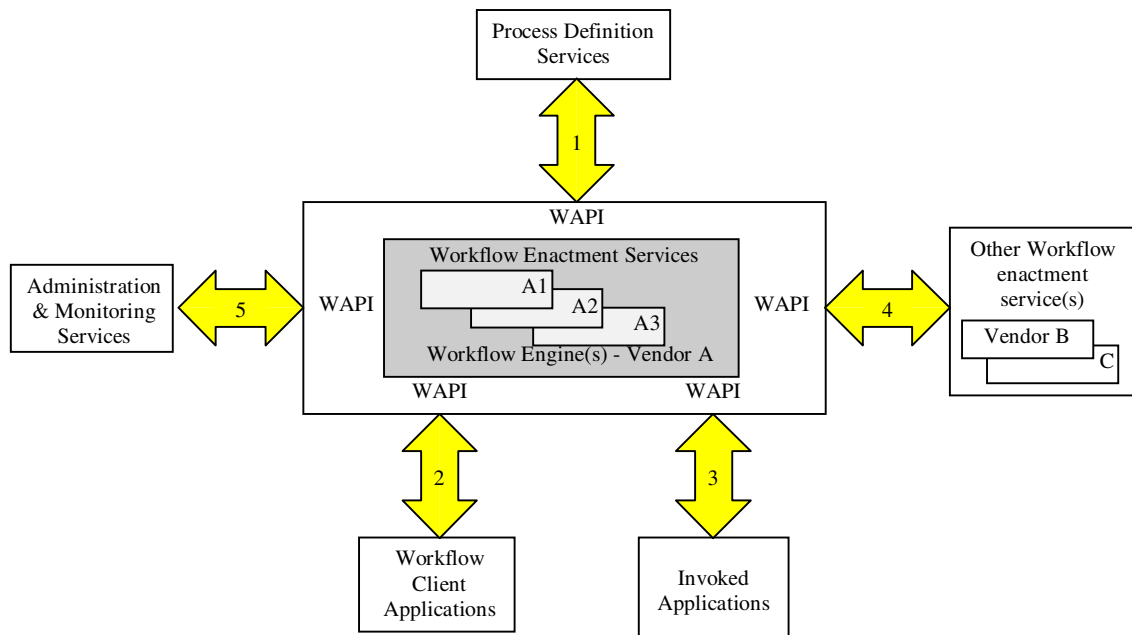
7 Vgl. Reinwald (1993), S. 37.

8 Vgl. Leymann, Altenhuber (1994); Jablonski (1995), S. 66-70.

9 Im Rahmen der Rollenauflösung werden die einer Funktion zugeordneten Qualifikationsanforderungen an den Aufgabenträger interpretiert und geeignete Mitarbeiter identifiziert.

anstehender Aufgaben (Workitems) an die betroffenen Mitarbeiter sowie das Monitoring der Prozeßdurchführung.

Eine Referenzarchitektur, die insbesondere die Interoperabilitätsschnittstellen eines Workflowmanagementsystems beschreibt, bildet das von der Workflow Management Coalition vorgestellte Referenzmodell (Abbildung 2).



Quelle: Workflow Management Coalition (1994), S. 5.

Abb. 2: Referenzmodell der Workflow Management Coalition

Die *Workflow Enactment Services* bilden den herstellerindividuellen Kern eines Workflowmanagementsystems, der die zentralen Funktionen wie Datenbankschnittstellen und Prozeßrouting zur Verfügung stellt. Sie kommunizieren über die *Workflow Application Programming Interfaces* (WAPI) 1-5 mit den einzelnen Services. *Process Definition Services* dienen der Prozeßmodellierung. Für diese Aufgaben werden oftmals externe Werkzeuge verwendet, da diese Perspektiven wie der Organisationsgestaltung oder der Prozeßkostenrechnung im Regelfall besser gerecht werden als die Modellierungskomponenten innerhalb von Workflowmanagementsystemen. Über die *Workflow Client Applications* werden die Systemnutzer u. a. über die auszuführenden Aufgaben informiert. Die *Invoked Applications* beinhalten die jeweilige Anwendungslogik. *Administration und Monitoring Services* verdichten Laufzeitdaten und unterstützen durch deren (graphische) Aufbereitung den Prozeßmanager bei der Beurteilung der Systemperformance. Ferner sieht das Referenzmodell eine Schnittstelle zwischen unterschiedlichen Workflowmanagementsystemen vor.

1.2 Marktüberblick

Mittlerweile ist eine Vielzahl an Workflowmanagementsystemen am Markt verfügbar,¹⁰ die sich aufgrund ihrer Entstehungsgeschichte und ihrer Intention hinsichtlich ihrer Leistungsfähigkeit deutlich unterscheiden. Eine mögliche Positionierung von ausgewählten Workflowmanagementsystemen ist in Abbildung 3 enthalten.¹¹

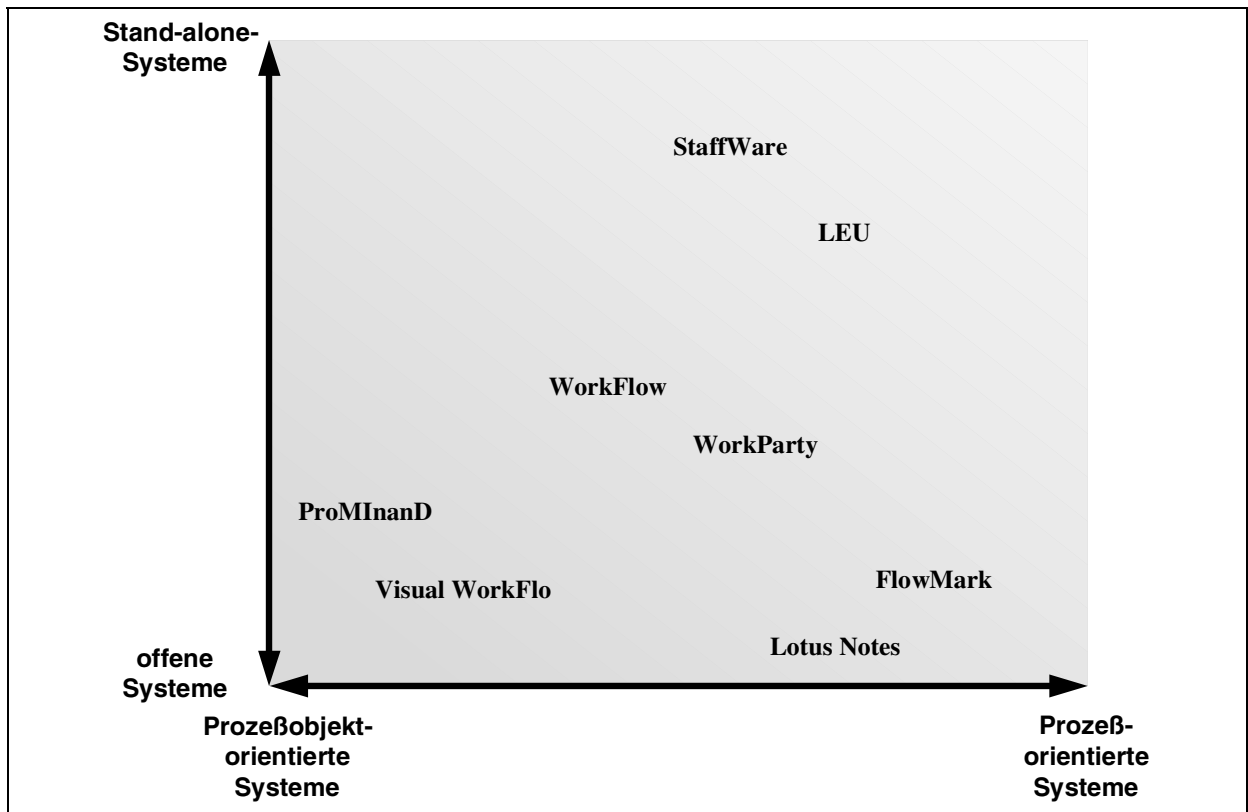


Abb. 3: Positionierung ausgewählter Workflowmanagementsysteme

Das Einsatzfeld von Workflowmanagementsystemen wird gemeinhin bei Prozessen gesehen, in denen sowohl die einzelnen Funktionen als auch der gesamte Prozeß routinisiert abgearbeitet werden können, d. h. in denen ein hoher Strukturierungsgrad vorliegt.¹²

Die Auswahl eines bedarfsgerechten Workflowmanagementsystems aus der quantitativ und qualitativ vielfältigen Produktpalette bedingt ein effizientes Auswahlverfahren. Im folgenden soll

¹⁰ Zu Marktübersichten vgl. beispielsweise Erdl, Schoenecker (1992); Kock, Rehäuser, Krcmar (1995); Schulze, Böhm (1996).

¹¹ Dabei wird zwischen offenen und Stand-alone-Systemen differenziert, sowie zwischen prozeßorientierten Systemen, deren Schwerpunkt in der Ablaufmodellierung liegt, und prozeßobjektorientierten Systemen, bei denen das Bearbeitungsobjekt (Dokument, Datei etc.) im Mittelpunkt der Betrachtung steht.

¹² Vgl. Picot, Rohrbach (1995), S. 32.

analysiert werden, welchen Lösungsbeitrag Metadatenmodelle bei der Selektion von Workflowmanagementsystemen besitzen. Hierzu werden zunächst Metamodelle allgemein und Metadatenmodelle im besonderen erläutert (Kapitel 2). Anschließend werden exemplarisch die produktindividuellen Metadatenmodelle der Systeme WorkParty (SNI) und FlowMark (IBM) vorgestellt (Kapitel 3). Ein Vergleich dieser beiden Modelle gestattet bereits wesentliche Aussagen über diese beiden Systeme. Der Lösungsbeitrag dieses Vorgehens wird abschließend kritisch beurteilt (Kapitel 4).

2 Definition und Intention der Metamodellierung

2.1 Merkmale und Ausprägungen von Metamodellen

Ein Modell ist ein immaterielles Abbild eines Realweltausschnitts für die Zwecke eines Subjekts.¹³ Ein Modell setzt damit zwei Systeme miteinander in Beziehung und besteht folglich aus drei Komponenten:¹⁴

- Das *Objektsystem* (S_O) repräsentiert die subjektive Interpretation eines zweckorientiert gewählten Realweltausschnitts (sog. Diskurswelt¹⁵) inklusive des relevanten Umweltausschnitts.
- Das *Modellsystem* (S_M) stellt das subjektive Abbild des Objektsystems dar. Zur Erstellung des Modellsystems bedarf es einer Syntax (auch: Notation, Sprache).
- Durch die *Abbildungsrelation* (auch: *Modellabbildung*) wird die Beziehung, welche das Objektsystem auf das Modellsystem abbildet, formuliert. Dabei erfolgt durch die *Elimination* irrelevanter Sachverhalte eine Reduktion der real vorhandenen Komplexität, die sich in der Elemente- und Beziehungsanzahl begründet. Durch die *Typisierung* relevanter Sachverhalte wird darüber hinaus die Verschiedenartigkeit der realen Sachverhalte verringert.

Wird ein Modellsystem M1 als Objektsystem eines Modellsystems M2 aufgefaßt, repräsentiert das Modellsystem M2 bezüglich des dem Modellsystems M1 zugrundeliegenden Objektsystems ein *Metamodellsystem*.¹⁶ Aufgrund dieses Abstraktionsgrads kann ein Metamodell auch als „ein

¹³ Vgl. Becker, Rosemann, Schütte (1995), S. 466.

¹⁴ Vgl. Guntram (1985), S. 312f.; Hars (1994), S. 7-11.

¹⁵ Zum Begriff und der Bedeutung der Diskurswelt vgl. ausführlicher Sinz, Popp (1992), S. 2-4. Vgl. auch Ferstl, Sinz (1995), S. 211.

¹⁶ Vgl. Hars (1994), S. 11. Vgl. auch Nissen et al. (1996), S. 38. Hier wird eine vereinfachende Sprachregelung verwendet, wonach mit Modell bzw. Metamodell im folgenden jeweils das Modellsystem bzw. das Metamodellsystem gemeint ist.

Gestaltungsrahmen verstanden werden, der die verfügbaren Arten von Modellbausteinen [...] und Beziehungen zwischen Modellbausteinen zusammen mit ihrer Semantik festlegt sowie Regeln für die Verwendung und Verfeinerung von Modellbausteinen und Beziehungen definiert.”¹⁷

Im Kontext der Informationsmodellierung lassen sich Metamodelle, die eine Notation, und Metamodelle, die eine Vorgehensweise spezifizieren, unterscheiden. Zur Beschreibung einer für die Informationsmodellierung verwendbaren Notation dienen *Metadatenmodelle*, zur Erläuterung der Vorgehensweise bei der Modellierung *Metaprozeßmodelle*.¹⁸ Ein Datenmodell eines Prozeßmodells ist damit ein Metadatenmodell; ein Prozeßmodell, das die Datenmodellierung beschreibt, ein Metaprozeßmodell. Jedes Metamodell wird seinerseits durch ein Metamodell beschrieben, wobei die Notation der beiden Metamodelle identisch sein kann. So kann beispielsweise die Notation der Entity-Relationship-Modellierung (ERM) mit einem ERM beschrieben werden. Ein Metamodell, das wiederum ein Modell über ein Metamodell darstellt, wird als *Meta-Metamodell* bezeichnet. Meta-Metadatenmodelle für Notationen zur Prozeßbeschreibung enthalten beispielsweise Entitytypen wie Knoten und Kante. Werden die Gemeinsamkeiten mehrerer Metamodelle in einem allgemeingültigen Modell konsolidiert, das darüber hinaus eine hohe semantische Qualität für sich in Anspruch nimmt, handelt es sich bei diesem Modell um ein *Referenz-Metamodell*. Ein solches Modell trägt insbesondere zur sprachlichen Normierung und zur Methodologienbildung bei.

2.2 Metamodelle für Workflowmanagementsysteme

In jüngster Zeit sind zahlreiche Metamodelle für Workflowmanagementsysteme publiziert worden, wobei es sich zumeist um Referenz-Metamodelle handelt.¹⁹

Ein Metamodell, das die wesentlichen Merkmale von Workflowmanagementsystemen wiedergeben soll, wird von JABLONSKI (1995) vorgestellt. JABLONSKI unterscheidet dabei sachliche (funktionale, operationale, verhaltensbezogene, informationsbezogene, organisatorische und kausale) sowie technische (historische, transaktionale) Aspekte.²⁰ Zur semiformalen Darstellung

¹⁷ Ferstl, Sinz (1994), S. 86. Vgl. auch Steele, Zaslavsky (1994), S. 317: "Meta models [...] might be expressed using one or more modelling techniques, that in combination are able to adequately model all relevant aspects of any given modelling technique."

¹⁸ Vgl. dazu auch Strahinger (Metamodell) (1995), S. 10ff.

¹⁹ Darüber hinaus gibt es mittlerweile eine Vielzahl an Metamodellen zur Prozeßmodellierung, auf die hier nicht weiter eingegangen wird. Vgl. u. a. Richter et al. (1995).

²⁰ Weitere Aspekte finden sich in Jablonski (1996), S. 79.

seines Metamodells verwendet er ein Beziehungsdiagramm sowie eine der Backus-Naur-Form²¹ verwandte Notation.

DERUNGS ET AL. (1995) - Mitarbeiter des St. Gallener Kompetenzzentrums 'Prozeß- und Systemintegration' - präsentieren ein Metadatenmodell, das aus vier Perspektiven (Workflow, Organisation, Integration, Datentransfer) besteht. Jeder Metaentitätstyp wird zusätzlich durch eine Beschreibung, durch Attribute sowie durch Beziehungen näher charakterisiert.

Zentrale Motivation des Ansatzes von JOOSTEN (1995) ist es, einen Beitrag zur Konzeptualisierung des Workflowmanagement zu leisten. Sein Ziel ist der Entwurf einer „generic architecture of a workflow management support system“²², die als Framework der Positionierung unterschiedlicher Konzepte dienen soll. Er unterteilt sein Metamodell in eine organisatorische und eine prozessuale Sicht.

Angelehnt an diese und weitere Arbeiten stellt GALLER (1995) ein Metamodell des Workflowmanagements vor, dessen Perspektiven den vier Sichten der Architektur integrierter Informationssysteme - Daten, Funktionen, Prozesse und Organisation - entsprechen.

Ziel dieses Arbeitsberichts ist zunächst nicht die Vorstellung eines weiteren Referenz-Metamodells, sondern die Erörterung *produktindividueller Metamodelle* sowie die Diskussion der mit dem Vergleich dieser (detaillierteren) Metamodelle verbundenen Erkenntnisse. Erst in einem nächsten Arbeitsschritt werden die im folgenden skizzierten sowie weitere produktindividuelle Metamodelle zu einem - insbesondere induktiv gewonnenen - Referenz-Metamodell konsolidiert. Dieses Metamodell soll u. a. als konzeptionelle Grundlage des Fachkonzepts eines Prozeßinformationssystems, das die Aufgaben des Prozeßmonitoring und -controlling unterstützt, dienen.

2.3 Beschreibung der verwendeten Notation

In den folgenden Ausführungen werden Metadatenmodelle in der Notation erweiterter Entity-Relationship-Modelle verwendet.²³ Eine besondere Anreicherung stellt hier die Verwendung *variabilisierter Integritätsbedingungen* dar.²⁴ Diese erlauben die instanzindividuelle Abbildung von semantischen Beziehungen (z. B. Antivalenzen) zwischen Kardinalitäten. Hierzu werden die betroffenen Kardinalitäten durch eine Variable ersetzt und in modellseparaten Integritätsbe-

²¹ Vgl. zur Notation der Backus-Naur-Form z. B. Appelrath, Ludwig (1992), S. 54ff.

²² Joosten (1995), S. 3.

²³ Vgl. Chen (1976). Zu den verwendeten Erweiterungen vgl. Scheer (1995), S. 31-47.

²⁴ Vgl. Rosemann (1995), S. 121f.; Strahinger (ERM) (1995), S. 3ff.

dingungen zueinander in Beziehung gesetzt. Weiterhin werden ausschließlich aus Gründen der Klarheit Generalisierungen gebildet, die beispielsweise alle für die Rollenauflösung relevanten Informationsobjekte zusammenfassen.

3 Vergleich zweier Metadatenmodelle

3.1 Konfliktidentifikation und -auflösung

Bevor zwei (Meta-)Datenmodelle miteinander verglichen werden können, sind etwaige Konflikte zwischen den Modellen zu identifizieren und aufzulösen. Hierbei sind insbesondere folgende drei Konflikttypen zu unterscheiden:²⁵

- Namenskonflikte

In dieser Phase gilt es, etwaige Sprachdefekte (Synonyme, Homonyme) zu identifizieren und aufzulösen. Während Homonyme (identische Begriffe für unterschiedliche Realweltobjekte) zu fehlerhaften Erkenntnissen führen können, erschweren Synonyme (unterschiedliche Begriffe für identische Realweltobjekte) die Identifikation von Korrespondenzen.

- Typkonflikte

Typkonflikte entstehen durch die (syntaktisch und semantisch richtige) Verwendung unterschiedlicher methodischer Konzepte zur Abbildung eines Sachverhalts und somit aufgrund der gegebenen Modellierungsfreiräume einer Notation.

- Strukturkonflikte

Strukturkonflikte liegen vor, wenn die zu integrierenden Modelle semantische Widersprüche zueinander aufweisen. Stärker noch als Typkonflikte bedingt die Auflösung von Strukturkonflikten deshalb den Einbezug eines fachkompetenten Mitarbeiters.

Typ- und Strukturkonflikte müssen insbesondere dann berücksichtigt werden, wenn unterschiedliche Personen an der Erstellung der Metamodelle beteiligt sind. Dies ist in der vorliegenden Arbeit jedoch nicht der Fall. Daher ist für die folgenden Ausführungen nur die Identifikation und Auflösung von Namenskonflikten relevant. Synonym- und Homonymkonflikte zwischen den herangezogenen Systemen (WorkParty und FlowMark) werden identifiziert und durch Umbenennung aufgelöst.

²⁵ Vgl. Hars (1994), S. 194-206.

3.2 Generelle Ansatzpunkte eines Metadatenmodellvergleichs

Bei der Gegenüberstellung zweier konfliktbereinigter Metadatenmodelle können folgende Informationsobjekte miteinander verglichen werden:

Entitytypen

Der Vergleich der Anzahl und Art der Entitytypen liefert die wesentlichsten Informationen bei der Gegenüberstellung von Metadatenmodellen. Je mehr Entitytypen das Metadatenmodell eines Workflowmanagementsystems bei gleichbleibendem Abstraktionsgrad aufweist, desto individueller ist das System an unterschiedliche Bedürfnisse anpaßbar. Die Aussagekraft des quantitativen und qualitativen Vergleichs der Entitytypen wird wesentlich durch den Detaillierungsgrad der Modelle determiniert. So bedingt beispielsweise der Vergleich der jeweils vorhandenen Kontrollflußkonstrukte, daß in beiden Fällen eine Spezialisierung des Entitytyps Kontrollflußkonstrukt erfolgt.

Relationshiptypen

Die zwischen den Entitytypen bestehenden Beziehungstypen geben ebenfalls Hinweise über die mögliche Flexibilität des Workflowmanagementsystems, die bei gleicher Anzahl und Art der Entitytypen mit steigender Anzahl an Beziehungstypen tendenziell zunimmt. Relationshiptypen unterscheiden sich in Metadatenmodellen von Workflowmanagementsystemen beispielsweise in der Organisationssicht durch unterschiedliche Beziehungstypen, die zwischen Mitarbeiter, Stellen, Rollen, Organisationseinheiten etc. bestehen.

Kardinalitäten

Aus den Kardinalitäten - vorrangig den Maximalkardinalitäten - kann auf die Ausgestaltungsvielfalt der möglichen Beziehungen geschlossen werden. Ein Workflowmanagementsystem, das beispielsweise bei der Strukturbeziehung, welche die Aufbauorganisation beschreibt, lediglich eine (0,1)-(0,n)-Beziehung erlaubt (d. h. nur streng hierarchische Formen der Aufbauorganisation zuläßt), ist weniger flexibel als ein System, das eine (0,n)-(0,n)-Beziehung gestattet (d. h. mehrdimensionale Organisationsformen ermöglicht, z. B. Projekt- oder Matrixorganisation). Durch die Aufnahme variabilisierter Integritätsbedingungen wird die Abbildungsgenauigkeit erhöht, der Modellvergleich jedoch erschwert.

Attribute

Über die einem ERM gemeinhin zu entnehmenden Konstrukte hinaus gibt ein Attributvergleich weitere Informationen über die hinterlegbaren und nutzbaren Informationen. Die Attribute sind danach zu unterscheiden, ob sie vom System interpretiert werden oder rein informatorischen

Wert besitzen. Benutzerdefinierbare Attribute²⁶ erlauben eine höhere Flexibilität als vollständig vorgegebene Attribute. Im folgenden wird angesichts der ansonsten zu weit führenden Detailliertheit der Ausführungen auf einen umfassenden Attributvergleich verzichtet. Exemplarisch werden die Attribute, die in WorkParty bzw. FlowMark an einem Mitarbeiter hängen, verglichen.

3.3 Spezifikation der betrachteten Systeme

WorkParty

WorkParty ist ein erstmals 1992 vorgestelltes Workflowmanagementsystem der Siemens Nixdorf AG.²⁷ Es basiert auf einer Client-Server-Architektur, bei der die Dienste zur Verwaltung und Weiterleitung von Vorgängen²⁸ sowie die Anbindung einer Datenbank auf einem zentralen System bereitgestellt werden, während auf den Clients die für den Benutzer sichtbare Entwicklungs- bzw. Laufzeitumgebung verfügbar ist.²⁹ Die häufig anzutreffende Aufteilung in eine Entwicklungs- und eine Laufzeitkomponente (Buildtime und Runtime)³⁰ findet bei WorkParty ihren Niederschlag in den Komponenten *Organisatortool* und *Laufzeitsystem*.³¹ Im Gegensatz zu anderen Workflowmanagementsystemen erfolgt im Organisatortool lediglich die Ablaufmodellierung. Die Organisationsmodellierung wird mittels des externen Programms *ORM* (Organisations- und Ressourcenmanagement) durchgeführt, dessen Daten sowohl der Entwicklungs- als auch der Laufzeitkomponente durch eine SQL-Schnittstelle zur Verfügung stehen.³² Änderungen der Unternehmensstruktur können zur Laufzeit im ORM implementiert werden und führen nicht zu einer Unterbrechung des Produktivbetriebs. Der Übergang vom Entwicklungs- zum Laufzeitsystem erfolgt durch die Freigabe des erstellten Prozeßmodells zur Instanziierung. Das Laufzeitsystem ist in die SNI ComfoDesk-Shell integriert und stellt den Endbenutzern die Worklist mit den zu bearbeitenden Vorgängen zur Verfügung.

FlowMark

FlowMark ist ein seit 1994 von der IBM distribuiertes Workflowmanagementsystem. Ebenso wie WorkParty ist FlowMark ein Client-Server-System, bei der die Entwicklungs- und Lauf-

²⁶ Vgl. auch Priemer (1995), S. 162, der diese als *freie* Attribute bezeichnet.

²⁷ Zu WorkParty vgl. auch Jablonski (1995), S. 87-89; Wersch (1995), S. 115-118.

²⁸ Der Begriff *Vorgang* wird in diesem Zusammenhang synonym zum Begriff *Workflow* verwendet.

²⁹ Vgl. Siemens Nixdorf (1994), S. 1f.

³⁰ Vgl. Kock, Rehäuser, Krcmar (1995), S. 36f.; Lang (1995); Versteegen (1995), S. 153.

³¹ Vgl. Siemens Nixdorf (1994), S. 1f.

³² Vgl. Kock, Rehäuser, Krcmar (1995), S. 38f.; Lukosch (1995); Siemens Nixdorf (1994).

zeitumgebung auf den Clients, die Datenbankbindung und zentralen Dienste hingegen auf dem Server ausgeführt werden. Im Gegensatz zu WorkParty erfolgt die Organisations- und Ablaufmodellierung in einer gemeinsamen Entwicklungsumgebung, dem Buildtime-Client. Neben den Modellierungswerkzeugen stellt der Buildtime-Client eine Animationskomponente zur Verfügung, mit deren Hilfe eine exemplarische Instanziierung des Workflowmodells vorgenommen werden kann. Der Im- und Export von Prozeßmodellen erfolgt durch eine Textdatei im FDL-Format (FlowMark Definition Language). Der Übergang von der Build- zur Runtime erfolgt durch die Kompilierung des erstellten Prozeßmodells in ein instanzitierbares Workflow-Template. Der FlowMark Runtime-Client stellt die Funktionalitäten zur Verwaltung der kompilierten Prozeßmodelle und zur Überwachung der instanziierten Workflows zur Verfügung. Die für Endbenutzer sichtbare Oberfläche mit der Liste zu bearbeitender Aktivitäten (Worklist) ist ebenfalls Bestandteil des Runtime-Client.

3.4 Vergleich der Organisationssicht

3.4.1 Die Organisationssicht von WorkParty

Ein *Akteur* ist ein Mitglied der Aufbauorganisation. Er ist einer oder mehreren Stellen zugeordnet, wobei eine Stelle davon immer die sogenannte *Stammstelle* des Akteurs ist. Die Stammstelle stellt die disziplinarische Zuordnung des Mitarbeiters dar.³³ Die Zuordnung von Akteuren zu Stellen ist weiterhin über eine Stellvertreterbeziehung möglich. Jeder Akteur kann die Stellvertretung für keine, eine oder mehrere Stellen ausüben.

Eine *Stelle* ist ein Aufgabenbereich eines Akteurs innerhalb einer Organisationseinheit und genau einer Organisationseinheit zugeordnet. Eine Stelle wird von maximal einem Akteur besetzt, kann aber auch unbesetzt sein. In der Sprachregelung von WorkParty wird eine Stelle als Abstraktion einer konkreten Person angesehen.³⁴ Eine Stelle kann keinen oder mehrere Akteure als Stellvertreter besitzen, wobei nur ein Akteur aktiver Stellvertreter sein kann. Eine Stelle ist keinem oder genau einem Stellentyp zugeordnet.

Stellentypen (Rollen) dienen in WorkParty dazu, Stellen mit ähnlichen Charakteristika über verschiedene Organisationseinheiten hinweg zu gruppieren.³⁵ Ein Stellentyp kann einer oder mehreren Stellen zugeordnet werden.

³³ Vgl. Rupietta (1992), S. 28.

³⁴ Vgl. Siemens Nixdorf (1995), S. 80.

³⁵ Vgl. Siemens Nixdorf (1995), S. 99.

Eine *Organisationseinheit* ist ein Teil der Aufbauorganisation, z. B. eine Abteilung. Sie wird durch die Komposition einer Menge von Stellen oder untergeordneten Organisationseinheiten gebildet. Organisationseinheiten können sowohl temporär als auch dauerhaft existieren. Im ersten Fall handelt es sich um Projekte, im letzteren um Linieneinheiten oder Stabsabteilungen, welche die dauerhafte Organisationshierarchie bilden.³⁶ Organisationseinheiten können nur streng hierarchisch angeordnet sein, d. h. jede Organisationseinheit kann beliebig viele untergeordnete, jedoch nur eine übergeordnete Organisationseinheit besitzen, wobei Linieneinheiten keinem Projekt untergeordnet sein dürfen. Einer Organisationseinheit können keine, eine oder mehrere Rollen und Stellen zugeordnet sein. Sie kann Besitzer mehrerer Ressourcen sein.

Eine *Rolle (Kompetenz)* beschreibt den Handlungsspielraum eines Akteurs durch Angabe seiner Zuständigkeiten und Befugnisse. Die Zuständigkeiten eines Akteurs werden im ORM durch die ihm übertragenen Aufgaben beschrieben, während seine Befugnisse durch die Zugriffsmöglichkeiten auf verschiedene Ressourcen abgebildet werden. Der Besitz der Ressource ist für die Zugriffsrechte unerheblich.³⁷

Rollen können im objektorientierten Sinne vererbt werden. WorkParty stellt dazu fünf Vererbungsregeln zur Verfügung, von denen minimal zwei Anwendung finden:

- *Stellenregel*: Akteure erben die Rollen der ihnen zugeordneten Stellen. Diese Vererbungsregel ist obligatorisch.
- *Stellentypregel*: Stellen erben die Rollen der ihnen zugeordneten Stellentypen. Diese Vererbungsregel ist ebenfalls obligatorisch.
- *Verteilerregel*: Stellen erben die Rollen ihrer Organisationseinheiten. Die Anwendung dieser Regel ist optional.
- *Leitungsregel*: Die Leitungsstelle einer Organisationseinheit erbt die Rollen dieser Organisationseinheit. Diese Regel ist ebenfalls optional, jedoch können Verteiler- und Leitungsregel nicht gleichzeitig angewandt werden.
- *Hierarchieregel*: Organisationseinheiten erben die Rollen der ihnen übergeordneten Organisationseinheiten. Die Anwendung dieser Regel ist optional.

³⁶ Vgl. Siemens Nixdorf (1994), S. 69; Lang (1995).

³⁷ Aus Gründen der Übersichtlichkeit wurden die besitzrechtlichen Beziehungen nicht in das ERM (Abbildung 4) aufgenommen.

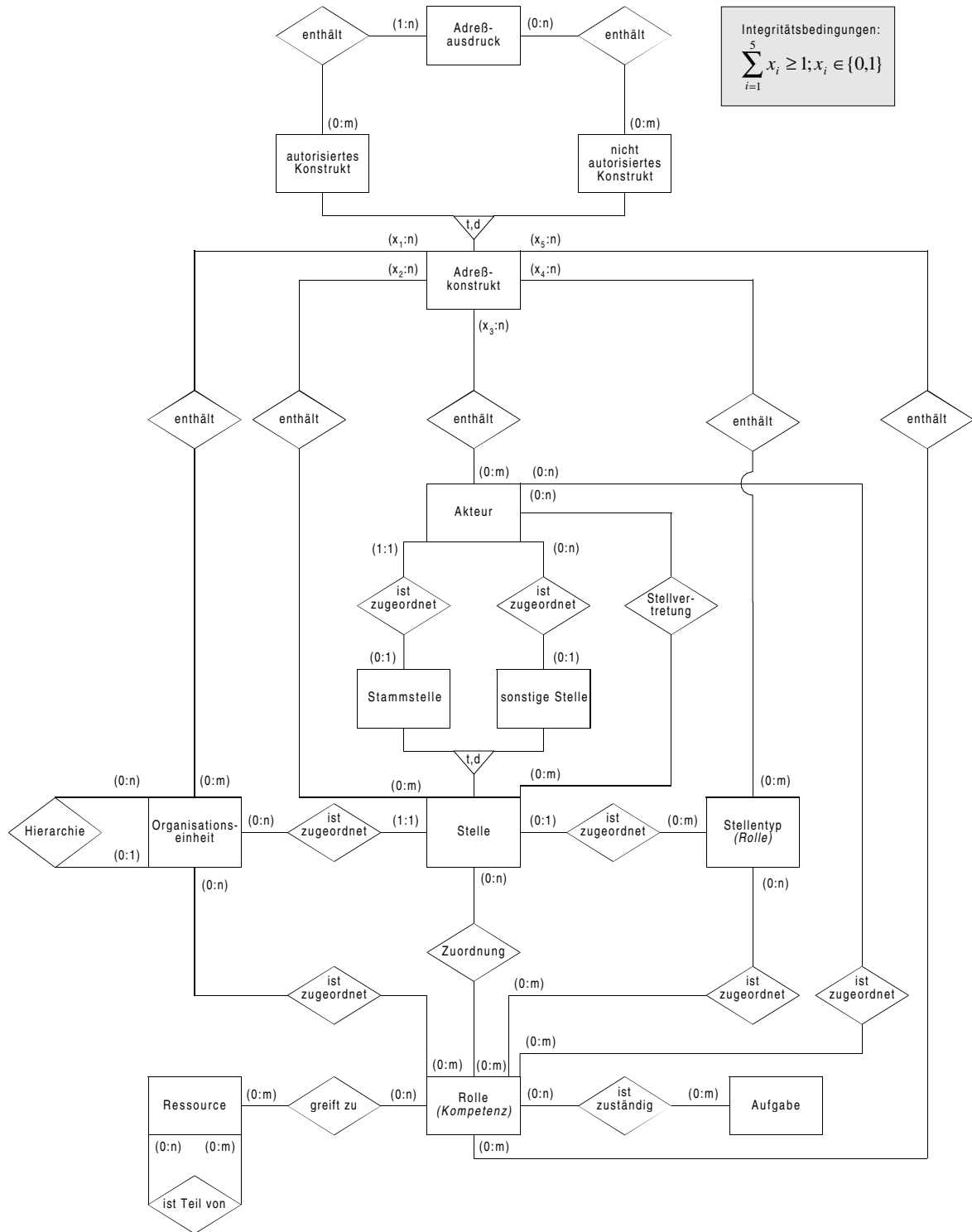


Abb. 4: ERM der WorkParty-Organisationsansicht³⁸

³⁸ Die Notation t,d innerhalb von Spezialisierungen/Generalisierungen besagt, daß es sich dabei um eine *totale* (t) und *disjunkte* (d) im Gegensatz zu einer *unvollständigen* und *überlappenden* Spezialisierung/Generalisierung handelt. Vgl. Vossen (1994), S. 73.

Eine *Aufgabe* ist eine Zielvorgabe für die Tätigkeit von Akteuren.³⁹ Wird eine Aufgabe einer Rolle zugeordnet, entstehen Zuständigkeiten im Organisationsmodell, d. h. die organisatorische Einheit mit der entsprechenden Rolle zur Ausführung einer Aufgabe ist für diese zuständig. Eine Aufgabe kann einer oder mehreren Rollen zugeordnet werden.

Ressourcen sind Sachmittel, die Aufgabenträgern zur Erfüllung ihrer Aufgaben zur Verfügung stehen, z. B. Arbeitsmittel oder Werkzeuge. Jede Ressource kann einer oder mehreren Rollen zugeordnet werden. Dadurch werden Zugriffsbeziehungen, d. h. Berechtigungen einzelner Akteure für die Benutzung bestimmter Ressourcen, definiert. Ressourcen haben einen eindeutigen Besitzer. Dies kann ein Akteur, eine Organisationseinheit oder eine Stelle sein. Ressourcen können hierarchisch angeordnet sein und so anderen Ressourcen über- bzw. untergeordnet werden.

3.4.2 Die Organisationssicht von FlowMark

Eine *Rolle* stellt eine abstrakte Qualifikation eines Akteurs dar. Jede Rolle kann einen Koordinator haben (z. B. einen Supervisor). Der Koordinator kann, muß aber nicht Mitglied der koordinierten Rolle sein. Jede Rolle kann kein oder mehrere Mitglieder besitzen. Wenn eine Aktivität nur durch Träger einer bestimmten Rolle ausgeführt werden darf, erscheint sie zur Laufzeit bei allen eingeloggten Trägern dieser Rolle in der individuellen Worklist und kann von einem dieser Akteure ausgewählt und bearbeitet werden, wobei sie bei allen anderen Anwendern aus der Worklist verschwindet.

Ebenso wie in WorkParty repräsentiert eine *Organisationseinheit* in FlowMark einen Teil der Aufbauorganisation, z. B. eine Abteilung einer Unternehmung. Sie hat genau einen Manager, wobei dieser eine Organisation leiten kann und gleichzeitig auch Mitglied dieser Organisation sein muß. Der Manager einer Organisation kann Koordinator von Rollen sein. Wird einer Organisation vom Anwender kein Manager zugeordnet, wählt FlowMark den zuletzt generierten Akteur aus, der noch keiner Organisation angehört. Ist ein solcher Akteur nicht vorhanden, legt FlowMark selbständig ein Akteurobjekt als Manager an. Zu einer Organisation können mehrere Mitglieder mit unterschiedlichen Rollen gehören, mindestens jedoch eines (der Manager). Organisationen können in FlowMark ebenfalls nur in strengen Hierarchie angelegt werden, d. h. eine Organisation hat genau eine übergeordnete Organisation, kann aber mehrere untergeordnete Organisationen besitzen.

³⁹ Vgl. auch Kosiol (1976), S. 43: „Unter Aufgaben sind Zielsetzungen für [...] Handlungen - Handlungsziele - zu verstehen.“

Hierarchieebenen (Levels) bilden eine Hierarchiestruktur der Unternehmung ab, die nicht zwingend der aufbauorganisatorischen Hierarchie entsprechen muß. Es stehen bis zu zehn unterschiedliche Stufen zur Verfügung, wobei ein Kriterium zur Festlegung der Ebenen nicht vorgegeben ist, d. h. es kann ein beliebiges Kriterium wie beispielsweise Dienstalder oder Tarifstufe (z. B. Top-, mittleres und operatives Management) abgebildet werden. Höhere Hierarchieebenen schließen in der Rollenauflösung alle darunterliegenden Ebenen ein.

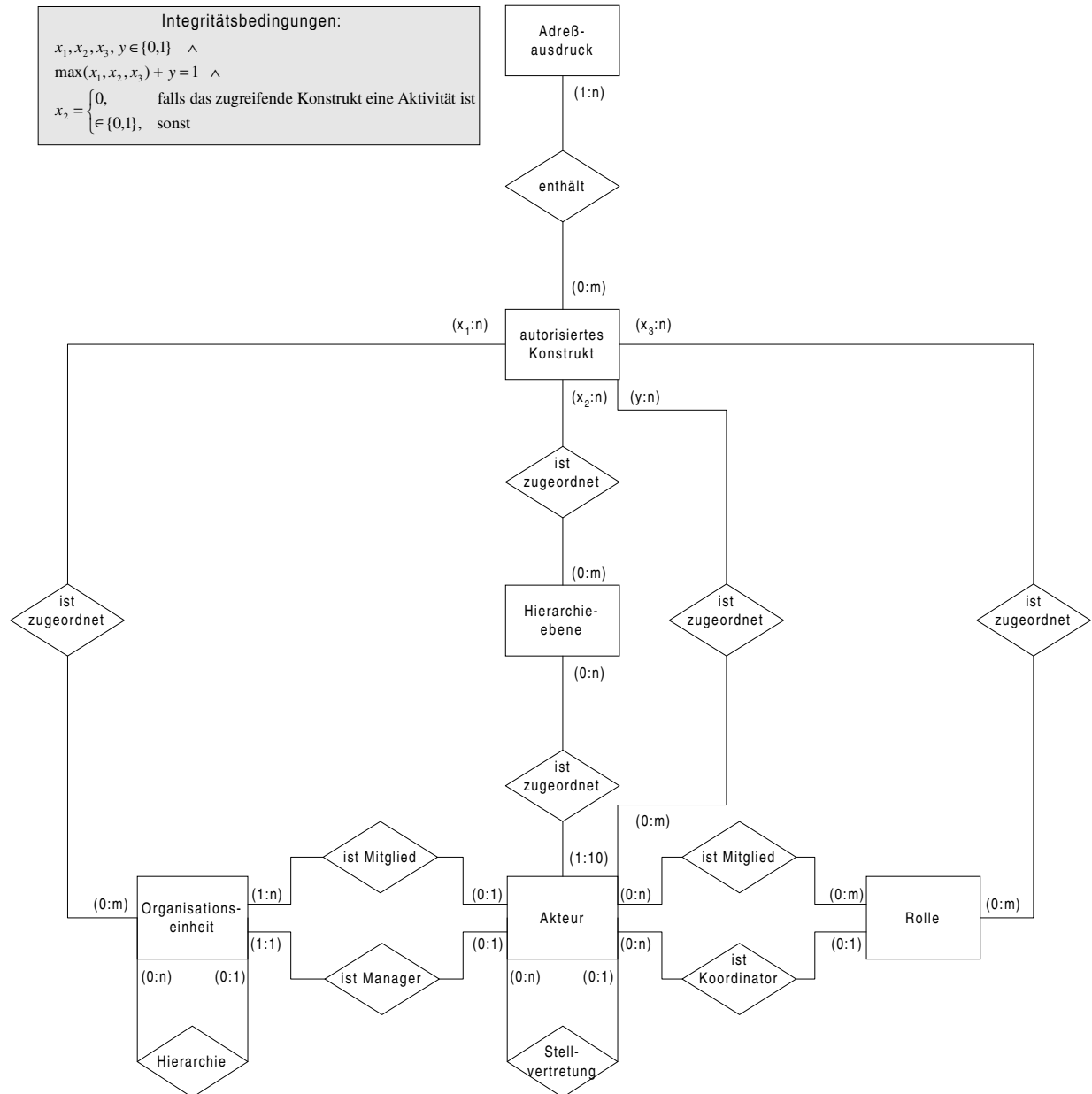


Abb. 5: ERM der FlowMark-Organisationssicht

Ein *Akteur* ist ein Aufgabenträger innerhalb der Aufbauorganisation. Er kann je nach Autorisierung Prozeß- bzw. Organisationsmodelle erstellen und/oder Aktivitäten ausführen. Jeder Akteur muß einer Hierarchieebene zugeordnet werden. Jedem Akteur können keine, eine oder mehrere

Rollen zugeordnet werden, er kann Mitglied maximal einer Organisation sein. Wenn der Akteur Manager der Organisation ist, muß er gleichzeitig auch Mitglied dieser Organisation sein.

Jedem Akteur kann eine Reihe von Autorisierungen zugeordnet werden:

- Definition und Modifikation von Personal, Rollen und Organisationen
- Modellierung von Prozessen
- Start und Kontrolle von Prozessen
- Zugriff auf die Aktivitäten anderer Akteure (generell oder speziell)

Jedem Akteur kann ein *Stellvertreter* zugeordnet werden, der im Fall der Abwesenheit des Akteurs die Aktivitäten des Akteurs auf seiner Worklist erhält. Jeder Akteur kann Stellvertreter mehrerer anderer Akteure sein. Ein Akteur kann sich nicht selbst vertreten.

3.4.3 Vergleich

Bereits während der Modellierung der zum Vergleich herangezogenen ERM wurde auf eine Auflösung von Namenskonflikten geachtet. So wurden beispielsweise die Konstrukte *Mitarbeiter* (WorkParty) und *Person* (FlowMark) in *Akteur* umbenannt, um einen Synonymkonflikt zu vermeiden. Ein Homonymkonflikt kann bei dem Konstrukt *Rolle* identifiziert werden, was nicht zuletzt auf das in der Praxis anzutreffende unterschiedliche Verständnis des Rollenkonstrukts zurückzuführen ist.⁴⁰ Eine Rolle in WorkParty dient zur Beschreibung von *Stellentypen*, die in mehreren Organisationseinheiten vorkommen können. Das Rollenkonstrukt innerhalb von FlowMark entspricht hingegen dem Kompetenzkonstrukt in WorkParty. Daher wurden im ERM von WorkParty die Konstrukte *Rolle* und *Kompetenz* in *Stellentyp* und *Rolle* umbenannt.

Im direkten Vergleich weist die Organisationssicht von WorkParty eine höhere Zahl an Entitytypen auf, was auf eine tendenziell höhere Modellierungsflexibilität schließen läßt. Insbesondere die Konstrukte *Stelle* und *Stellentyp* fehlen in der Organisationssicht von FlowMark. Im Gegensatz dazu bietet FlowMark in den Relationen *Akteur-Organisationseinheit* und *Akteur-Rolle* jeweils zwei verschiedene Beziehungsrollen an, während diese Relationen in WorkParty nur durch einen Beziehungstyp bzw. indirekt durch die Beziehung *Akteur-Stelle-Organisationseinheit* abgebildet werden. Dies deutet auf eine größere Differenzierungsmöglichkeit bei diesen Relationen in FlowMark hin.

Beim Vergleich der Kardinalitäten zeigt FlowMark restriktivere Relationen. Insbesondere die starre (1:1)-(1:n)-Relation zwischen *Akteur* und *Organisationseinheit* dürfte sich in praxi als hinderlich erweisen. WorkParty bietet hier durch die Möglichkeit, einem Akteur mehrere Stellen

⁴⁰ Vgl. Rupiotta (1992), S. 29ff.; Esswein (1993), S. 555ff.; Galler (1995), S. 7ff.

zuzuweisen, eine größere Flexibilität. In beiden Produkten ist die rekursive Beziehung auf dem Konstrukt *Organisationseinheit* als (0:1)-(0:n)-Relation ausgeprägt und somit auf strenge Hierarchien beschränkt. Dadurch lassen sich nicht alle Organisationsformen modellieren (z. B. Matrixorganisation). Ebenfalls restriktiv wirkt die Beschränkung auf einen Stellvertreter je Akteur in FlowMark ((0:1)-(0:n)-Relation). In WorkParty hingegen muß jeder Akteur einer Stelle und somit einer Organisationseinheit zugeordnet sein. Es ist somit nicht ohne weiteres möglich, unternehmensexterne Akteure zu modellieren.

Zum exemplarischen Attributvergleich sei an dieser Stelle das Konstrukt *Akteur* betrachtet. In WorkParty können zu einem Akteur folgende Attribute angegeben werden: Name, Vorname, Titel, Telefonnummer, Personalnummer, Anrede, Sprachcode, Stammstelle, Kennwort und Adresse. Zusätzlich können beliebig viele selbstdefinierte Attribute vergeben werden. In FlowMark besitzt ein Akteur folgende Attribute: UserID (eindeutig), PersonenID (optional), Paßwort, Name, Vorname, Telefonnummer1, Telefonnummer2. Da zur Laufzeit nur die Attribute *Name* bzw. *UserID* bei der Rollenauflösung berücksichtigt werden, besitzen die übrigen Attributen rein informatorischen Charakter. Die Existenz freier Attribute in WorkParty erhöht die Möglichkeit, die vorhandene Organisation möglichst vollständig abzubilden, während in FlowMark mit einem Informationsverlust zu rechnen ist.

Die Schnittstelle zwischen Prozeß- und Organisationssicht erfolgt sowohl bei FlowMark als auch bei WorkParty durch einen sog. *Adreßausdruck*. Dieser ermöglicht zur Laufzeit eine dynamische *Rollenauflösung*, d. h. anhand der im Adreßausdruck angegebenen Kriterien und der aktiven Anwender werden aktivitätenindividuell die Akteure ermittelt, welche die jeweiligen Aktivitäten des Workflows zur Bearbeitung erhalten. Allgemein wird dabei zwischen personengebundenen und nicht personengebundenen Adreßausdrücken unterschieden.⁴¹ Erstere referenzieren explizit die autorisierten Akteure, während letztere Kriterien wie Stellentyp oder Organisationseinheit enthalten, über die zur Laufzeit indirekt die Menge aller autorisierten Akteure ermittelt wird. Dies macht die Workflowmodelle gegenüber Änderungen der Organisationsstruktur robuster, als dies bei einer festen Zuordnung von Akteuren zu Aktivitäten möglich wäre. WorkParty bietet diesbezüglich zusätzlich die Möglichkeit, *nicht* autorisierte Konstrukte anzugeben, beispielweise „Alle Stellentypen *außer* Stellentyp XY“.

Die Modellierung der Aufbauorganisation, wie sie innerhalb des ORM wahrgenommen wird, gilt nicht nur im Vergleich mit FlowMark als leistungsstark.⁴² Das Konzept der Emanzipation der Organisationsmodellierung von der Prozeßmodellierung erscheint insbesondere interessant,

⁴¹ Vgl. Siemens Nixdorf (1994), S. 298.

⁴² Vgl. Kock, Rehäuser, Krcmar (1995), S. 38.

wenn durch die Offenlegung von Schnittstellen für den Anwender die Möglichkeit besteht, zwischen unterschiedlichen Modellierungstools zu wählen bzw. bereits existierende Anwendungen zu integrieren.

3.5 Vergleich der Prozeßsicht

Sowohl WorkParty als auch FlowMark verwenden Workflowmodelle, die dem Prinzip der Vorgangsmodelle folgen.⁴³ Während Workflows in FlowMark als gerichtete und mit Übergangsbedingungen gewichtete Graphen durch die explizite Festlegung von Knoten und Kanten modelliert werden, erfolgt die Workflowmodellierung in WorkParty durch die Verfeinerung vorgegebener Prozeßbausteine, die grundlegende Kontrollflußkonstrukte, wie Schleifen, Parallelitäten und Verzweigungen, zur Verfügung stellen.

3.5.1 Die Prozeßsicht von WorkParty

Ein *Prozeß (Ablauf)* in WorkParty ist ein gerichteter Graph, der die Ausführungsreihenfolge für Aktivitäten bzw. Programme festlegt.⁴⁴ Er besteht aus einem Anfangs- und einem oder mehreren Endesymbolen, Knoten und Kanten. Im Anfangssymbol werden globale Informationen hinterlegt, die für alle Knoten des Graphs gültig sind. Knoten enthalten entweder *Aktivitäten (Tätigkeiten)* oder *Programme*, Kanten regeln deren Ausführungsreihenfolge. Endesymbole bilden den Abschluß einer Ablaufbeschreibung. Durch die Verwendung mehrerer Endesymbole können alternative Endzustände modelliert werden, ein Prozeß hat jedoch immer einen eindeutigen Beginn.⁴⁵ Iterationen werden durch die Verwendung universeller Schleifenkonstrukte modelliert. Prozesse treten in WorkParty in den Ausprägungen *Aktivitätensequenz (Tätigkeitsablauf)* und *Programmsequenz* auf. Aktivitätensequenzen enthalten eine oder mehrere Aktivitäten, Programmsequenzen ein oder mehrere Programme. Prozesse werden in sog. Themenakten verwaltet und von Muster- bzw. Vorgangsakten referenziert.⁴⁶

Ein *Programm* ist eine Anwendung außerhalb von WorkParty, die im Rahmen einer (Programm-)Aktivität oder Programmsequenz gestartet wird. Programme können Daten mit dem aufrufenden Vorgang austauschen oder autonom ablaufen.

⁴³ Vgl. z. B. Reinwald (1993), S. 69-81.

⁴⁴ In WorkParty wird zwischen Programmsequenzen und Ablaufbeschreibungen differenziert. In *Programmsequenzen* werden nur externe Programme referenziert, in *Ablaufbeschreibungen* nur Tätigkeiten, die ihrerseits allerdings wieder Programme aufrufen können. Programmsequenzen eignen sich daher vorrangig für Batchverarbeitungen, die ohne Benutzereingriffe im Hintergrund ablaufen sollen.

⁴⁵ Vgl. Siemens Nixdorf (1994), S. 207.

⁴⁶ Aus Gründen der Übersichtlichkeit wurde im zugehörigen ERM (Abbildung 6) auf die Spezialisierung des Entitytyps *Akte* verzichtet.

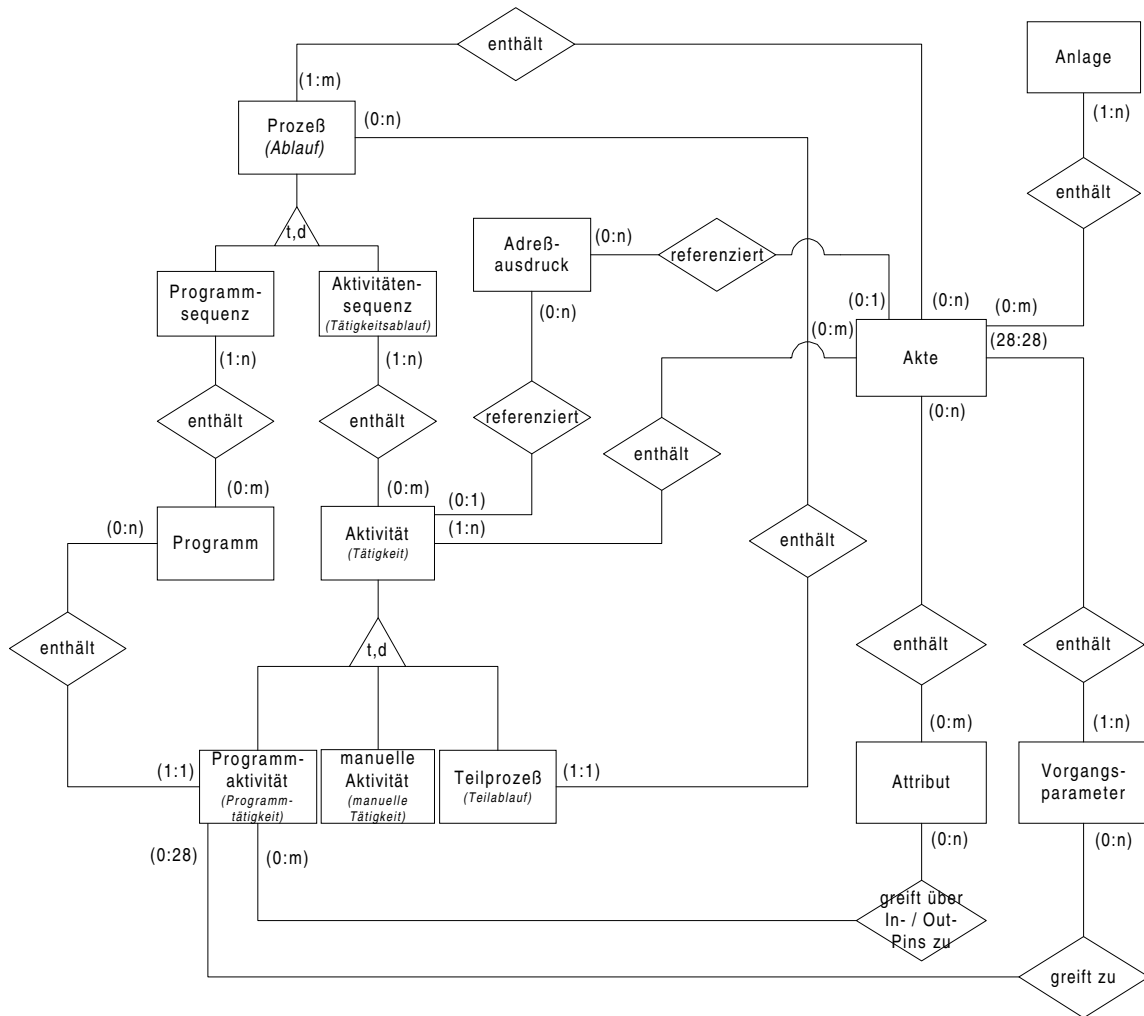


Abb. 6: ERM der WorkParty-Prozeßsicht

Eine *Aktivität* ist ein Teilschritt in einem Prozeß. Aktivitäten können sowohl manuell sein (Leertätigkeiten), als auch jeweils ein Programm referenzieren (programmgeführte bzw. programmgestützte Aktivitäten).⁴⁷ Leertätigkeiten ermöglichen dem Benutzer zur Laufzeit die Definition von Ad-hoc-Workflows durch das Einfügen neuer Aktivitäten in den Prozeß.

Vorgangsparameter sind 28 vom System vorgegebene Parameter, die Informationen über den aktuellen Status des Prozesses bzw. der aktuellen Aktivität beinhalten. Auf sie ist teilweise nur ein lesender, teilweise ein lesender und schreibender Zugriff möglich. Programme können über die Ablaufsteuerung auf die Vorgangsparameter zugreifen. Jeder Vorgangsparameter kommt in einer oder mehreren Akten vor.

⁴⁷ Zum Unterschied zwischen programmgeführten und programmgestützten Tätigkeiten vgl. Lang (1995).

Anlagen sind Dateien, die allen Aktivitäten eines Prozesses zur Verfügung stehen, beispielsweise eingescannte Dokumente oder vorgefertigte Briefe. Eine Anlage kann von einer oder mehreren Akten referenziert werden.

Attribute sind benutzerdefinierte Parameter, die über In- und Out-Pins an Programme übergeben, bzw. von diesen gelesen werden können. Attribute können sowohl prozeßglobal existieren als auch für einzelne Aktivitäten definiert werden.

Eine *Akte* ist eine Datenbank, in der WorkParty Prozeßmodelle und Aktivitätenbeschreibungen ablegt. Jede Akte kann keine oder beliebig viele Prozeßmodelle und Aktivitäten enthalten. Weiterhin können zu jeder Akte keine oder beliebig viele Anlagen angegeben und keine oder beliebig viele Attribute definiert werden.

3.5.2 Die Prozeßsicht von FlowMark

Ein *Prozeß* in FlowMark entspricht dem Ablaufkonstrukt in WorkParty, wobei neben der Ausführungsreihenfolge der Aktivitäten auch die Datenflüsse zwischen den Aktivitäten geregelt werden. Ein Prozeßmodell stellt die Summe aller möglichen Workflows dar, d. h. aller möglichen Pfade, die zwischen Prozeßstart und -ende abgearbeitet werden können. Es besteht aus einer oder mehreren Aktivitäten, die zur Erfüllung einer Aufgabe abgearbeitet werden müssen. Zur Laufzeit wird ein Pfad im Prozeßmodell abgearbeitet, d. h. ein bestimmter Workflow wird instanziiert. Durch Dead-Path-Elimination werden unerreichbare Zweige des Prozeßmodells zur Laufzeit aus dem Prozeß ausgeblendet. Jeder Prozeß kann einer von zehn Klassen (Categories) zugeordnet werden. Durch die Klassenzuordnung wird der Kreis der Benutzer eingeschränkt, der zur Bearbeitung des Prozesses berechtigt ist. Zu einer Klasse können mehrere Prozesse gehören. Ein Prozeß kann über einen Adreßausdruck nur einem Teil der Benutzer zugänglich gemacht werden. Weiterhin kann ein Prozeßadministrator bestimmt werden, der bei Problemen automatisch benachrichtigt wird. Wird kein Prozeßadministrator explizit angegeben, wird der Starter des Prozesses für die Laufzeit des Prozesses zum Prozeßadministrator. Einem Prozeß kann ein bestimmtes Zeitlimit eingeräumt werden, nach dessen Ablauf der Prozeßadministrator benachrichtigt wird, falls der Prozeß nicht innerhalb des Zeitlimits terminiert.

Eine *Aktivität* ist ein Teilschritt in einem Prozeßmodell und entspricht einer Aktivität in WorkParty. Die im Rahmen eines Prozesses auftretenden Aktivitäten können seriell, parallel oder alternativ abgearbeitet werden. Aktivitäten können Benutzereingaben erfordern oder automatisch, d. h. im Hintergrund, ablaufen. In FlowMark werden drei Arten von Aktivitäten unterschieden:

- *Prozeßaktivität*

Im Rahmen von Prozessen können mit Hilfe von Prozeßaktivitäten andere Prozesse, sogenannte Subprozesse, aufgerufen werden. Subprozesse sind vollständige Prozesse, die durch die Prozeßaktivität zusätzlich mit einer Start- und einer Endbedingung versehen werden. Es ist möglich, innerhalb eines Subprozesses den aufrufenden Prozeß selbst wieder aufzurufen.

- *Programmaktivität*

Programmaktivitäten sind (teil-)automatisierte Aktivitäten, bei deren Aufruf ein hinterlegtes Anwendungsprogramm gestartet wird. Ein solches Programm kann automatisiert ablaufen, ohne daß der Benutzer eingreifen muß (z. B. das automatische Ausdrucken einer Rechnung), oder auch Benutzereingaben erfordern (z. B. das Erfassen von Kundendaten in einer Datenbankmaske).

- *Schleifenkonstrukt*

Da in FlowMark - im Gegensatz zu WorkParty - Schleifen nicht durch Kontrollflußkonstrukte abgebildet werden können, existieren hier zwei Konstrukte zur Schleifenbildung: Block und Bundle.

a) Block

Ein *Block* entspricht einem Prozeß, hat jedoch keinen Namen und eine Endbedingung, welche der Schleifenbildung dient. Während ein Prozeß im Normalfall mit dem Ende der letzten Aktivität terminiert, wird ein Block so lange zyklisch ausgeführt, bis seine Endbedingung erfüllt ist. Dies entspricht dem Programmkonstrukt einer DO...UNTIL-Schleife. Ein Block kann alle Elemente eines Prozeßmodells enthalten, also auch einen anderen Block.

b) Bundle

Ein *Bundle* entspricht einem Block, mit dem Unterschied, daß ein Bundle lediglich eine Aktivität enthält, von der mehrere Instanzen parallel ablaufen und zyklisch wiederholt werden, bis eine definierte Endbedingung erfüllt ist. Der wesentliche Unterschied in der Modellierung liegt darin, daß beim Bundle zum Zeitpunkt der Modellierung *nicht* bekannt ist, wie viele Instanzen der Aktivität zur Laufzeit parallel ausgeführt werden sollen. Diese Zahl wird zur Laufzeit durch eine vorgeschaltete *Planungsaktivität* ermittelt, die eine spezielle Programmaktivität ist. Sie referenziert ein externes Programm, innerhalb dessen entweder die Anzahl der aktiven Benutzer bestimmt wird, welche die Bundle-Aktivität bearbeiten dürfen (People-Driven-Instantiation), oder welches anhand des Inhalts des Inputcontainers der Programmaktivität die Anzahl der zu erzeugenden Instanzen der Bundle-Aktivität ermittelt (Data-Driven-Instantiation).

Jeder Programm- und Prozeßaktivität muß eine Priorität zwischen null (niedrig) und neun (hoch) zugeordnet werden. Diese Priorität regelt, an welcher Stelle der Worklist des Benutzers eine Aktivität erscheint. Start- und Endbedingungen für die einzelnen Aktivitäten können optional definiert werden. Als Startbedingung steht zur Wahl, ob der Status aller eingehenden Kontrollkonnektoren oder nur der Status eines Kontrollkonnektors wahr sein muß, damit die Aktivität gestartet werden kann. Die Endbedingung kann ein beliebiger boolescher Ausdruck sein. Wird die Endbedingung einer Aktivität nicht erfüllt, wird diese wieder in den *ready*-Status versetzt und kann erneut gestartet werden. Erst wenn die Endbedingung erfüllt ist, können die von der Aktivität ausgehenden Kontrollkonnektoren verfolgt werden. Die Einstellung, ob der Anwender den Start und/oder das Ende der Aktivität manuell bestätigen muß, ist ebenfalls optional. Für Blöcke gilt allerdings, daß diese automatisch gestartet werden, sobald ihre Startbedingung wahr wird. Sie enden automatisch, sobald ihre Endbedingung wahr wird.

Jeder Programmaktivität muß ein Programm zugeordnet werden, welches zur Laufzeit der Aktivität ausgeführt wird. Das entsprechende Programm muß in der FlowMark-Datenbank registriert sein. Jeder Prozeßaktivität muß ein Prozeß zugeordnet werden, der zur Laufzeit der Aktivität gestartet wird.

Ein *Programm* ist eine Anwendung außerhalb von FlowMark. Damit innerhalb eines Workflows solche Anwendungen aufgerufen werden können, müssen sie in der FlowMark-Datenbank registriert werden. FlowMark unterstützt OS/2-Programme und DLLs, Microsoft-DOS und Windows-Programme sowie DLLs, AIX-Programme und Shell Scripts, Application Support Facility Tools, sowie das Bundle Planning Tool und das FlowMark Manual Checklist Program, welches dem Anwender eine Checkliste für manuell auszuführende Tätigkeiten zur Verfügung stellt.

Jeder Programm- und Prozeßaktivität können ein oder mehrere *Support Tools* zugeordnet werden. Dabei handelt es sich um registrierte Programme, die der Anwender von seiner Worklist starten kann. Diese Programme sollen die Abarbeitung der Aktivität unterstützen und sind nicht mit den Programmen der Programmaktivitäten zu verwechseln, denn ihr Start ist optional und hat keinen unmittelbaren Einfluß auf den Ablauf der Programm- bzw. Prozeßaktivität. Beispiele für Support Tools sind ein Dictionary oder ein Lexikon auf CD-ROM, welche den Anwender beim Erstellen eines Briefes unterstützen.

Jedem Prozeß und jeder Aktivität muß mindestens eine Person zugeordnet werden, die zur Bearbeitung des Prozesses bzw. der Aktivität berechtigt ist. Die Zuordnung von Personal zu Aktivitäten kann statisch oder dynamisch erfolgen. Bei der dynamischen Zuordnung wird jeder Aktivität eine Rolle und/oder Organisation zugeordnet, der die Bearbeiter der Aktivität angehören.

ren müssen. Im Falle der statischen Zuordnung wird die UserID der Mitarbeiter angegeben, die zur Bearbeitung der Aktivität berechtigt sind.

Sollte im Rahmen der Dead-Path-Elimination eine Aktivität Daten aus Aktivitäten benötigen, die sich in einem nicht durchlaufenen Pfad des Prozeßmodells befinden, so kann wahlweise der Benutzer aufgefordert werden, die fehlenden Daten manuell einzugeben, oder der Prozeß mit einer Fehlermeldung abgebrochen werden.

Container sind Behälter für Parameter, die für den Datenfluß zwischen Aktivitäten, bzw. zwischen Aktivitäten und Prozessen benötigt werden. Dabei wird zwischen Input- und Output-containern unterschieden. Inputcontainer beinhalten die Eingabeparameter einer Aktivität (bzw. eines Prozesses), Outputcontainer die Ausgabeparameter. Jede Aktivität sowie jeder Prozeß und jeder Block besitzen einen Input- und einen Outputcontainer.

Ein *Konnektor* ist eine gerichtete Kante zwischen zwei Aktivitäten oder zwei Containern. In FlowMark wird zwischen Kontrollfluß- und Datenkonnektoren unterschieden, als Spezialisierung der Kontrollflußkonnektoren existieren Kontroll- und Defaultkonnektoren.

- *Kontrollflußkonnektoren*

Aktivitäten können mit Hilfe von Kontrollflußkonnektoren verbunden werden, welche die logische Reihenfolge der Aktivitäten zur Laufzeit bestimmen. Dabei wird zwischen Kontroll- und Defaultkonnektoren unterschieden. Kontrollkonnektoren können mit einer booleschen Bedingung, der sogenannten *Transitionsbedingung*, versehen werden. Ist diese erfüllt, wird die verbundene Aktivität gestartet, sonst nicht. Durch die Verwendung von Kontrollkonnektoren kann festgelegt werden, daß bestimmte Aktivitäten erst gestartet werden können, wenn andere Aktivitäten abgeschlossen sind. Auch alternative Routings im Prozeßmodell können so abgebildet werden.

Der Defaultkonnektor ist ein Sonderfall der Kontrollflußkonnektoren, der verfolgt wird, wenn die Transitionsbedingungen der anderen Kontrollkonnektoren nicht erfüllt sind (Else-Bedingung). Defaultkonnektoren unterstützen demnach auch das Exception-Handling im Prozeßablauf.

- *Datenkonnektoren*

Datenkonnektoren verbinden Datencontainer. Sie regeln, welche Daten aus dem Outputcontainer einer Aktivität in den Inputcontainer der verbundenen Aktivität fließen. Es werden nur Daten aus korrekt terminierten Aktivitäten an nachgelagerte Aktivitäten weitergereicht. Die Inputcontainer von Prozessen können mit den Inputcontainern von Aktivitäten und Blöcken verbunden werden, gleiches gilt für die Outputcontainer von Prozessen, Aktivitäten

und Blöcken. Es ist möglich, den Outputcontainer einer Aktivität mit dem Inputcontainer derselben Aktivität zu verbinden.

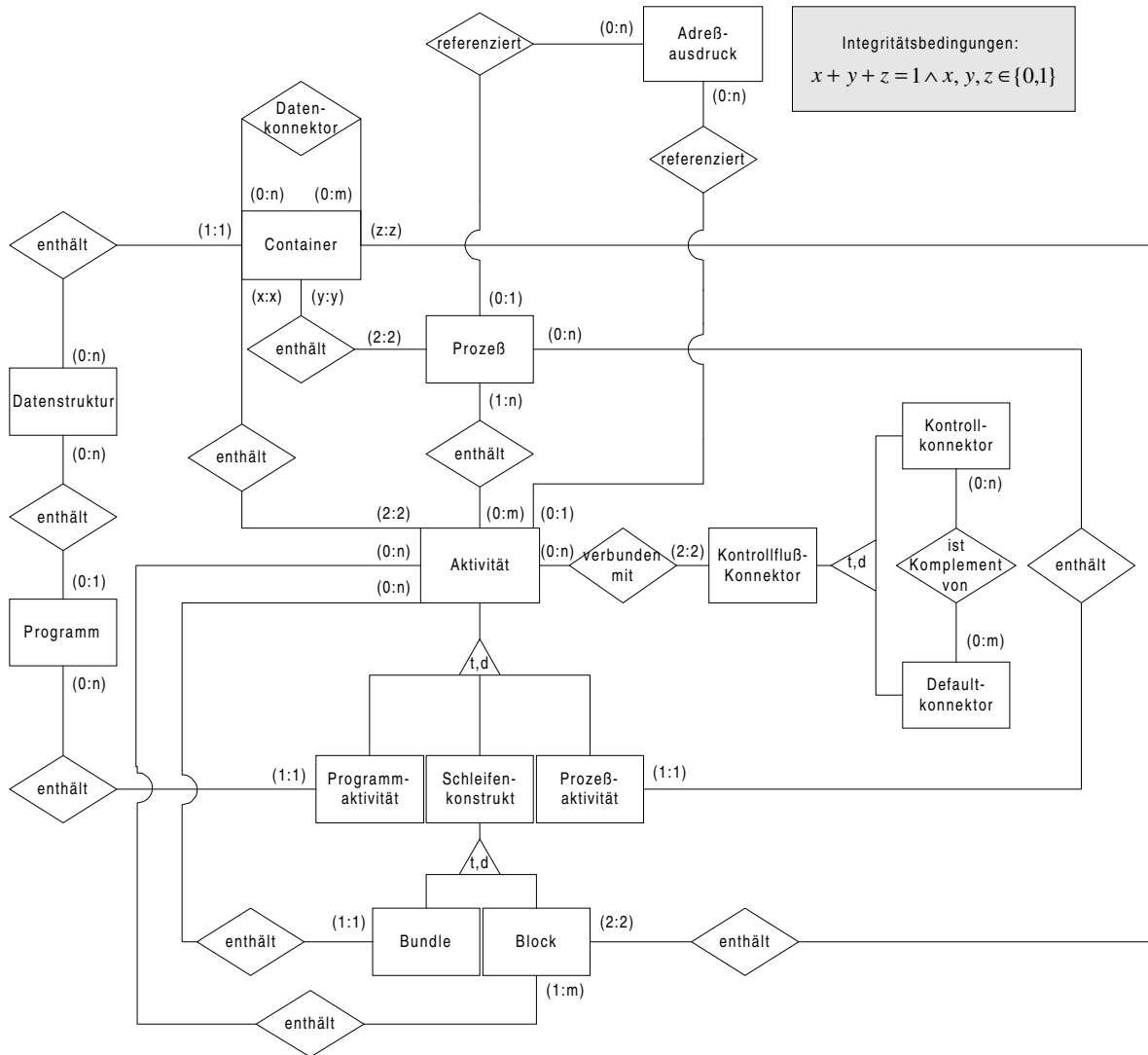


Abb. 7: ERM der FlowMark-Prozeßsicht

Jeder Container enthält genau eine Datenstruktur. Eine *Datenstruktur* in FlowMark ist der Inhalt eines Datencontainers, d. h. sie enthält die Ein- bzw. Ausgabeparameter einer Aktivität, eines Blocks oder eines Prozesses. Wenn die verbundenen Datenstrukturen ungleich sind, wird anhand eines *Data Mapping* festgelegt, welches Datum des Outputcontainers mit welchem Datum des Inputcontainers verbunden wird. Registrierte Programme enthalten ebenfalls je eine In- und eine Outputdatenstruktur, die im Bedarfsfall als Muster für die Datenstrukturen der entsprechenden Programmaktivität genutzt wird. Es ist möglich, eine bereits definierte Datenstruktur als Parameter anzugeben. In diesem Fall spricht man von einer geschachtelten (*nested*) Datenstruktur. Wenn vom Anwender keine spezielle Datenstruktur spezifiziert wird, wird eine Standard-Datenstruktur angenommen.

3.5.3 Vergleich

Der Vergleich der Prozeßsichten offenbart eine Reihe von Konflikten, die in den programm-spezifischen Terminologien begründet liegen. Beispielsweise wird das WfMC-Konstrukt *activity* in WorkParty als *Tätigkeit* bezeichnet, in FlowMark hingegen als *Aktivität*. Ein *Prozeß* in FlowMark wird in WorkParty als *Ablauf* bezeichnet. Die WorkParty-Konstrukte im ERM wurden daher in Anlehnung an die Terminologie der WfMC umbenannt.

In FlowMark werden Kontrollflüsse separat modelliert und erscheinen daher als Kontroll- und Defaultkonnektoren im ERM. In WorkParty hingegen erfolgt die Modellierung von Kontrollflüssen aus vorgefertigten Bausteinen, die zum Teil in FlowMark nachgebildet werden können (und vice versa), teilweise jedoch keine Entsprechung finden (z. B. unterschiedliche Endzustände in WorkParty, unterschiedliche Startaktivitäten in FlowMark). Der bei FlowMark auftretende Entitytyp *Bundle* kann in WorkParty beispielsweise durch die Kombination von Parallelverarbeitungs- und Schleifenkonstrukten emuliert werden, jedoch mit teilweise abweichender Funktionalität. Der Vergleich der Kontrollflußsemantik ist mithin durch die vorgestellten ERM nicht möglich.

Im Gegensatz zur Organisationssicht erscheint bei der Prozeßsicht ein Vergleich der Zahl der Entitytypen nicht sinnvoll, da die reine Anzahl an Entitytypen aufgrund der unterschiedlichen Philosophien keine Rückschlüsse auf Vor- oder Nachteile zuläßt. Die Wahl des Detaillierungsgrades gestaltet sich ebenfalls schwierig, da einerseits die Datenmodelle nicht unbegrenzt wachsen, andererseits genug Vergleichsmöglichkeiten vorhanden sein sollten.

Im Bereich der Datenflüsse zeigt WorkParty eine feinere Auflösung als FlowMark. Während in FlowMark Daten in Form von *Datenstrukturen* und *Containern* abgelegt sind, werden bei WorkParty prozeßglobale *Anlagen*, *Attribute* und *Vorgangparameter* differenziert. Weiterhin sind die Daten in FlowMark aktivitätsbezogen abgelegt, da die Datencontainer an *Prozesse*, *Blöcke* und *Aktivitäten* gebunden sind. In WorkParty können Daten hingegen sowohl aktivitätsgebunden als auch prozeßglobal abgelegt werden.

Durch die große Varietät in der Workflowmodellierung, welche die unterschiedlichen Systeme aufweisen, gestaltet sich der Vergleich der Prozeßsicht ungleich schwieriger als der Vergleich der mit deutlich weniger Freiheitsgraden versehenen Organisationssicht. Deshalb erscheint die Erweiterung des Vergleichs um einen Abgleich mit einer Referenzbibliothek von Kontrollflußprimitiven sinnvoll, welche generelle Kontrollflußkonstrukte enthält. Die Ergebnisse eines solchen Abgleichs sind ein Ansatz zur Bewertung der Mächtigkeit der jeweiligen Prozeßmodellierung. Weiterhin ist die Ausdehnung des Systemvergleichs um eine Analyse der zugrundeliegenden Vorgehensmodelle (Metaprozeßmodelle) geplant.

4 Résumé und Forschungspotential

Metamodelle bewegen sich stets in einem Spannungsfeld zwischen Komplexität (des zugrundeliegenden Modells) und Kompliziertheit (des Metamodells). Je mehr Elemente ein Metamodell enthält, desto komplizierter ist es. Zugleich steigt aber im Regelfall mit zunehmender Anzahl an Elementen im Metamodell die Möglichkeit Sachverhalte im Modell kompakter abzubilden. Dies trägt zur Reduktion der Modellkomplexität bei.

Der Lösungsbeitrag von Metadatenmodellen beim Vergleich von Workflowmanagementsystemen hängt zu großen Teilen vom Kontext der Untersuchung ab.

Im Bereich der *Organisationsmodellierung* bieten Metadatenmodelle aufgrund der relativ geringen Individualität der Organisationsmodellierung und der sich dadurch herausgebildeten weitgehend allgemeingültigen Terminologie (Stelle, Organisationseinheit, Mitarbeiter etc.) einen guten Ansatzpunkt für einen Systemvergleich.

Im Gegensatz dazu sind Metadatenmodelle im Bereich der *Prozeßmodellierung* nur als ein Instrument innerhalb eines Systemvergleichs zu betrachten. Hier sind weitere Methoden zur Entscheidungsunterstützung gefordert, z. B. Vorgehensmodelle oder Kontrollflußprimitive.

Zusammenfassend bleibt festzuhalten, daß Metadatenmodelle *eine* Möglichkeit zur Beschreibung von Workflowmanagementsystemen darstellen. Zugleich wurde aber auch deutlich, daß sie alleine für die Systembewertung nicht hinreichend sein können. Insbesondere einsatzkritische Leistungsmerkmale wie die Synchronisation im Falle von Nebenläufigkeit oder die Transaktionssicherung können durch Metadatenmodelle nicht hinreichend beschrieben werden.

Es wurden einige Ansätze zur Erweiterung bestehender Modellierungsmethoden vorgestellt (z. B. die variabilisierten Integritätsbedingungen), welche die Eignung dieser Modelle erhöhen. Zukünftige Arbeiten müssen hier u. a. die Abbildung weiterer instanzabhängiger Integritätsbedingungen zum Gegenstand haben (z. B. der Manager einer Organisationseinheit muß zugleich deren Mitglied sein).

Gegenstand der weiteren Forschung auf diesem Gebiet sind weiterhin - neben der Ausweitung der Betrachtung auf weitere Systeme - die Erstellung von Metaprozeßmodellen, die Ermittlung eines Katalogs grundlegender Kontrollflußkonstrukte sowie die Entwicklung eines vorrangig induktiv gewonnenen Referenz-Metamodells als konsolidierendes Ergebnis dieser Arbeiten.

Literatur

- Amberg, M.: Transformation von Geschäftsprozeßmodellen des SOM-Ansatzes in workflow-orientierte Anwendungssysteme. In: Workflowmanagement - State-of-the-Art aus Sicht von Theorie und Praxis. Hrsg.: J. Becker, M. Rosemann. Arbeitsbericht des Instituts für Wirtschaftsinformatik Nr. 47. Hrsg.: J. Becker, H. L. Grob, U. Müller-Funk, G. Vossen. Münster 1996, S. 46-56.
- Amberg, M., Raue, H.: Eine Beschreibungsform für Informationssystem-Architekturen. Rundbrief des GI-Fachausschusses 5.2, 2 (1995) 1, S. 67-75.
- Appelrath, H.-J., Ludewig, J.: Skriptum Informatik - Eine konventionelle Einführung. 2. Aufl., Stuttgart, Zürich 1992.
- Becker, J., Rosemann, M., Schütte, R.: Grundsätze ordnungsmäßiger Modellierung. Wirtschaftsinformatik, 37 (1995) 5, S. 435-445.
- Chen, P. P.-S.: The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems, 1 (1976) 1, S. 9-36.
- Derungs, M., Vogler, P., Österle, H.: Metamodell Workflow. Report Nr. HSG/CC PSI/3. Version 1.0. St. Gallen 1995 (<http://www-iwi.unisg.ch/iwi2/cc/psi/abpsi3.html>).
- Erdl, G., Schoenecker, H. G.: Geschäftsprozeßmanagement - Vorgangssteuerungssysteme und integrierte Vorgangsbearbeitung. Studie. Hrsg.: B.BIT Consult GmbH. München 1992.
- Esswein, W.: Das Rollenmodell der Organisation: Die Berücksichtigung aufbauorganisatorischer Regelungen in Unternehmensmodellen. Wirtschaftsinformatik, 35 (1993) 6, S. 551-561
- Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. Band 1. 2. Aufl., München, Wien 1994.
- Ferstl, O. K.; Sinz, E. J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. Wirtschaftsinformatik, 37 (1995) 3, S. 209-220.
- Galler, J.: Metamodelle des Workflow-Managements. Veröffentlichung des Instituts für Wirtschaftsinformatik Nr. 121. Hrsg.: A.-W. Scheer. Saarbrücken 1995.
- Georgakopoulos, D., Horwick, M., Sheth, A.: An Overview of Workflow-Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, 3 (1995) 2, S. 119-153.
- Guntram, U.: Die Allgemeine Systemtheorie. Ein Überblick. ZfB, 55 (1985) 3, S. 296-323.
- Hars, A.: Referenzdatenmodelle. Grundlagen effizienter Datenmodellierung. Wiesbaden 1994.
- Heilmann, H.: Workflow Management: Integration von Organisation und Informationsverarbeitung. HMD, 31 (1994) 176, S. 8-21.
- IBM Flowmark: Modeling Workflow. Release 2.1. Document Number: SH19-8241-00. Stand: März 1995. Hrsg.: IBM Deutschland Entwicklung GmbH. Wien 1995.

- Jablonski, S.: Workflow-Management-Systeme. Modellierung und Architektur. Bonn u. a. 1995.
- Jablonski, S.: Anforderungen an die Modellierung von Workflows. In: Praxis des Workflow-Managements. Grundlagen, Vorgehen, Beispiele. Hrsg.: H. Österle, P. Vogler. Baunschweig, Wiesbaden 1996, S. 65-81.
- Jablonski, S., Stein, K.: Die Eignung objektorientierter Analysemethoden für das Workflow Management. HMD, 32 (1995) 185, S. 95-115.
- Joosten, S.: Conceptual Theory for Workflow Management Support Systems. Technical Report. Centre for Telematics and Information Technology. University of Twente. Enschede 1995.
- Kock, T., Rehäuser, J., Krcmar, H.: Ein Vergleich ausgewählter Workflow-Systeme. Information Management, 10 (1995) 1, S. 36-43.
- Kosiol, E.: Organisation der Unternehmung. 2. Aufl., Wiesbaden 1976.
- Lang, G.: Konzeption der Abbildung von Geschäftsprozeßmodellen nach dem SOM-Ansatz in Spezifikationen des Workflowmanagementsystems WorkParty. Diplomarbeit am Lehrstuhl für Wirtschaftsinformatik der Universität Bamberg (Prof. Dr. E. J. Sinz), Bamberg 1995.
- Lippold, H., Hett, H.-M., Hilgenfeldt, J., Klagge, D.: BIFOA-Marktübersicht: Vorgangsmanagementsysteme. Köln 1993.
- Lukosch, T.: Konzeption der Abbildung von Geschäftsprozeßmodellen nach dem SOM-Ansatz in Spezifikationen des Workflowmanagementsystems FlowMark. Diplomarbeit am Lehrstuhl für Wirtschaftsinformatik der Universität Bamberg (Prof. Dr. E. J. Sinz), Bamberg 1995.
- Nissen, H. W., Jeusfeld, M. A., Jarke, M., Zemanek, G. V., Huber, H.: Managing Multiple Requirements Perspectives with Metamodels. IEEE Software, 13 (1996) 3, S. 37-48.
- Oberweis, A.: Verteilte betriebliche Abläufe und komplexe Objektstrukturen: Integriertes Modellierungskonzept für Workflow-Managementsysteme. Habilitationsschrift, Uni Karlsruhe 1994.
- Österle, H., Gutzwiller, T., Sprenger, U.: Referenz-Metamodell für den Entwurf betrieblicher Informationssysteme. Rundbrief des GI-Fachausschusses 5.2, 1 (1994) 1, S. 12-15.
- Picot, A., Rohrbach, P.: Organisatorische Aspekte von Workflow-Management-Systemen. Information Management, 10 (1995) 1, S. 28-35.
- Priemer, J.: Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle. Sinzheim 1995.
- Reinwald, B.: Workflow-Management in verteilten Systemen. Stuttgart, Leipzig 1993.
- Richter, H., Senger, G., Holz, C., Müller-Luschnat, G., Thiele, W.: Metamodell zur Modellierung von Geschäftsprozessen. FAST-Bericht 95-12. München 1995.
- Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden 1996.

- Rupietta, W.: Organisationsmodellierung zur Unterstützung kooperativer Vorgangsbearbeitung. *Wirtschaftsinformatik*, 34 (1992) 1, S. 26-37.
- Siemens Nixdorf: Organisations- und Ressourcenmanagement Version 2.0, Referenzhandbuch, Ausgabe Juni 1995, München 1995.
- Siemens Nixdorf: WorkParty Benutzerhandbuch Version 2.0, München 1994.
- Scheer, A.-W.: *Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse*. 6. Aufl., Berlin et al. 1995.
- Schulze, W., Böhm, M.: Klassifikation von Workflowmanagementsystemen. Modelle, Methoden, Werkzeuge. In: *Geschäftsprozeßmodellierung und Workflow-Management*. Hrsg.: G. Vossen, J. Becker. Bonn et al. 1996, S. 279-293.
- Sinz, E. J., Popp, K. M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt. In: *Bamberger Beiträge zur Wirtschaftsinformatik Nr. 12*. Hrsg.: W. Augsburg, O. K. Ferstl, E. J. Sinz. Bamberg 1992.
- Steele, P. M., Zaslavsky, A. B.: The Role of Meta Models in Federating System Modelling Techniques. In: *Proceedings of the 12th International Conference on the Entity-Relationship Approach - ER '93*. Hrsg.: R. A. Elmasri, V. Kouramajian, B. Thalheim. Berlin et al. 1994, S. 315-326.
- Strahinger, S. (ERM): Eine kardinalitätsbezogene Erweiterung der Entity-Relationship-Modellierung. In: *Schriften zur Quantitativen Betriebswirtschaftslehre der TH Darmstadt Nr. 8/95*. Hrsg.: W. Domschke, H. Stadtler, H. J. Petzold. Darmstadt 1995.
- Strahinger, S. (Metamodell): Zum Begriff des Metamodells. In: *Schriften zur Quantitativen Betriebswirtschaftslehre der TH Darmstadt Nr. 6/95*. Hrsg.: W. Domschke, H. Stadtler, H. J. Petzold., Darmstadt 1995.
- Versteegen, G.: Alles im Fluß. Die Ansätze der Workflow Management Coalition. *iX*, 3 (1995), S. 152-160.
- Vossen, G.: *Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme*. 2. Aufl., Bonn u. a. 1994.
- Wersch, M.: *Workflow Management. Systemgestützte Steuerung von Geschäftsprozessen*. Wiesbaden 1995.
- Workflow Management Coalition: *Glossary. A Workflow Management Coalition Specification*. Brüssel 1994.

Arbeitsberichte des Instituts für Wirtschaftsinformatik

- Nr. 1 Bolte, Ch., Kurbel, K., Moazzami, M., Pietsch, W.: Erfahrungen bei der Entwicklung eines Informationssystems auf RDBMS- und 4GL-Basis; Februar 1991.
- Nr. 2 Kurbel, K.: Das technologische Umfeld der Informationsverarbeitung - Ein subjektiver 'State of the Art'-Report über Hardware, Software und Paradigmen; März 1991.
- Nr. 3 Kurbel, K.: CA-Techniken und CIM; Mai 1991.
- Nr. 4 Nietsch, M., Nietsch, T., Rautenstrauch, C., Rinschede, M., Siedentopf, J.: Anforderungen mittelständischer Industriebetriebe an einen elektronischen Leitstand - Ergebnisse einer Untersuchung bei zwölf Unternehmen; Juli 1991.
- Nr. 5 Becker, J., Prischmann, M.: Konnektionistische Modelle - Grundlagen und Konzepte; September 1991.
- Nr. 6 Grob, H. L.: Ein produktivitätsorientierter Ansatz zur Evaluierung von Beratungserfolgen; September 1991.
- Nr. 7 Becker, J.: CIM und Logistik; Oktober 1991.
- Nr. 8 Burgholz, M., Kurbel, K., Nietsch, Th., Rautenstrauch, C.: Erfahrungen bei der Entwicklung und Portierung eines elektronischen Leitstands; Januar 1992.
- Nr. 9 Becker, J., Prischmann, M.: Anwendung konnektionistischer Systeme; Februar 1992.
- Nr. 10 Becker, J.: Computer Integrated Manufacturing aus Sicht der Betriebswirtschaftslehre und der Wirtschaftsinformatik; April 1992.
- Nr. 11 Kurbel, K., Dornhoff, P.: A System for Case-Based Effort Estimation for Software-Development Projects; Juli 1992.
- Nr. 12 Dornhoff, P.: Aufwandsplanung zur Unterstützung des Managements von Softwareentwicklungsprojekten; August 1992.
- Nr. 13 Eicker, S., Schnieder, T.: Reengineering; August 1992.
- Nr. 14 Erkelenz, F.: KVD2 - Ein integriertes wissensbasiertes Modul zur Bemessung von Krankenhausverweildauern - Problemstellung, Konzeption und Realisierung; Dezember 1992.
- Nr. 15 Horster, B., Schneider, B., Siedentopf, J.: Kriterien zur Auswahl konnektionistischer Verfahren für betriebliche Probleme; März 1993.
- Nr. 16 Jung, R.: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering: Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht; Juli 1993.
- Nr. 17 Miller, C., Weiland, R.: Der Übergang von proprietären zu offenen Systemen aus Sicht der Transaktionskostentheorie; Juli 1993.
- Nr. 18 Becker, J., Rosemann, M.: Design for Logistics - Ein Beispiel für die logistikgerechte Gestaltung des Computer Integrated Manufacturing; Juli 1993.
- Nr. 19 Becker, J., Rosemann, M.: Informationswirtschaftliche Integrationsschwerpunkte innerhalb der logistischen Subsysteme - Ein Beitrag zu einem produktionsübergreifenden Verständnis von CIM; Juli 1993.

- Nr. 20 Becker, J.: Neue Verfahren der entwurfs- und konstruktionsbegleitenden Kalkulation und ihre Grenzen in der praktischen Anwendung; Juli 1993.
- Nr. 21 Becker, K., Prischmann, M.: VESKONN - Prototypische Umsetzung eines modularen Konzepts zur Konstruktionsunterstützung mit konnektionistischen Methoden; November 1993
- Nr. 22 Schneider, B.: Neuronale Netze für betriebliche Anwendungen: Anwendungspotentiale und existierende Systeme; November 1993.
- Nr. 23 Nietsch, T., Rautenstrauch, C., Rehfeldt, M., Rosemann, M., Turowski, K.: Ansätze für die Verbesserung von PPS-Systemen durch Fuzzy-Logik; Dezember 1993.
- Nr. 24 Nietsch, M., Rinschede, M., Rautenstrauch, C.: Werkzeuggestützte Individualisierung des objektorientierten Leitstands ooL; Dezember 1993.
- Nr. 25 Meckenstock, A., Unland, R., Zimmer, D.: Flexible Unterstützung kooperativer Entwurfsumgebungen durch einen Transaktions-Baukasten; Dezember 1993.
- Nr. 26 Grob, H. L.: Computer Assisted Learning (CAL) durch Berechnungsexperimente; Januar 1994.
- Nr. 27 Kirn, St., Unland, R. (Hrsg.): Tagungsband zum Workshop "Unterstützung Organisatorischer Prozesse durch CSCW". In Kooperation mit GI-Fachausschuß 5.5 "Betriebliche Kommunikations- und Informationssysteme" und Arbeitskreis 5.5.1 "Computer Supported Cooperative Work", Westfälische Wilhelms-Universität Münster, 4.-5. November 1993.
- Nr. 28 Kirn, St., Unland, R.: Zur Verbundintelligenz integrierter Mensch-Computer-Teams: Ein organisationstheoretischer Ansatz; März 1994.
- Nr. 29 Kirn, St., Unland, R.: Workflow Management mit kooperativen Softwaresystemen: State of the Art und Problemabriß; März 1994.
- Nr. 30 Unland, R.: Optimistic Concurrency Control Revisited; März 1994.
- Nr. 31 Unland, R.: Semantics-Based Locking: From Isolation to Cooperation; März 1994.
- Nr. 32 Meckenstock, A., Unland, R., Zimmer, D.: Controlling Cooperation and Recovery in Nested Transactions; März 1994.
- Nr. 33 Kurbel, K., Schnieder, T.: Integration Issues of Information Engineering Based I-CASE Tools; September 1994.
- Nr. 34 Unland, R.: TOPAZ: A Tool Kit for the Construction of Application Specific Transaction; November 1994.
- Nr. 35 Unland, R.: Organizational Intelligence and Negotiation Based DAI Systems - Theoretical Foundations and Experimental Results; November 1994.
- Nr. 36 Unland, R., Kirn, St., Wanka, U., O'Hare, G.M.P., Abbas, S.: AEGIS: Agent oriented organisations; Februar 1995.
- Nr. 37 Jung, R., Rimpler, A., Schnieder, T., Teubner, A.: Eine empirische Untersuchung von Kosteneinflußfaktoren bei integrationsorientierten Reengineering-Projekten; März 1995.
- Nr. 38 Kirn, St.: Organisatorische Flexibilität durch Workflow-Management-Systeme?; Juli 1995.
- Nr. 39 Kirn, St.: Cooperative Knowledge Processing: The Key Technology for Future Organizations; Juli 1995.

- Nr. 40 Kirn, St.: Organisational Intelligence and Distributed AI; Juli 1995.
- Nr. 41 Fischer, K., Kirn, St., Weinhard, Ch. (Hrsg.): Organisationsaspekte in Multiagentensystemen; September 1995.
- Nr. 42 Grob, H. L., Lange, W.: Zum Wandel des Berufsbildes bei Wirtschaftsinformatikern, Eine empirische Analyse auf der Basis von Stellenanzeigen; Oktober 1995.
- Nr. 43 Abu-Alwan, I., Schlagheck, B., Unland, R.: Evaluierung des objektorientierten Datenbankmanagementsystems ObjectStore; Dezember 1995.
- Nr. 44 Winter, R., Using Formalized Invariant Properties of an Extended Conceptual Model to Generate Reusable Consistency Control for Information Systems; Dezember 1995.
- Nr. 45 Winter, R.: Design and Implementation of Derivation Rules in Information Systems; Februar 1996.
- Nr. 46 Becker, J.: Eine Architektur für Handelsinformationssysteme; März 1996.
- Nr. 47 Becker, J., Rosemann, M. (Hrsg.): Workflowmanagement - State-of-the-Art aus Sicht von Theorie und Praxis. Proceedings zum Workshop vom 10. April 1996. April 1996.
- Nr. 48 Rosemann, M., zur Mühlen, M.: Der Lösungsbeitrag von Metadatenmodellen beim Vergleich von Workflowmanagementsystemen; Juni 1996.