

Heft 95

W. Hoffmann, A.-W. Scheer, R. Backes

**Konzeption eines Ereignisklassifikationssystems
in Prozeßketten**

November 1992

Inhaltsverzeichnis

1.	Gründe für eine Ereignisklassifikation	1
2.	Einordnung der Ereignisklassifikation in das Vorgehensmodell zum Entwurf betrieblicher Informationssysteme	1
3.	Das Konzept eines Klassifikationssystems für Ereignistypen	4
3.1.	Das Klassifikationsverfahren	4
3.2.	Die Komponenten des Klassifikationssystems	5
3.3.	Die Facetten des Klassifikationssystems	7
3.3.1.	Die Abstraktionsebene	7
3.3.2.	Der Modellierungsaspekt	8
3.3.3.	Die Systemeigenschaft	10
3.3.4.	Das Eintrittsmerkmal	13
3.3.5.	Der Eintrittszeitpunkt	14
3.4.	Das Vorgehensmodell zur Ereignisklassifikation	15
3.5.	Das Datenmodell der Ereignisklassifikation	17
4.	Fazit und Ausblick	18
	Literaturverzeichnis	20

1. Gründe für eine Ereignisklassifikation

Mit der Methode der ereignisgesteuerten Prozeßkette werden Geschäftsprozesse eines Unternehmens modelliert, wobei insbesondere der zeitlich, logische Ablauf der Unternehmensfunktionen dargestellt wird. Hierbei existieren in der Regel sowohl Funktionen, die bereits computerunterstützt bearbeitet werden als auch solche, die rein manueller Natur sind. Mit Hilfe einer Klassifikation existierender Ereignistypen soll eine soweit formalisierte und strukturierte Darstellung entwickelt werden, mit deren Hilfe Integritäts- und Konsistenzprüfungen von Prozeßketten hinsichtlich ihrer Syntax ermöglicht wird. Dabei muß im Rahmen der Integritätsprüfung untersucht werden, welche Ereignistypen generell existieren, d.h. welche Klassen von Ereignistypen mit welchen Eigenschaften überhaupt gebildet werden können.

Weiterhin müssen entwickelte Prozeßketten auf ihre Konsistenz überprüft werden, um ihre Widerspruchsfreiheit zu garantieren. Damit kann eine korrekte Verknüpfung der Prozeßketten ermöglicht und gegenseitige Prozeßabhängigkeiten dargestellt werden.

Neben diesen modellierungstechnischen Aspekten existiert noch ein anderer Grund, ein Ereignisklassifikationssystem zu entwickeln. Ein Ereignistyp innerhalb der ereignisgesteuerten Prozeßkette wird definiert, als "eine eindeutig benannte Sammlung von Ereignissen, die aufgrund des Eintretenseins von Ausprägungen derselben Attribute einer Klasse zugeordnet werden. Der Ereignistyp ist eine Generalisierung von internem und externem Ereignis" [1]. Man sieht also, daß hier generell von einem Ereignistyp gesprochen wird ohne weitere Klassifikationsgesichtspunkte zu berücksichtigen und somit keinerlei Detaillierung bzgl. der verschiedenen Ereignistypen existiert. Mit Hilfe eines entsprechenden Klassifikationssystems soll ein Ansatz geliefert werden, Ereignistypen besser strukturieren zu können, d.h. eine gewisse "Ordnung" innerhalb dieser Objekte zu schaffen.

2. Einordnung der Ereignisklassifikation in das Vorgehensmodell zum Entwurf betrieblicher Informationssysteme

Ausgangspunkt beim Entwurf betrieblicher Informationssysteme sind die Ziele, die durch den Einsatz dieses Systems erreicht werden sollen. Über diese Ziele muß zwischen allen involvierten Gruppen - Management, Entwickler und Fachabteilungen - ein Konsens erreicht werden, um so spätere Akzeptanzschwierigkeiten von vornherein vermeiden zu helfen. Ein Vorgehensmodell, das auf die individuellen Gegebenheiten des Unternehmens abgestimmt ist kann dabei als grundlegendes Mittel zur Erreichung dieser Ziele angesehen werden [2].

[1] Vgl. Baresch, Keller, Kern, Kirsch: *Methodenhandbuch zur Funktions-/Prozeßmodellierung; Kooperationsprojekt IWI/IDS/SAP April 1991 S.15*

[2] Vgl. Schönthaler, F.: *Software - Entwicklungswerkzeuge, Methodische Grundlagen, In: Leitfäden der angewandten Informatik, Stuttgart 1990 S. 328*

Das entwickelte Meta-Modell umfaßt gemäß der ARIS-Architektur [3] die Daten-, Funktions- und Steuerungssicht, wobei aus Vereinfachungsgründen die Organisationssicht nicht näher betrachtet wird. Der zeitlich, logische Ablauf des Projektes Entwurf eines betrieblichen Informationssystems wird in der Steuerungssicht dargestellt, wobei das Gesamtprojekt in drei Teilprojekte untergliedert werden kann:

- (1) Entwicklung des Datenmodells
- (2) Entwicklung des Funktionsmodells
- (3) Entwicklung des Prozeßmodells

Während bzw. vor Beginn der Modellierung der Teilbereiche wird die anzuwendende Methode definiert. Methoden sind Vorschriften, wie planmäßig nach einem bestimmten Prinzip (oder einer Kombination von Prinzipien) zur Erreichung festgelegter Ziele vorzugehen ist [4].

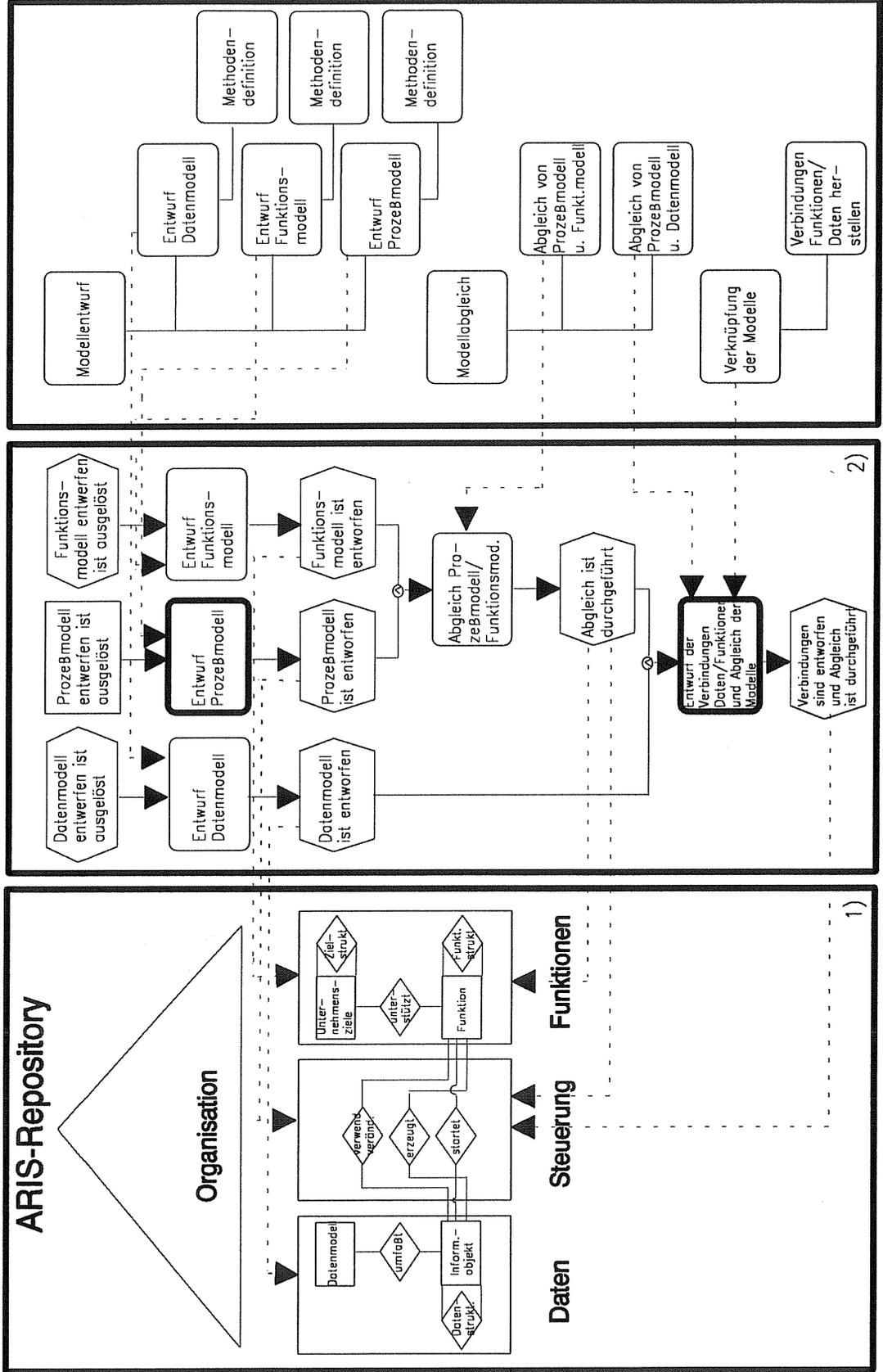
Für die einzelnen Aktivitäten des Vorgehensmodells werden nun geeignete Methoden ausgewählt. Wichtig ist, darauf zu achten, daß sich die ausgewählten Methoden im Rahmen des Vorgehensmodells kombinieren lassen. Da dies nicht immer problemlos sein wird empfiehlt es sich, die Definition des Vorgehensmodells und die Methodenauswahl in einem zyklischen Prozeß durchzuführen [5].

Nachdem die Modelle entworfen sind, werden sie in den entsprechenden Teilen des Repositories abgelegt (siehe Abb. 1). Anschließend erfolgt ein Abgleich zwischen Funktions- und Prozeßmodell, wo die in beiden Modellen auftretenden Funktionen untersucht, verglichen und gegebenenfalls vereinheitlicht werden. Ist dieser Abgleich durchgeführt, so wird das im Repository bereits gespeicherte Funktions- bzw Prozeßmodell entsprechend geändert. Analog wird nun beim Abgleich zwischen Prozeßmodell und Datenmodell vorgegangen, d.h. es wird überprüft ob die vorliegenden Ereignisse auf entsprechende Informationsobjekte referenzieren. In der folgenden Funktion werden die Verbindungen zwischen Daten und Funktionen hergestellt, wobei die Start- und Endereignisse der einzelnen Funktionen, die als "Daten" im Repository abgelegt sind, festgelegt werden. Das Startereignis muß hierbei nicht mit den gesamten Inputdaten der Funktion identisch sein, d.h. eine bestimmte Datenveränderung reicht aus, um die Funktion auszulösen. Die sonstigen Inputdaten sind ebenfalls im Repository abgelegt.

[3] Vgl Scheer, A.-W.: *Architektur integrierter Informationssysteme*, Berlin et al. 1991

[4] Vgl. Stahlknecht, P.: *Einführung in die Wirtschaftsinformatik* Berlin et al. 1989 S.222

[5] Vgl. Schönthaler, F.: *Software - Entwicklungswerkzeuge Methodische Grundlagen*. In: *Leitfäden der angewandten Informatik*. Stuttgart 1990 S.329



Funktionssicht

Steuerungssicht

Datensicht

1) Quelle: Scheer
 2) Quelle: Keller/Nüttgens/Scheer

Abbildung 2: Das Vorgehensmodell (Meta-Modell)

Es stellt sich die Frage welche Funktionen innerhalb des Entwurfs betrieblicher Informationssysteme durch die Ereignisklassifikation unterstützt werden.

Im Rahmen der Funktion "Entwurf der Prozeßmodelle" können mit Hilfe dieser Klassifikation grobe syntaktische Fehler vermieden werden, da insbesondere die Integrität der Prozeßketten bzgl. der Ereignistypen bereits hier überprüft werden kann. Es besteht die Möglichkeit nicht zulässige Eigenschaftskombinationen von Ereignistypen (siehe Kapitel 3.5) zu erkennen und zu eliminieren. Auch beim Abgleich der Prozeßmodelle innerhalb der Funktion "Entwurf der Verbindungen Daten/Funktionen und Abgleich der Modelle" kann mit Hilfe eines entsprechenden Klassifikationssystems die Existenz widerspruchsfreier und damit konsistenter Prozeßketten auch über die Systemgrenze hinaus garantiert werden.

3. Das Konzept eines Klassifikationssystems für Ereignistypen

3.1. Das Klassifikationsverfahren

Das Konzept der Ereignisklassifikation basiert auf einem synthetischen Klassifikationsverfahren (Facettenklassifikation), da eine hierarchische Ereignisklassifikation zu Redundanzen führt (siehe Abbildung 2) und dem Anspruch der Vollständigkeit genügen sollte. Ergeben sich nämlich nach Beendigung des Klassifikationsprozesses neue Erkenntnisse, so können diese nicht ohne weiteres in einem hierarchischen Klassifikationssystem berücksichtigt werden, da nur verwendet werden darf, was explizit im Klassifikationssystem aufgeführt ist und neue Klassen nur sehr schwierig in das Klassifikationssystem zu integrieren sind.

Durch das Festlegen bestimmter Facetten besteht die Möglichkeit grundsätzliche Eigenschaften von Ereignistypen zu definieren. Diese Facetten sind disjunkt und monodimensional untergliedert, d.h. es existiert pro Facette nur ein einziges Unterteilungskriterium, das identisch ist mit der entsprechenden "Haupteigenschaft" des Ereignistyps. Ein solches Klassifikationssystem kann durch Definition einer neuen Eigenschaft (Facette) ohne Probleme ergänzt werden. Durch die Kombination der Ausprägungen der einzelnen Facetten werden die Ereignistypen klassifiziert, wobei Kombinationsregeln angegeben werden müssen damit festgelegt werden kann auf welche Weise die einzelnen Teile miteinander kombiniert werden dürfen oder müssen. Dies wird nun an dem entwickelten Ereignisklassifikationssystem konkretisiert und erläutert.

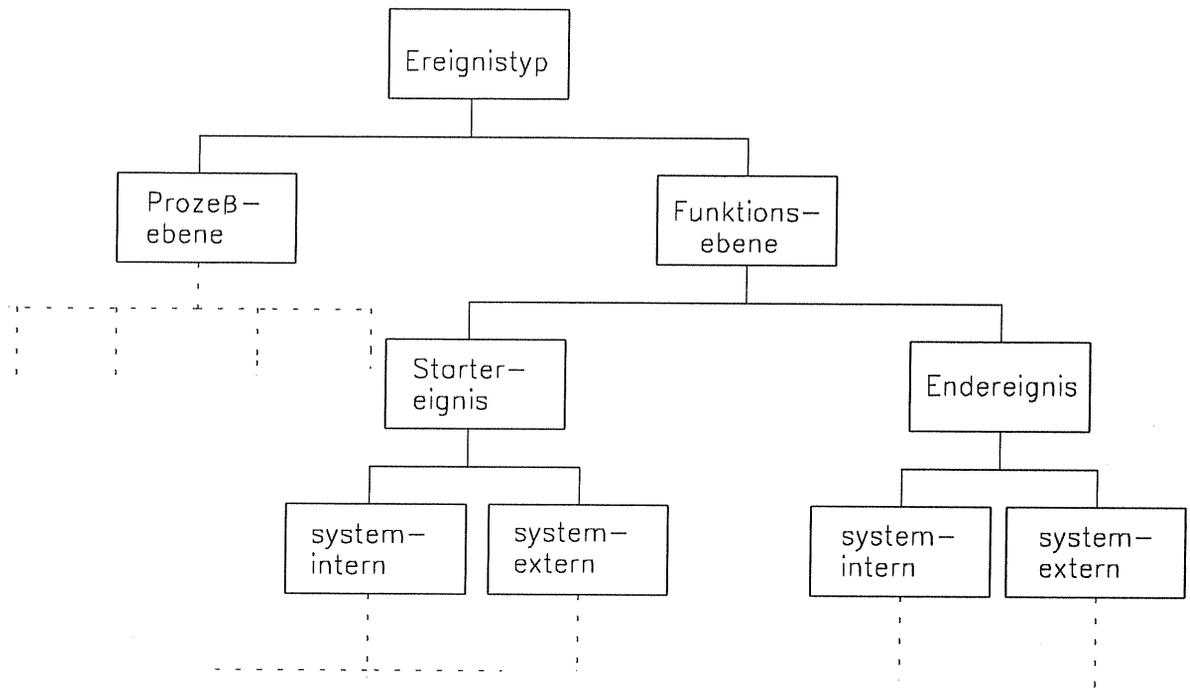


Abb. 3.: Hierarchische Ereignisklassifikation

3.2. Die Komponenten des Klassifikationssystems

Für die Ereignisklassifikation werden **Facetten** festgelegt, die übergeordnete Ereigniseigenschaften darstellen. Es sind die Facetten,

- (A) Abstraktionsebene
- (B) Modellierungsaspekt
- (C) Systemeigenschaft
- (D) Eintrittsmerkmal
- (E) Eintrittszeitpunkt

Diesen einzelnen Facetten werden die **Ausprägungen**,

- (1) Prozeß
- (2) Funktion
- (3) Starterereignis
- (4) Zwischenereignis
- (5) Endereignis
- (6) Start-/Endereignis
- (7) systemintern
- (8) systeminterdependent

- (9) systemextern
- (10) automatisiert
- (11) interaktiv
- (12) manuell
- (13) bestimmt
- (14) unbestimmt

entsprechend zugeordnet, was zu einem Klassifikationssystem führt. (siehe Abbildung 4)
 Die Strukturierung eines Ereignistyps entsteht durch Kombination der Ausprägungen der einzelnen Facetten, wobei immer genau eine Ausprägung pro Facette in die entsprechende Kombination eingehen muß. So ergibt z.B. die Kombination A1-B1-C1-D1-E1 ein Ereignistyp auf Prozeßebene, der als Startereignis systemintern, automatisiert und bestimmt ist.

Die Facetten und ihre Ausprägungen, sowie deren Kombinationsmöglichkeiten werden in den nächsten Kapiteln genauer beschrieben.

Klassifikationssystem für Ereignistypen (Facettenklassifikation)	
Facette A: Abstraktionsebene	A1 Prozeß A2 Funktion
Facette B: Modellierungsaspekt	B1 Startereignis B2 Endereignis B3 Start-/Endereignis B4 Zwischenereignis
Facette C: Systemeigenschaft	C1 systemintern C2 systeminterdependent/ intern C3 systemextern
Facette D: Eintrittsmerkmal	D1 automatisiert D2 interaktiv D3 manuell
Facette E: Eintrittszeitpunkt	E1 bestimmt E2 unbestimmt

Abb. 4: Klassifikationssystem für Ereignistypen

3.3. Die Facetten des Klassifikationssystems

3.3.1. Die Abstraktionsebene

Innerhalb der Prozeßmodellierung bzw. der ereignisgesteuerten Prozeßkette lassen sich zwei grundsätzliche Abstraktionsebenen feststellen (siehe Abbildung 5).

Auf übergeordneter Ebene existieren Geschäftsprozesse z.B. die Kundenauftragssteuerung, die wiederum in Teilprozesse z.B. Kundenangebotsbearbeitung, Angebotsverfolgung untergliedert werden können. Diese Abstraktion könnte man beliebig lange weiterführen, wobei sich die Frage stellt, wann es betriebswirtschaftlich noch sinnvoll ist weiter zu untergliedern. Ein Prozeß bei dem eine weitere Untergliederung nicht sinnvoll erscheint wird als Elementarprozeß bzw. Elementarfunktion bezeichnet. Ein Beispiel hierfür ist die Vorkalkulation im Rahmen der Auftragserstellung.

In jedem Prozeß müssen Funktionen ausgeführt werden, die durch bestimmte Ereignistypen ausgelöst werden. Man sieht, daß Ereignisse zum einen Prozesse auslösen können und zum anderen lediglich Anstoß einer Funktion innerhalb eines Prozesses sein können. Durch Konstruktion beliebiger Sichten (views) kann der Gesamtprozeß der betrieblichen Leistungserstellung in Prozesse, Teilprozesse und Funktionen untergliedert werden, wobei die Bildung dieser Sichten ein intellektueller Vorgang ist und somit mehrere Ansätze möglich sind. Mit Hilfe der Ereignisklassifikation kann ein Ereignistyp den Abstraktionsebenen **Prozeß** bzw. Teilprozess oder **Funktion** zugeordnet werden.

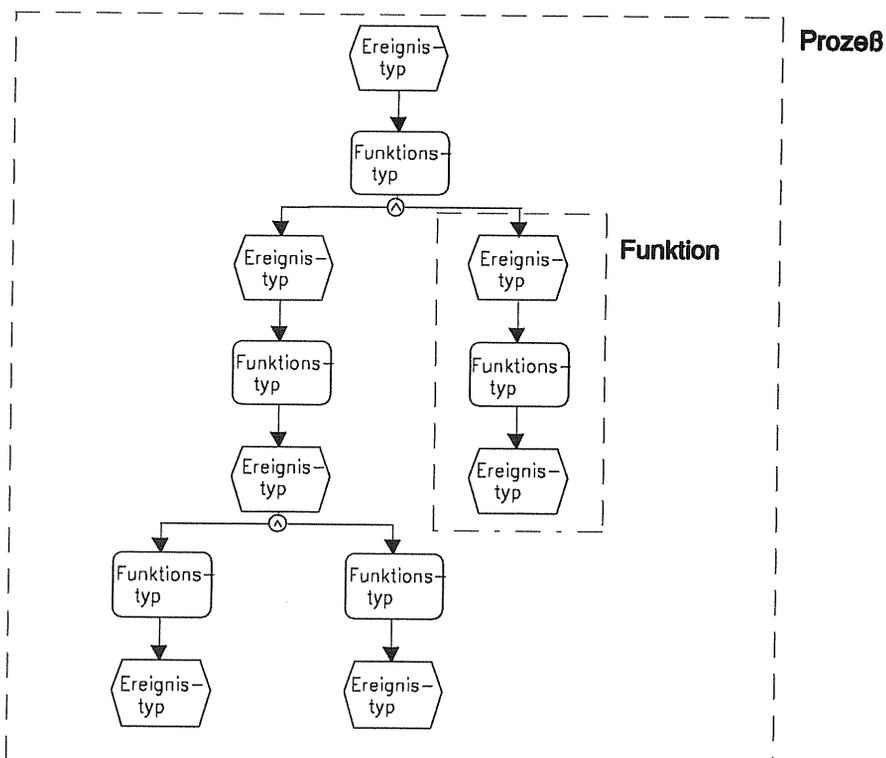


Abb. 5: Abstraktionsebenen

3.3.2. Der Modellierungsaspekt

Bei der Facette Modellierungsaspekt wird untersucht, welcher zeitliche Aspekt bzw. Zusammenhang zwischen Ereignistyp und Funktionstyp besteht, wobei dies in Abhängigkeit von der Abstraktionsebene zu sehen ist.

Auf Prozeßebene ist ein Ereignis eindeutig als **Startereignis** definiert, wenn durch dieses Ereignis die erste Funktion eines Prozesses ausgelöst wird, während innerhalb des Prozesses ein Startereignis einer Funktion gleichzeitig ein Endereignis der vorherigen Funktion ist und somit als **Start-/Endereignis** bezeichnet wird. Jedes Ereignis kann aber auch ein **Zwischenereignis** innerhalb einer Prozeßkette sein, das eine Funktion innerhalb eines anderen Prozesses auslöst (siehe Abbildung 6). Somit stellt ein Zwischenereignis eine Schnittstelle zwischen zwei verschiedenen Prozessen dar, womit die Verknüpfung von Funktionen über die "Prozeßgrenze" hinaus aufgezeigt werden kann.

Ein reines **Endereignis** liegt vor, wenn dieses Ereignis im Unternehmen keinen anderen Prozeß mehr auslöst, z.B. Produkt ist an Kunde ausgeliefert, wobei hierbei unternehmensspezifische Gesichtspunkte beachtet werden müssen, da z.B. im Anlagenbau mit Sicherheit das Ereignis "Produkt ist an Kunde ausgeliefert" kein Endereignis ist sondern wiederum Startereignis z.B. für den Prozeß "Anlageninstallation" ist. Man sieht also, daß "reine" Endereignisse auf Prozeßebene sehr stark branchenabhängig sind.

Auf Funktionsebene wird dagegen eine einzelne Funktion untersucht, wobei der zugehörige Prozeß nicht weiter berücksichtigt wird.

Es werden also zeitlich vor- und nachgelagerte Funktionen in die Untersuchung nicht mit einbezogen. Dies hat zur Folge, daß einer Elementarfunktion eindeutig Start- und Endereignis zugeordnet werden können.

Zwischenereignisse können hierbei nicht auftreten, da es sich in diesem Falle gerade um die Verknüpfung verschiedener Prozesse handelt und damit die (Elementar-) Funktionsebene verlassen wird. Man sieht also, daß in Abhängigkeit von der Abstraktionsebene verschiedene Modellierungsaspekte zu beachten sind.

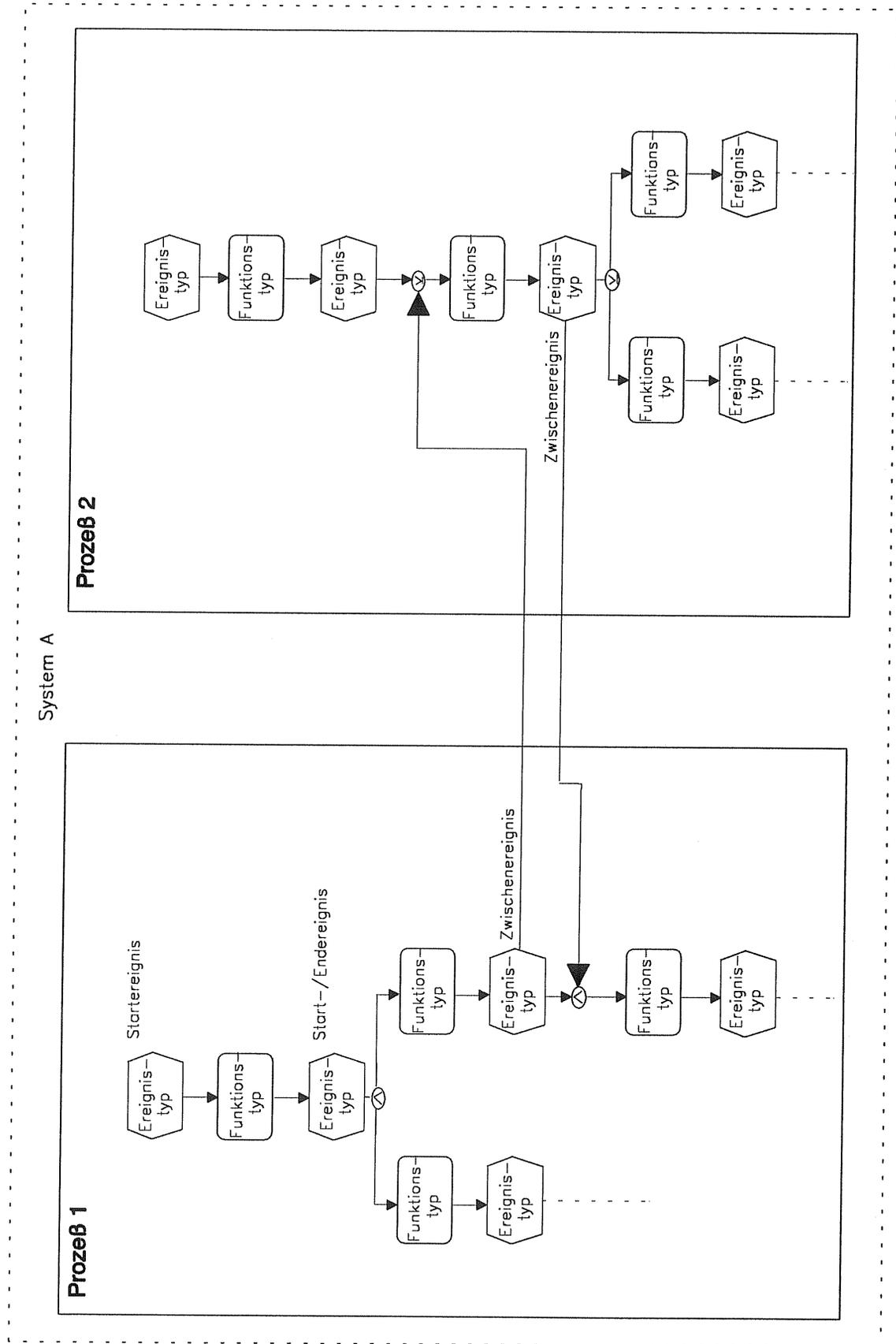


Abb. 6: systeminterne Prozeßverknüpfung über Zwischenereignisse

3.3.3. Die Systemeigenschaft

Unter der Systemeigenschaft wird festgelegt, welche Auswirkungen das Eintreten eines Ereignisses auf ein computergestütztes System bzw. dessen Systemumgebung hat. Bei der Betrachtung dieser Eigenschaft bewegen wir uns auf DV- Konzeptebene bzw. im Schnittstellenbereich zwischen Fachkonzept und DV-Konzept, da es sinnvoll erscheint gerade der Umsetzungsproblematik vom Fachkonzept in das DV-Konzept Rechnung zu tragen und auch solche Aspekte in das Klassifikationssystem einzubinden. Hierbei wird zwischen **systeminternem, systeminterdependentem und systemexternem Ereignistyp** unterschieden. Systeminterne Ereignisse haben nur Auswirkungen auf Funktionen im inneren eines bestimmten computergestützten Systems (z.B. PPS-System), wobei keinerlei Schnittstellen zu anderen Systemen (z.B. Leitstand) existieren. Systeminterdependente Ereignisse dagegen haben gerade Auswirkungen auf andere computerunterstützte Systeme, d.h. treten systeminterdependente Ereignisse auf, so existieren Abhängigkeiten bzw. Schnittstellen zwischen einzelnen Informationssystemen, was bei der Schaffung der konzeptionellen Basis integrierter Systeme berücksichtigt werden muß.

Systemexterne Ereignisse dagegen treten nur in der natürlichen Umgebung computergestützter Systeme auf und können diese aktivieren. Eine schriftlicher Kundenauftrag ist z.B. ein systemexternes Ereignis, das das PPS-System, insbesondere das Teilmodul Auftragssteuerung auslöst, wobei der zuständige Sachbearbeiter die entsprechenden Daten dem System erst zuführen muß (Datenerfassung).

Die mit Sicherheit interessanteste Komponente ist das systeminterdependente Ereignis, das im folgenden näher untersucht werden soll.

In der Realität wird eine komplexe Aufgabe, wie z.B. die betriebliche Leistungserstellung häufig mit Hilfe verschiedener computergestützten Systeme (PPS-System, Leitstand, Lagerverwaltungssystem, flexible Fertigungszellen usw.) durchgeführt. CIM verfolgt dabei das Ziel alle fertigungstechnischen Arbeitsabläufe sowie alle betriebswirtschaftlich-organisatorischen Dispositions- und Steuerungsaufgaben als Bestandteil eines völlig neu strukturierten Gesamtsystems zusammenzufassen [6]. Bevor solche Systeme jedoch "integriert arbeiten" können, müssen ihre Interdependenzen untersucht werden, die durch den Entwurf und die Analyse der betrieblichen Prozeßketten aufgezeigt werden. Insbesondere durch das Erkennen von systeminterdependenten Ereignissen kann die Konsistenz betrieblicher Prozeßketten über die Systemgrenze hinaus untersucht werden.

Betrachtet man zwei unterschiedliche Systeme (siehe Abbildung 7), so kann durch ein systeminterdependentes Ereignis innerhalb einer Prozeßkette des Systems A, eine Funktion innerhalb einer Prozeßkette des Systems B ausgelöst werden. Dies kann eine Elementarfunktion oder ein Prozeß sein. Es besteht also die Möglichkeit, daß eine Funktion innerhalb eines Prozesses eines Systems A erst dann durchgeführt wird, wenn ein bestimmtes

[6] Vgl. Stahlknecht, P.: *Einführung in die Wirtschaftsinformatik* Berlin et al. 1989 S.331

Ereignis eines Prozesses des Systems B eintritt (Trigger). Somit ist ein systeminterdependentes Ereignis immer systemintern, ein systeminternes Ereignis jedoch nicht immer systeminterdependent.

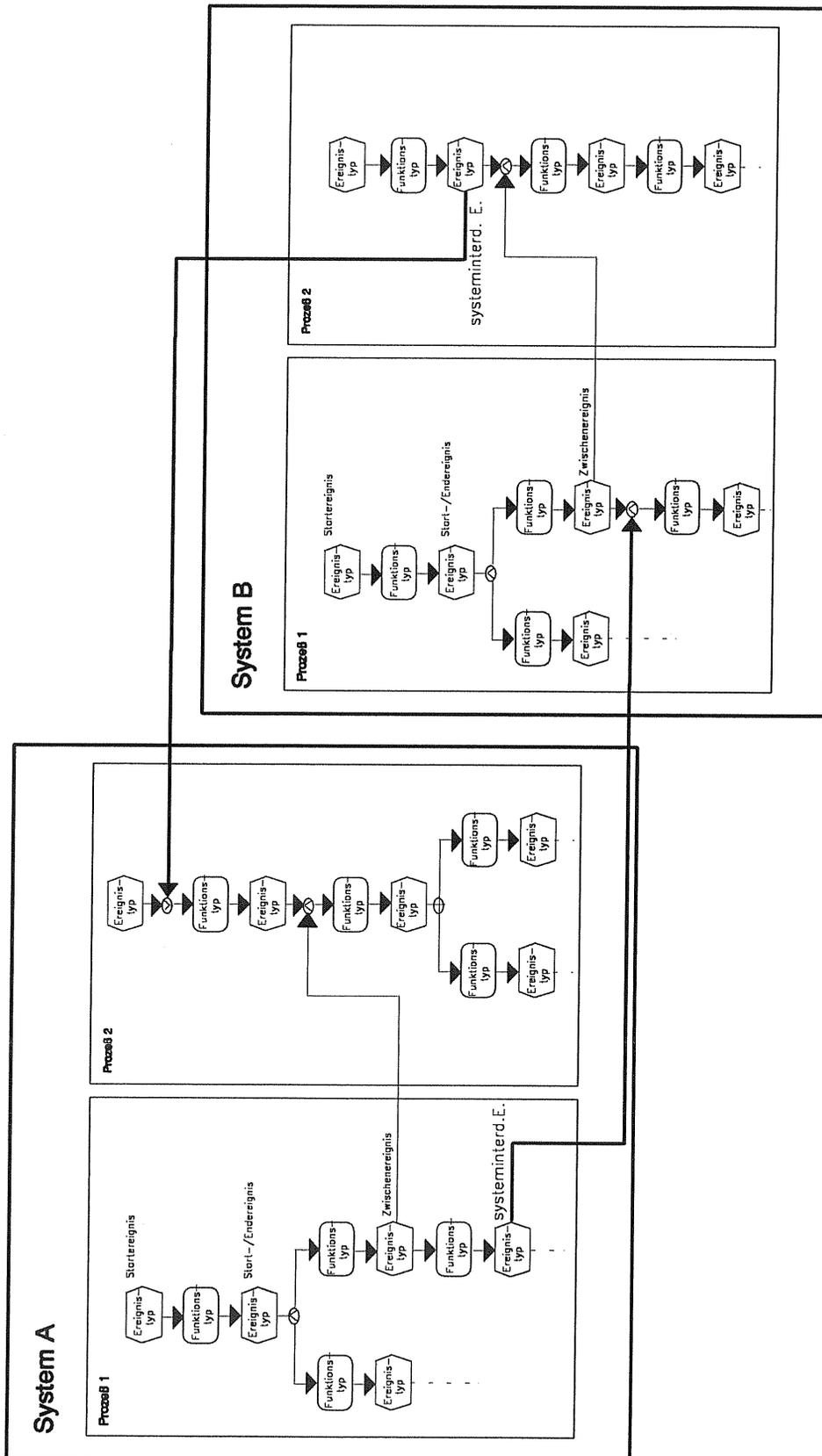


Abb. 7: systeminterdependente Ereignisse

Durch die Analyse von Prozeßketten bzgl. systeminterdependenter Ereignisse die Schnittstellen zwischen Systemen aufgezeigt und untersucht werden können und somit eine Konsistenzprüfung der Prozeßketten unterstützt wird.

Durch eine Analyse betrieblicher Prozeßketten über die Systemgrenzen hinaus besteht die Möglichkeit Prozesse oder Funktionen, die durch systeminterdependente Ereignisse gestartet werden, in das auslösende System zu verlagern, um somit einen zuvor systeminterdependenten Prozeß nun systemintern durchführen zu können und damit die Systemabhängigkeiten zu verringern.

Gerade die Schnittstellenproblematik tritt bei den Unternehmen immer mehr in den Vordergrund, da die Rationalisierungspotentiale bzgl. der funktionalen Kriterien von Software-Systemen weitestgehend ausgeschöpft sind und eine konsistente Datenverwaltung d.h. eine Datenintegration der heterogenen EDV-Systeme in der Regel nicht existiert. Die benötigte Daten werden dezentral und teilweise redundant in den unterschiedlichen Systemen gehalten, was eine konsistente Datenverwaltung unmöglich macht und dem CIM-Gedanken widerspricht. Dies hat zur Folge, daß die Datenübergabe von einem System zu einem anderen häufig noch über Papierformulare erfolgt. Da die Realisierung einer gemeinsamen CIM-Datenbasis kurzfristig nicht zu erwarten ist wird eine individuell konfigurierbare Schnittstelle benötigt, die unterschiedliche heterogene Systeme miteinander verbindet [7]. Man sieht also, daß gerade die Betrachtung von Systeminterdependenzen immer wichtiger wird und auch bei dem Entwurf betrieblicher Prozeßketten zur Überwindung von Systemgrenzen erforderlich ist.

Mit Hilfe des Integrationswerkzeugs INMAS (Interface Management System), das im Rahmen des CIDAM-Projektes am Institut für Wirtschaftsinformatik an der Universität Saarbrücken in Zusammenarbeit mit Industriepartnern entwickelt wird, soll die Konsistenz des Datenbestandes eines Unternehmens mit heterogenen EDV-Systemen sichergestellt werden. INMAS ist eine frei konfigurierbare Schnittstelle, wobei die einzelnen Systeme durch eine neutrale Datenstruktur miteinander verbunden werden. Grundlage hierfür ist ein Referenz-Datenmodell, das die logische Konsistenz verteilter Datenbestände sicherstellt. Ändern sich durch eine Funktion Datenbestände eines EDV-Systems, die in anderen EDV-Systemen redundant gespeichert werden, so sendet die datenverändernde Funktion nach ihrer Beendigung eine Trigger-Nachricht an INMAS, woraufhin mit Hilfe von Aktivitätenketten die entsprechenden Datenänderungen in den anderen Systemen durchgeführt werden. Das Endereignis der datenverändernden Funktion ist somit das Startereignis für INMAS und kann bzgl. der Systemeigenschaft als systeminterdependent bezeichnet werden [8].

[7] Vgl. Herterich, R., Klein, J., Scheer, A.-W.: *INMAS - Eine individuell konfigurierbare Schnittstelle*. In: *Information Management 1/1990* S.16-26, s.bes.S. 16/17

[8] Vgl. Hoffmann, W., Nüttgens, M., Scholz, S., Scheer, A.-W.: *Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil I Produktionsplanung)*. In: Scheer, A.-W. (Hrsg.): *Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 85, Saarbrücken 1991, S.3*

3.3.4. Das Eintrittsmerkmal

Das Eintrittsmerkmal eines Ereignisses beschreibt die Art und Weise wie es eintritt. So kann ein **automatisiertes Ereignis** dadurch eintreten, daß innerhalb einer Funktion ein bestimmter Schwellwert erreicht wird. So kann z.B. das Unterschreiten des Mindestbestandes einer Lagerressource automatisch das Ereignis "Bedarf an Lagerressource ist aufgetreten" generieren. Ebenso kann dieses Ereignis aus einem interaktiven Vorgang resultieren, und zwar wenn ein Mitarbeiter das Unterschreiten des Lagerbestandes z. B. durch visuelle Überprüfung feststellt und eine entsprechende Bestellanforderung über PC an die Beschaffungsabteilung sendet. Diese Ereignis wird als **interaktives Ereignis** definiert. Ereignisse, die keinen direkten Bezug zu computergestützten Systemen haben, wie z.B. "Telefonanruf ist angekommen" oder "Kundenauftrag per Post ist eingegangen", werden als **manuelle Ereignisse** bezeichnet.

3.3.5. Der Eintrittszeitpunkt

Ein Ereignis kann zu einem bestimmtem Zeitpunkt eintreten und entsprechende Funktionen aktivieren, z.B. wird am 15. jeden Monats der Prozeß der Lohn- und Gehaltsabrechnung gestartet, oder am 30.12. wird eine Inventur durchgeführt. In allen diesen Fällen ist also der Auslösefaktor ein ganz **bestimmtes Datum**. Ereignisse treten jedoch in der Regel zu unbestimmten Zeitpunkten auf, z.B. Bedarf für Lagerressource ist aufgetreten oder Verfügbarkeitsprüfung ist durchgeführt, d.h. sie treten aufgrund der Beendigung von vorgelagerten Funktionen auf und sind daher **zeitlich unbestimmt**, bzw. nicht an ein bestimmtes Datum gebunden.

Ein Ereignistyp kann nun mit Hilfe der oben beschriebenen Facetten bzw. deren Ausprägungen klassifiziert werden. Dabei wird jedem Ereignistyp genau eine Eigenschaftsausprägung pro Facette zugeordnet, d.h. zwischen den Ausprägungen existiert eine eindeutige "Oder-Verknüpfung". Diese Kombinationsmöglichkeiten können in einer Matrix abgebildet werden (siehe Abbildung 8).

So kann man bei der Analyse entwickelter Prozeßketten durch Ankreuzen der jeweiligen Eigenschaften die Ereignistypen klassifizieren. Neben der Frage der möglichen Kombinationen stellt sich aber auch die Frage, welche Kombinationen niemals auftreten dürfen. Dieses Problem wird durch das folgenden Vorgehensmodell gelöst.

Facette/ Auspr. Ereign. typen	Abstraktionsebene		Modellierungsaspekt				Systemeigenschaft			Eintrittsmerkmal			Eintrittszeitpunkt	
	Prozeß	Funktion	Start- ereignis	End- ereignis	Zwischen- ereignis	Start/ End.E.	system- intern	system- interdep.	system- extern	auto- matisiert	manuell	inter- aktiv	bestimmt	unbe- stimmt
E.typ1	X		X				X			X			X	
E.typ2		X		X			X				X			X
E.typ3		X	X						X		X			X
E.typ4	X					X			X					X
E.typ5	X				X		X			X			X	
E.typ6		X		X						X			X	
E.typ7	X		X						X					X
E.typ8	X				X				X			X		X
E.typ9		X	X				X					X	X	

Abb. 8: Klassifikationsmatrix für Ereignistypen

3.4. Das Vorgehensmodell zur Ereignisklassifikation

Da die oben beschriebenen Zusammenhänge, insbesondere die Kombinationsmöglichkeiten der Facetten und der Modellierungsaspekt sehr komplex sind wird zur Verdeutlichung ein Vorgehensmodells zur entwickelt (siehe Abbildung 9), in dem die einzelnen Schritte der Ereignisklassifikation verdeutlicht werden.

Tritt ein Ereignistyp während dem Entwurf oder dem Abgleich einer ereignisgesteuerten Prozeßkette auf, so wird zunächst die Abstraktionsebene bestimmt. Hierbei besteht die Möglichkeit, daß wir uns auf Prozeß- oder Funktionsebene befinden. Das weitere Vorgehen ist nun abhängig von der ausgewählten Abstraktionsebene, da bei der folgenden Funktion "Ereignistyp bzgl. Modellierungsaspekt klassifizieren" unterschiedliche Gesichtspunkte zu beachten sind. Während auf Prozeßebe der zu klassifizierende Ereignistyp, Startereignis, Endereignis, Start/Endereignis oder Zwischenereignis sein kann, existieren auf Funktionsebene lediglich die Ausprägungen Start- und Endereignis. Auch bei dem nächsten Schritt, nämlich der Klassifizierung des Ereignistyps bzgl. der Systemeigenschaft sind die einzelnen Ausprägungen von der Abstraktionsebene abhängig. Auf Prozeßebe können neben systeminternen und -externen Ereignistypen auch systeminterdependente Ereignisse auftreten, während auf (Elementar)Funktionsebene eine Systeminterdependenz nicht möglich ist, da hier gerade Funktionen oder Prozesse miteinander verknüpft werden. In der folgenden Funktion "Ereignistyp bzgl. Eintrittsmerkmal klassifizieren" bedarf es keiner parallelen Vorgehensweise, da hier unabhängig von der Abstraktionsebene vorgegangen werden kann. Die Eintrittsmerkmale automatisiert, interaktiv oder manuell treten sowohl auf Prozeßebe, als auch auf Funktionsebene auf. Gleiches gilt für den Eintrittszeitpunkt mit den möglichen Ausprägungen bestimmt oder unbestimmt. In der letzten Funktion des Vorgehensmodells werden nun die vorher bestimmten Eigenschaften des Ereignistyps durch Angabe der entsprechenden Kombination festgelegt. Damit kann die Klassifikationsmatrix ausgefüllt werden, wobei unzulässige Kombinationsmöglichkeiten nicht mehr auftreten können.

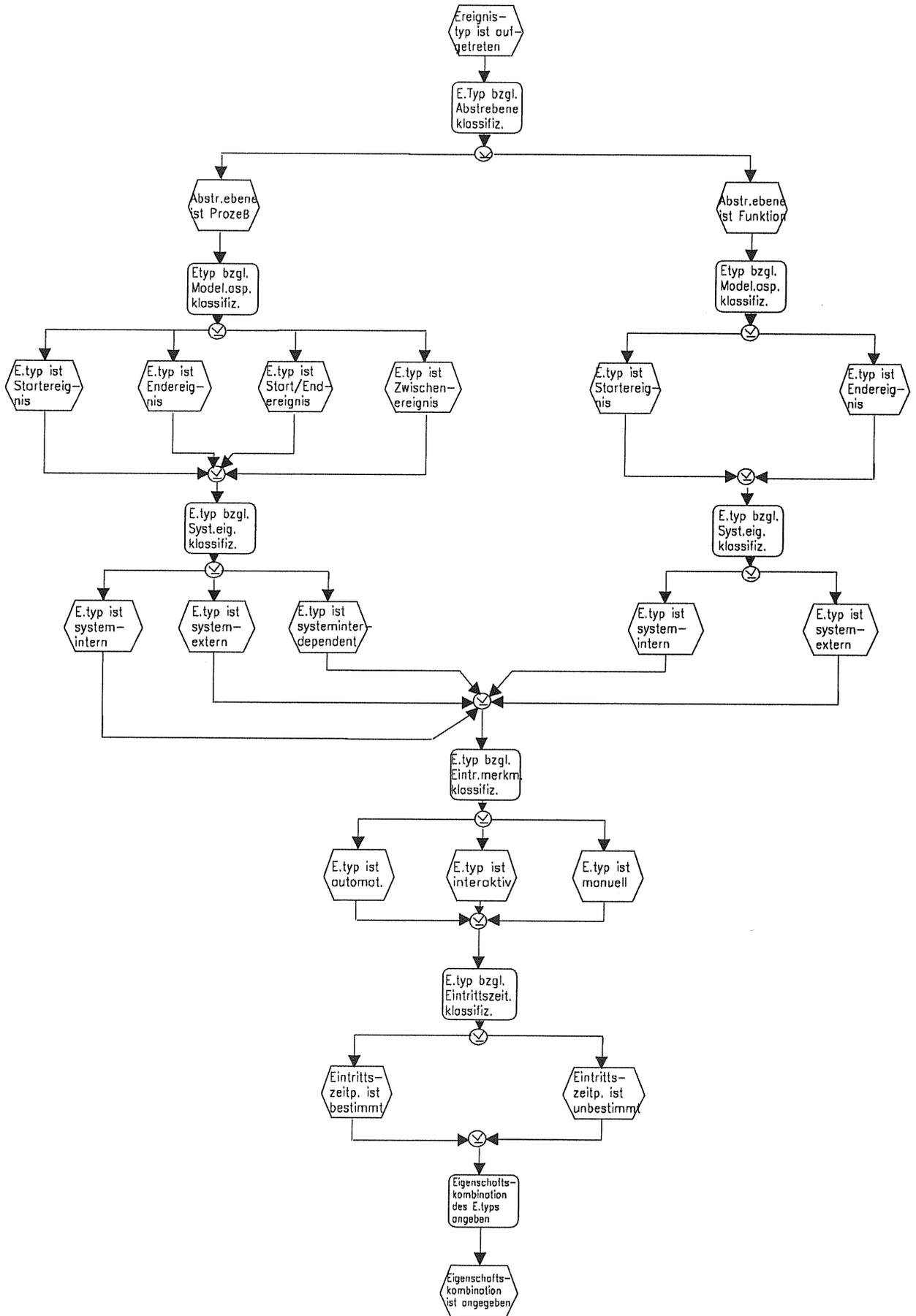


Abb. 9: Vorgehensmodell der Ereignisklassifikation

3.5. Das Datenmodell der Ereignisklassifikation

Im folgenden wird nun das Datenmodell für die Ereignisklassifikation entwickelt (siehe Abbildung 10)

Ausgangspunkt ist der Entitytyp Ausprägung, der durch die Ausprägungsnummer eindeutig identifiziert wird. Beschreibende Attribute sind dabei Bezeichnung und Beschreibung. Ausprägungen sind, wie oben beschrieben z.B. Startereignis und Zwischenereignis. Diese Ausprägungen werden nun über den Beziehungstyp Facetten/Ausprägungszuordnung den entsprechenden Facetten zugeordnet, wobei hier eine 1:n Beziehung vorliegt, d.h. eine Ausprägung ist genau einer Facette zugeordnet, eine Facette kann aber mehrere Ausprägungen besitzen. So hat z.B. die Facette Systemeigenschaft die Ausprägungen systemintern, systeminterdependent und systemextern; diese Ausprägungen können jedoch keiner anderen Facette zugeordnet werden. Der Entitytyp Facette wird durch die Facettennummer eindeutig identifiziert und durch die Attribute Beschreibung und Bezeichnung beschrieben.

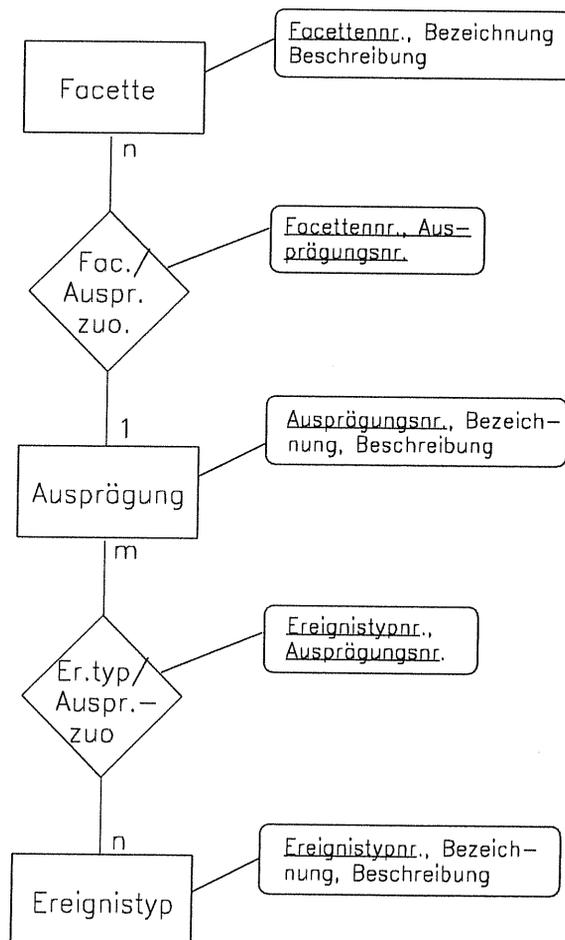


Abb. 10: Datenmodell der Ereignisklassifikation

Die einzelnen Ausprägungen werden nicht nur Facetten, sondern auch Ereignistypen zugeordnet. Durch den Beziehungstyp Ereignistyp/Ausprägungszuordnung wird gerade die Klassifizierung des Ereignistyps durchgeführt. Hier besteht eine n:m Beziehung, da ein Entitytyp verschiedene Ausprägungen besitzt und umgekehrt eine Ausprägung verschiedenen Ereignistypen zugeordnet werden kann. Der Entitytyp Ereignistyp wird durch die Ereignistypnummer identifiziert. Beschreibende Attribute sind ebenfalls Bezeichnung und Beschreibung.

4. Fazit und Ausblick

Beim Abgleich und Entwurf ereignisgesteuerter Prozeßketten ist es möglich durch die Klassifizierung existierender Ereignistypen entsprechende Konsistenz- und Integritätsprüfungen durchzuführen, wobei die Korrektheit der Syntax der Prozeßketten bzgl. der Ereignistypen überprüft wird. Dabei kann z.B. festgestellt werden, daß der gleiche Ereignistyp Funktionen innerhalb verschiedener Prozesse auslöst und damit eine Prozeßverknüpfungen existiert. So sollen redundante Ereignisse erkannt und wenn möglich eliminiert werden, damit die entsprechenden Prozesse syntaktisch richtig miteinander verknüpft werden können.

Darüberhinaus können durch das Erkennen systeminterdependenter Ereignistypen Verknüpfungen über die Systemgrenze hinaus aufgezeigt werden, was insbesondere bei einer Schnittstellenanalyse betrieblichen EDV-Systeme von Bedeutung ist.

Integritätsprüfungen ereignisgesteuerter Prozeßketten müssen aber auch dahingehend durchgeführt werden, daß untersucht werden muß, welcher Ereignistyp welchen Funktionstyp startet und welcher Funktionstyp welchen Ereignistyp als Ergebnis hat und welche Verknüpfungsoperatoren dabei auftreten können. Es stellt sich dabei z.B. die Frage ob eine automatisierte Funktion durch ein natürliches Ereignis (z.B. Telefonanruf) ausgelöst werden kann. Es zeigt sich bereits hier, daß Ereignisklassifikation und Funktionsklassifikation als Werkzeuge bzw. Hilfsmittel zur Optimierung von Prozeßketten hinsichtlich ihrer Syntax nur im Verbund wirksam sein können. Somit muß eine detaillierte Betrachtung der Funktionen hinsichtlich ihrer Struktur und Klassifikationsgesichtspunkte dieser Arbeit folgen.

F.typen E.typen	Funktionstyp1	Funktionstyp2	Funktionstyp3	Funktionstyp4
Ereignistyp1	X			
Ereignistyp2		X		
Ereignistyp3		X		
Ereignistyp4				X
Ereignistyp5			X	

Abb. 11: Ereignis-/Funktionsmatrix

Die Abhängigkeiten zwischen Ereignis- und Funktionstypen könnten dann z.B. ebenfalls in einer Matrix dargestellt werden (siehe Abbildung 11).

Mit dem entwickelten Klassifikationssystem für Ereignistypen und einer entsprechenden Funktionsklassifikation, sowie der Verknüpfung dieser Klassifikationen kann ein Teil der konzeptionellen Basis innerhalb der Entwicklung eines Prozeßanalyse-Tools geschaffen werden, mit dessen Hilfe ereignisgesteuerte Prozeßkette hinsichtlich ihrer Syntax optimiert werden können und die Umsetzung vom Fachkonzept in das DV-Konzept unterstützt wird.

Literaturverzeichnis

- Baresch, Keller, Kern, Kirsch: Methodenhandbuch zur Funktions-
/Prozeßmodellierung: Kooperationsprojekt IWI/IDS/SAP
1991
- Dahlberg, I.: Prinzipien der Klassifikation, Hrsg.: Gesellschaft für
Klassifikation e.V. , Frankfurt/Main 1977
- Engelien, G.: Der Begriff der Klassifikation
Hamburg 1971
- Fiedler, F.: Klassifikation der Wissenschaften, In: Europäische
Enzyklopädie zu Philosophie und Wissenschaften Bd. 3,
Hamburg 1990
- Herterich, Klein, Scheer: INMAS - eine individuell konfigurierbare
Schnittstelle, In: Information Management 1/1990
- Hoffmann, Nüttgens, Scholz, Scheer: Das Integrationskonzept am CIM-TTZ
Saarbrücken, In: Scheer, A.-W. (Hrsg.). Veröffentlichungen
des Instituts für Wirtschaftsinformatik an der Universität
Saarbrücken, Heft 85, Saarbrücken 1991
- Keller, Kirsch, Nüttgens, Scheer: Informationsmodellierung in der
Fertigungssteuerung, In: Scheer, A.-W. (Hrsg.).
Veröffentlichungen des Instituts für Wirtschaftsinformatik
an der Universität Saarbrücken, Heft 80,
Saarbrücken 1991
- Keller, Nüttgens, Scheer: Semantische Prozeßmodellierung auf der
Grundlage ereignisgesteuerter Prozeßketten, In: Scheer,
A.-W. (Hrsg.). Veröffentlichungen des Instituts für
Wirtschaftsinformatik an der Universität Saarbrücken,
Heft 89, Saarbrücken 1991
- Scheer, A.-W.: Architektur integrierter Informationssysteme - Grundlagen
der Unternehmensmodellierung, Berlin et al. 1991

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Institut für empirische Wirtschaftsforschung an der Universität des Saarlandes erscheinen in unregelmäßiger Folge.

* Die Hefte 1 - 31 werden nicht mehr verlegt.

- Heft 32: A.-W. Scheer: Einfluß neuer Informationstechnologien auf Methoden und Konzepte der Unternehmensplanung, März 1982, Vortrag anläßlich des Anwendergespräches "Unternehmensplanung und Steuerung in den 80er Jahren in Hamburg vom 24. - 25.11.1981
- Heft 33: A.-W. Scheer: Dispositio- und Bestellwesen als Baustein zu integrierten Warenwirtschaftssystemen, März 1982, Vortrag anläßlich des gdi-Seminars "Integrierte Warenwirtschafts-Systeme" in Zürich vom 10. - 12. Dezember 1981
- Heft 34: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS - Ein Ansatz zur Entwicklung prüfungsgerechter Software-Systeme, Mai 1982
- Heft 35: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Konzept einer computergestützten Prüfungsumgebung, Juli 1982
- Heft 36: A.-W. Scheer: Rationalisierungserfolge durch Einsatz der EDV - Ziel und Wirklichkeit, August 1982, Vortrag anläßlich der 3. Saarbrücker Arbeitstagung "Rationalisierung" in Saarbrücken vom 04. - 06. 10.1982
- Heft 37: A.-W. Scheer: DV-gestützte Planungs- und Informationssysteme im Produktionsbereich, September 1982
- Heft 38: A.-W. Scheer: Interaktive Methodenbanken: Benutzerfreundliche Datenanalyse in der Marktforschung, Mai 1983
- Heft 39: A.-W. Scheer: Personal Computing - EDV-Einsatz in Fachabteilungen, Juni 1983
- Heft 40: A.-W. Scheer: Strategische Entscheidungen bei der Gestaltung EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 41: H. Krcmar: Schnittstellenprobleme EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 42: A.-W. Scheer: Factory of the Future, Vorträge im Fachausschuß "Informatik in Produktion und Materialwirtschaft" der Gesellschaft für Informatik e. V., Dezember 1983
- Heft 43: A.-W. Scheer: Einführungsstrategie für ein betriebliches Personal-Computer-Konzept, März 1984
- Heft 44: A.-W. Scheer: Schnittstellen zwischen betriebswirtschaftlicher und technische Datenverarbeitung in der Fabrik der Zukunft, Juli 1984
- Heft 45: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Ein Werkzeug zur Messung der Qualität von Software-Systemen, August 1984
- Heft 46: H. Krcmar: Die Gestaltung von Computer am-Arbeitsplatz-Systemen - ablauforientierte Planung durch Simulation, August 1984
- Heft 47: A.-W. Scheer: Integration des Personal Computers in EDV-Systeme zur Kosten-

rechnung, August 1984

- Heft 48: A.-W. Scheer: Kriterien für die Aufgabenverteilung in Mikro-Mainframe Anwendungssystemen, April 1985
- Heft 49: A.-W. Scheer: Wirtschaftlichkeitsfaktoren EDV-orientierter betriebswirtschaftlicher Problemlösungen, Juni 1985
- Heft 50: A.-W. Scheer: Konstruktionsbegleitende Kalkulation in CIM-Systemen, August 1985
- Heft 51: A.-W. Scheer: Strategie zur Entwicklung eines CIM-Konzeptes - Organisatorische Entscheidungen bei der CIM-Implementierung, Mai 1986
- Heft 52: P. Loos, T. Ruffing: Verteilte Produktionsplanung und -steuerung unter Einsatz von Mikrocomputern, Juni 1986
- Heft 53: A.-W. Scheer: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung, Juli 1986
- Heft 54: U. Leismann, E. Sick: Konzeption eines Bildschirmtext-gestützten Warenwirtschaftssystems zur Kommunikation in verzweigten Handelsunternehmungen, August 1986
- Heft 55: D. Steinmann: Expertensysteme (ES) in der Produktionsplanung und -steuerung (PPS) unter CIM-Aspekten, November 1987, Vortrag anlässlich der Fachtagung "Expertensysteme in der Produktion" am 16. und 17.11.1987 in München
- Heft 56: A.-W. Scheer: Enterprise wide Data Model (EDM) as a Basis for Integrated Information Systems, Juli 1988
- Heft 57: A.-W. Scheer: Present Trends of the CIM Implementation (A qualitative Survey) Juli 1988
- Heft 58: A.-W. Scheer: CIM in den USA - Stand der Forschung, Entwicklung und Anwendung, November 1988
- Heft 59: R. Herterich, M. Zell: Interaktive Fertigungssteuerung teilautonomer Bereiche, November 1988
- Heft 60: A.-W. Scheer, W. Kraemer: Konzeption und Realisierung eines Expertenunterstützungssystems im Controlling, Januar 1989
- Heft 61: A.-W. Scheer, G. Keller, R. Bartels: Organisatorische Konsequenzen des Einsatzes von Computer Aided Design (CAD) im Rahmen von CIM, Januar 1989
- Heft 62: M. Zell, A.-W. Scheer: Simulation als Entscheidungsunterstützungsinstrument in CIM, September 1989
- Heft 63: A.-W. Scheer: Unternehmens-Datenbanken - Der Weg zu bereichsübergreifenden Datenstrukturen, September 1989
- Heft 64: C. Berkau, W. Kraemer, A.-W. Scheer: Strategische CIM-Konzeption durch Eigenentwicklung von CIM-Modulen und Einsatz von Standardsoftware, Dezember 1989
- Heft 65: A. Hars, A.-W. Scheer: Entwicklungsstand von Leitständen^[1], Dezember 1989
- Heft 66: W. Jost, G. Keller, A.-W. Scheer: CIMAN - Konzeption eines DV-Tools zur

Gestaltung einer CIM-orientierten Unternehmensarchitektur, März 1990

- Heft 67: A.-W. Scheer: Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1: Logisches Informationsmodell), März 1990
- Heft 68: W. Kraemer: Einsatzmöglichkeiten von Expertensystemen in betriebswirtschaftlichen Anwendungsgebieten, März 1990
- Heft 69: A.-W. Scheer, R. Bartels, G. Keller: Konzeption zur personalorientierten CIM-Einführung, April 1990
- Heft 70: St. Spang, K. Ibach: Zum Entwicklungsstand von Marketing-Informationssystemen in der Bundesrepublik Deutschland, September 1990
- Heft 71: D. Aue, M. Baresch, G. Keller: **URMEL**, Ein UnternehmensMODELlierungsansatz, Oktober 1990
- Heft 72: M. Zell: Datenmanagement simulationsgestützter Entscheidungsprozesse am Beispiel der Fertigungssteuerung, November 1990
- Heft 73: A.-W. Scheer, M. Bock, R. Bock: Expertensystem zur konstruktionsbegleitenden Kalkulation, November 1990
- Heft 74: R. Bartels, A.-W. Scheer: Ein Gruppenkonzept zur CIM-Einführung, Januar 1991
- Heft 75: M. Nüttgens, St. Eichacker, A.-W. Scheer: CIM-Qualifizierungskonzept für Klein- und Mittelunternehmen (KMU), Januar 1991
- Heft 76: Ch. Houy, J. Klein: Die Vernetzungsstrategie des Instituts für Wirtschaftsinformatik - Migration vom PC-Netzwerk zum Wide Area Network (noch nicht veröffentlicht)
- Heft 77: W. Kraemer: Ausgewählte Aspekte zum Stand der EDV-Unterstützung für das Kostenmanagement: Modellierung benutzerindividueller Auswertungssichten in einem wissensbasierten Controlling-Leitstand, Mai 1991
- Heft 78: H. Heß: Vergleich von Methoden zum objektorientierten Design von Softwaresystemen, August 1991
- Heft 79: A.-W. Scheer: Konsequenzen für die Betriebswirtschaftslehre aus der Entwicklung der Informations- und Kommunikationstechnologien, Mai 1991
- Heft 80: G. Keller, J. Kirsch, M. Nüttgens, A.-W. Scheer: Informationsmodellierung in der Fertigungssteuerung, August 1991
- Heft 81: A.-W. Scheer: Papierlose Beratung - Werkzeugunterstützung bei der DV-Beratung, August 1991
- Heft 82: C. Berkau: VOKAL (System zur Vorgangskettendarstellung und -analyse) - Struktur der Modellierungsmethode - Juni 1991 (wird nicht verlegt)
- Heft 83: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - Theoretical Foundations - 1991
- Heft 84: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - A Survey - 1991
- Heft 85: W. Hoffmann, M. Nüttgens, A.-W. Scheer, St. Scholz: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 1: Produktionsplanung), Oktober 1991

- Heft 86: A.-W. Scheer: Koordinierte Planungsinself: Ein neuer Lösungsansatz für die Produktionsplanung, November 1991
- Heft 87: M. Nüttgens, G. Keller, A.-W. Scheer, S. Stehle: Konzeption hyperbasierter Informationssysteme, Dezember 1991
- Heft 88: W. Hoffmann, B. Maldener, M. Nüttgens, A.-W. Scheer: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 2: Produktionssteuerung), Januar 1992
- Heft 89: G. Keller, M. Nüttgens, A.-W. Scheer: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßkosten (EPK)", Januar 1992
- Heft 90: C. Berkau, A.-W. Scheer: VOKAL (System zur Vorgangskettendarstellung und -Analyse), Teil 2: VKD-Modellierung mit VOKAL
- Heft 91: C. Berkau: Konzept eines controllingbasierten Prozeßmanagers als intelligentes Multi-Agent-System, Januar 1992
- Heft 92: A. Hars, R. Heib, Chr. Kruse, J. Michely, A.-W. Scheer: Approach to Classification for Information Engineering - Methodology and Tool Specification, August 1992
- Heft 93: M. Nüttgens, A.-W. Scheer, M. Schwab: Integrierte Entsorgungssicherung als Bestandteil des betrieblichen Informationsmanagements, August 1992
- Heft 94: Chr. Kruse, A.-W. Scheer: Modellierung und Analyse dynamischen Systemverhaltens, Oktober 1992
- Heft 95: R. Backes, W. Hoffmann, A.-W. Scheer: Konzeption eines Ereignisklassifikationssystems in Prozeßketten, November 1992