

**Heft 110**

**M. Remme, A.-W. Scheer**

**Konzeption eines  
leistungsketteninduzierten  
Informationssystemmanagements**

**September 1994**

# Inhaltsverzeichnis

<b>1 EINLEITENDE ZUSAMMENFASSUNG</b>	<b>1</b>
<b>2 SOFTWARE-ENGINEERING</b>	<b>2</b>
<b>2.1 Problematik herkömmlicher Vorgehensmodelle des Software-Engineering</b>	<b>2</b>
<b>2.2 Software-Engineering mit Informationssystem-Architekturen und detaillierten Vorgehensmodellen</b>	<b>7</b>
2.2.1 Informationssystem-Architekturen am Beispiel von ARIS	7
2.2.2 Vorgehensmodelle am Beispiel des V-Modells	9
<b>3 LEISTUNGSKETTEN ALS AUSGANGSPUNKT DER ENTWICKLUNG VON INFORMATIONSSYSTEMEN</b>	<b>11</b>
<b>3.1 Architektur eines Materialtransformationssystems</b>	<b>11</b>
<b>3.2 Erweiterung der Materialtransformation zu Leistungskettensystemen</b>	<b>16</b>
<b>3.3 Ableitung von Prozeßketten aus den Leistungsketten</b>	<b>18</b>
<b>4 ABSCHLIEßENDE BETRACHTUNG</b>	<b>19</b>
<b>5 LITERATURVERZEICHNIS</b>	<b>21</b>

# 1 Einleitende Zusammenfassung

*Im Rahmen des Software-Engineering besteht die Tendenz, daß bei der Modifikation von Informationssystemen die bestehenden Strukturen nicht grundsätzlich in Frage gestellt, sondern in das neue Informationssystem kopiert werden. Führt man dennoch grundsätzliche Änderungen durch, dann geschieht dies in der Regel unter Anwendung von Paradigmen, wie Lean Production, JIT und Prozeßorientierung. Diese beschreiben aktuelle Tendenzen, wie Organisationen effektiver zu gestalten sind, indem sie das zentrale Unternehmensziel der Gewinnmaximierung in verständlichere und einfacher umzusetzende Unterziele konkretisieren, die auf wesentliche Schwachpunkte eines Unternehmens abzielen.*

*Das paradigmensorientierte Vorgehen war zwar bisher sehr hilfreich, ist allerdings problematisch. Fragwürdig ist, ob der Erfüllungsgrad der übergeordneten betrieblichen Ziele immer bei Anwendung eher einfacher Regeln, die den Betrieb nur in Ausschnitten betrachten, verbessert wird. Soll beispielsweise wirklich jedes Unternehmen "leaner" werden, oder gibt es Gründe "fat" zu bleiben? Darüber hinaus greifen diese einfachen Regeln nur bei eklatanten Schwachpunkten von Unternehmen.*

*Mit der Lösung solcher Probleme muß sich auch die Konzeption des Software-Engineering als Teildisziplin des Organizations-Engineering beschäftigen. Sie fordert ein ingenieurmäßiges Vorgehen bei der Erstellung von Informationssystemen. Die bisher verfügbare methodische Unterstützung wird diesem Anspruch allerdings nicht gerecht. Die Vielzahl der existierenden Vorgehensmodelle bewegt sich in der Regel auf aggregiertem Niveau und berücksichtigt zum Teil kaum die Schnittstellen zwischen Phasen. Im Gegensatz dazu werden vergleichsweise detaillierte Modellierungsmethoden weitgehend unabhängig vom Vorgehensmodell und auch isoliert voneinander definiert.*

*Die hier vorliegende Arbeit untersucht die Bedeutung des Begriffs des Software-Engineering, indem auf Literatur der Ingenieurwissenschaften zurückgegriffen wird. Aus diesem Bereich werden einige Kerngedanken aufgegriffen, mit der momentanen Praxis des Software-Engineering verglichen und darauf aufbauend die Grundzüge einer Konzeption des leistungsketteninduzierten Informationssystemmanagements erläutert. Dieser Ansatz konzentriert sich auf die frühen Phasen der Software-Entwicklung, die bisher nicht gemäß ihrer Bedeutung für ein Informationssystem unterstützt werden. Mit seiner Hilfe werden bisher vernachlässigte Basisanforderungen erfaßt, die solche Sachverhalte umfassen, die zwar von wesentlicher Bedeutung für das zu entwickelnde Informationssystem sind, selbst allerdings keine zu beeinflussende Komponente darstellen. Wesentlicher Bestandteil der Leistungskette ist der Materialfluß. Insbesondere wird auch die Schnittstelle zwischen Anforderungsanalyse und den folgenden Entwurfsphasen so definiert, daß eine bessere Verwendung der Ergebnisse dieser Phase gewährleistet ist.*

## 2 Software-Engineering

### 2.1 Problematik herkömmlicher Vorgehensmodelle des Software-Engineering

Die Entwicklung von Vorgehensmodellen zur Entwicklung eines Informationssystems ist Aufgabe des Software-Engineering. Die Konzeption des Software-Engineering postuliert ein ingenieurmäßiges Vorgehen, bei dem Software auf der Basis von gesicherter Erkenntnis und unter Verwendung von Prinzipien, Methoden und Werkzeugen erstellt wird (vgl. Balzert, *Die Entwicklung von Software-Systemen 1982, S.22ff.*). Zur Konkretisierung des Begriffs des ingenieurmäßigen Vorgehens wird hier auf die Literatur zu den Ingenieurwissenschaften zurückgegriffen. Pahl/Beitz schreiben zu den Aufgaben eines Ingenieurs beispielsweise folgendes:

*"Wesentliche Aufgabe eines Ingenieurs ist es, für technische Probleme mit Hilfe naturwissenschaftlicher Erkenntnisse Lösungen zu finden und sie unter den jeweils gegebenen Einschränkungen stofflicher, technologischer und wirtschaftlicher Art in optimaler Weise zu verwirklichen." (Pahl/Beitz, Konstruktionslehre 1986, S.1).*

Neben der Betonung der Erkenntnis, die zur Lösung eines Problems führt, wird hier gleichrangig die zu berücksichtigenden Einschränkungen als wesentlicher Bestandteil der Ingenieurstätigkeiten genannt. Dies ist sinnvoll, da erst umfassende Kenntnisse um die Grenzen eines Systems dazu führen, daß ein vorhandener Handlungsspielraum auch ausgenutzt wird, wohingegen unscharfe Grenzen dazu führen können, daß man sich bei der Erstellung eines Systems innerhalb der als sicher definierten Grenzen bewegt, was zu einem Verlust von Freiheitsgraden führt.

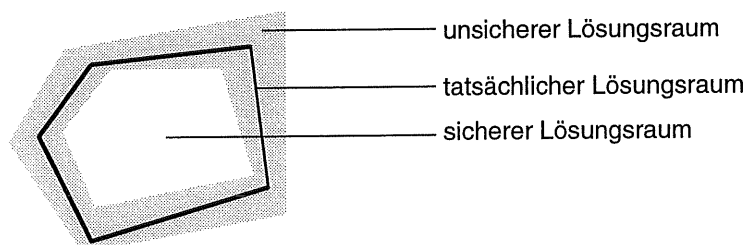


Abbildung 1: Verlust von Freiheitsgraden durch Unsicherheit

Abbildung 1 illustriert, wie Freiheitsgrade durch Unsicherheit verlorengehen. Der in der Regel unbekannte, tatsächliche Lösungsraum für ein System ist in der Abbildung der Bereich, der

durch die schwarze Linie eingeschlossen wird. Der graue Bereich beschreibt den Raum, in dem eine Lösung vom subjektiven Standpunkt eines Systemgestalters nur mit einer gewissen Wahrscheinlichkeit gültig ist. Wenn bei der Systementwicklung ein Fehler nur mit hohen Kosten wieder rückgängig zu machen ist, wie das bei der Informationssystementwicklung der

Fall ist, dann wird man sich bei der Konzeption nur in dem Lösungsraum bewegen, der als sicher eingeschätzt wird und verliert somit die Alternativen als Handlungsspielraum, die innerhalb des tatsächlichen Lösungsraum in der Abbildung grau gezeichnet sind. Um maximalen Handlungsspielraum nutzen zu können, muß also die Ingenieurwissenschaft, hier im Speziellen das Software-Engineering, eine Methodik zur Verfügung stellen, die den subjektiv sicheren Lösungsraum möglichst nah an den tatsächlichen Lösungsraum heranzuführt.

Ein solches "Grenz"-bewußtes Vorgehen darf allerdings nicht dazu führen, daß die Anforderungen eines benachbarten Systems an das betrachtete System keine Berücksichtigung finden. So schreibt Pahl:

*"Technische Erzeugnisse stehen nicht allein, sie sind Bestandteil eines übergeordneten Systems. In ihm wirkt vielfach der Mensch mit, indem er einwirkt. Dabei erfährt er Rückwirkungen, die ihn zum weiteren Handeln veranlassen. Der Mensch unterstützt so die gewollten Zweckwirkungen des technischen Systems."*  
(Pahl, Grundlagen der Konstruktionstechnik 1990, S. F3).

Pahl schafft damit ein Bewußtsein dafür, Systeme nicht isoliert, sondern immer im Kontext mit anderen Systemen, besonders mit solchen, in denen der Mensch eine Rolle spielt, zu betrachten.

Weiterhin schreiben Pahl/Beitz zur Konstruktionsmethodik:

*"Eine Konstruktionsmethodik soll: ...*

- *erfindungs- und erkenntnisfördernd sein, d. h. sie soll das Finden optimaler Lösungen erleichtern,*
- *Lösungen nicht zufallsbedingt erzeugen,*
- *Lösungen auf verwandte Aufgaben leicht übertragen lassen, ...*
- *den Erkenntnissen der Arbeitswissenschaft entsprechen, d. h. Arbeit erleichtern, Zeit sparen, Fehlentscheidungen vermeiden und tätige, interessierte Mitarbeit gewährleisten."* (Pahl/Beitz, Konstruktionslehre 1986, S.6).

Auch hier wird die Bedeutung des Begriffs der Erkenntnis betont. Zusätzlich werden die Nachvollziehbarkeit und die Übertragbarkeit in das Zentrum der Betrachtung gerückt. Wobei die Wichtigkeit dieses Punktes dadurch dokumentiert wird, daß er im Anschluß daran noch einmal in einer Textstelle aufgegriffen wird:

*"... muß die konstruktive Arbeit ebenfalls logischer und auch in den einzelnen Schritten verfolgbar, durchschaubar und korrigierbar werden ..."* (Pahl/Beitz, Konstruktionslehre 1986, S.6).

Analysiert man die Praxis der Informationssystem-Erstellung und die Vielzahl der Software-Lifecycle und Vorgehensmodelle (vgl. z. B. Stahlknecht, Einführung in die Wirtschaftsinformatik 1991, S. 229 oder Balzert, Die Entwicklung von Software-Systemen 1982, S. 17ff.), dann sieht die grundsätzliche Vorgehensweise zur Entwicklung eines Informationssystems folgen-

dermaßen aus: In einer ersten Phase werden Ausgangsdaten erhoben. Schwerpunkt dieser Phase ist die Ist-Analyse, in der insbesondere die Erfassung und Beschreibung der momentanen Abläufe erfolgt und daneben weitere Sachverhalte wie Unternehmensziele, Wünsche von betroffenen Mitarbeitern etc. erhebt. Anschließend wird die Ist-Situation im Hinblick auf die zu erreichenden Ziele bewertet und Schwachpunkte des Systems erarbeitet. Aufbauend auf den so erhobenen Daten wird eine DV-Konzeption erstellt und zum Schluß die Implementierung durchgeführt.

Abbildung 2 setzt diese Aussagen vereinfacht in eine Graphik um. Links ist der Lösungsweg des Konstruktionsingenieurs beschrieben. Ein Problem wird unter Berücksichtigung der relevanten Einschränkungen und unter Verwendung einer Methodik über die Erkenntnis in eine Lösung umgesetzt. Rechts findet sich der Lösungsweg des Softwareingenieurs, der zwar ähnlich aussieht, dennoch allerdings grundsätzlich verschieden ist. Initiale Wirkung hat in der Regel ein informationssystemorientiertes Problem, das unter Berücksichtigung der Ist-Situation des Informationssystems und unter Verwendung von Methoden der Informationssystementwicklung über die Erkenntnis in eine Lösung transformiert wird. Die Ausgangsdaten des Softwareingenieurs sind demnach auf informationssystemnahe Sachverhalte beschränkt.

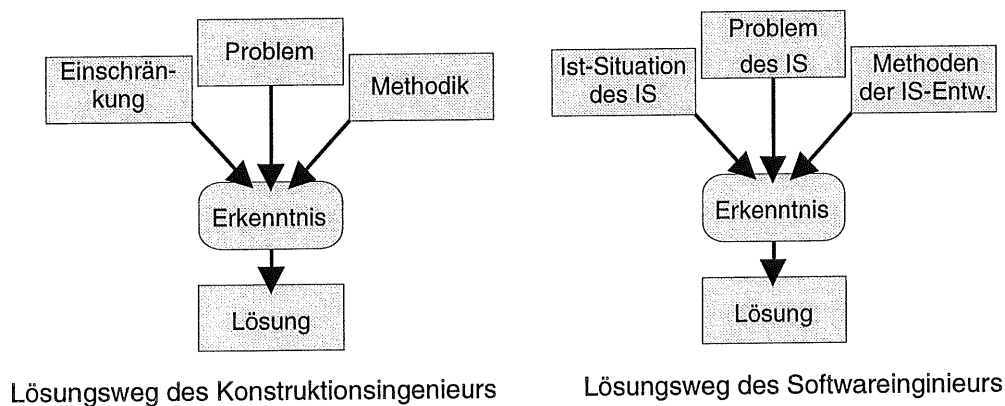


Abbildung 2: Unterschiedliche Philosophien der Lösungserstellung bei Konstruktions- und Softwareingenieuren

Bereits das Problem wird als informationstechnischer Natur angesehen. Statt eines Systems von Einschränkungen, wie es dem Konstruktionsingenieur vorgegeben ist, findet sich die Darstellung des momentan vorhandenen Informationssystems, das in der Art einer Restriktion angewendet wird. Bewußt wurden in der Abbildung den jeweils rechten Kästen einerseits Methodik<sup>1</sup> und andererseits Methoden der Informationssystementwicklung einbeschrieben. Während von den Ingenieurwissenschaften eine Methodik im Sinne einer planmäßigen,

<sup>1</sup>Zur Abgrenzung von Methode, Methodologie und Methodik: Eine Methode ist eine Vorschrift, wie planmäßig nach einer Kombination von Prinzipien zur Erreichung festgelegter Ziele vorzugehen ist. Die Methodologie ist eine Lehre über Methoden und abstrahiert damit vom Anwendungszusammenhang (vgl. auch Balzert, *Software-Systeme* 1982 und *Brockhaus-Enzyklopädie* 1991, S. 532ff.). Unter Methodik wird hier eine planmäßige, übergeordnete Verfahrensweise in einem Anwendungszusammenhang unter Verwendung einer ausgewählten Kombination aufeinander abgestimmter Methoden verstanden.

übergeordneten Vorgehensweise zur Verfügung gestellt wird und diese durch einzelne Methoden und Prinzipien konkretisiert wird (vgl. Pahl, *Grundlagen der Konstruktionstechnik 1990*, S. F5ff. und S. F11ff.), finden sich beim Software-Engineering eine Vielzahl nebeneinander existierender, unabhängiger Methoden und Vorgehensmodelle, die vorwiegend für die Sollkonzeption einsetzbar sind.

Insbesondere folgende Probleme ergeben sich aus diesem Vorgehen beim Software-Engineering:

- **Kein Orientierungsrahmen:** Dadurch, daß der Software-Ingenieur wenige und unformalisierte Anforderungen der Systeme erhält, die von dem zu erstellenden Informationssystem zu unterstützen sind, hat er wenige Anhaltspunkte, wie er das Informationssystem im Sinne des umgebenden Systems optimal gestalten muß. Aus diesem Grund tendiert man bei der Konzeption dazu, das Informationssystem nicht mehr als ein System anzusehen, das eine optimale Unterstützung anderer Systeme zu bieten hat, sondern die Informationssystemerstellung wird zum Selbstzweck. Optimierungen beziehen sich nur noch auf das Informationssystem selbst.
- **Lösungsprinzip wird zementiert:** Durch die starke Orientierung an den Ist-Systemen entspricht die momentan hinter der Informationssystemerstellung stehende Philosophie tendenziell eher einer Modifikation der bestehenden Systeme, als daß auch grundsätzliche Umgestaltungen mit in Betracht gezogen werden könnten. Die im Rahmen der Ist-Analyse ermittelten Schwachpunkte des Informationssystems, wie zum Beispiel Zeitverschleppungen, Medienbrüche etc. werden behoben. Bei diesem Vorgehen, das oft nur Ausschnittsbetrachtungen vornimmt, wird allerdings die übergeordnete fachliche Konzeption nicht in Frage gestellt, mehr oder minder wird nur die Art der Implementierung (im Extremfall von manuell zu DV-gestützt) geändert. Das Lösungsprinzip bleibt allerdings bestehen. So besteht also die vehemente Gefahr, daß zwar Probleme untergeordneter Bedeutung behoben werden, grundsätzlich falsche Konzeptionen aber in die neuen Informationssystemstrukturen hineinkopiert werden.

Besondere Probleme entstehen, seit in der letzten Zeit insbesondere im Rahmen des Paradigmenwechsels zur Prozeßorientierung hin immer wieder gefordert wird, Organisationen und Informationssysteme grundsätzlich in Frage zu stellen und zu ändern. So fordern Hammer/Champy beispielsweise, daß man das Reengineering der Strukturen nicht basierend auf den alten Strukturen durchführt, sondern daß man mit einem weißen Stück Papier beginnt (vgl. Hammer/Champy, *Business Reengineering 1994*, S.12-13). Einen praktikablen Weg, wie eine grundsätzliche Neuorientierung erreicht werden kann, wird allerdings nirgends beschrieben. Da der Software-Ingenieur bei Wegfall des realisierten Systems als Orientierungshilfe nur noch das Problem und eine mangelhafte Unterstützung mit Modellierungsmethoden vorfindet (vgl. dazu nochmals Abbildung 2), hat er praktisch überhaupt keinen Anhaltspunkt für seine Informationssystemerstellung. Die oben aufgeführten, von

Pahl/Beitz geforderten Eigenschaften einer Konstruktionsmethodik können in keinem Punkt erfüllt werden.

- **Lösungsraum diffus:** Insgesamt scheint aus der Sicht des Handelnden der scheinbare Lösungsraum im Ingenieurwesen exakter dem tatsächlichen zu entsprechen als das beim Software-Engineering der Fall ist. Es wurde bereits beschrieben, daß dies quasi einer Einschränkung des Handlungsspielraums gleichkommt.
- **Neue Paradigmen können nicht beurteilt werden:** Diese grundsätzlichen Änderungen von Organisationen und deren Informationssystemen wurde in den letzten Jahren nur unter Anwendung von Paradigmen, wie beispielsweise Lean Production, CIM, JIT und Total Quality Management durchgeführt. Diese beschreiben jeweils aktuelle Tendenzen, wie Organisationen effektiver zu gestalten sind. Sie konkretisieren das zentrale Unternehmensziel der Gewinnmaximierung in verständlichere und einfacher umzusetzende Unterziele, die auf wesentliche Schwachpunkte eines Unternehmens abzielen. Das paradigmensorientierte Vorgehen war zwar bisher sehr hilfreich, ist allerdings problematisch. So stellt sich die Frage, ob der Erfüllungsgrad der übergeordneten betrieblichen Ziele immer bei Anwendung derartiger, eher einfacher Regeln, die den Betrieb nur in Ausschnitten betrachten, verbessert wird. Soll wirklich jedes Unternehmen "leaner" werden, oder gibt es Gründe "fat" zu bleiben? Welches Paradigma ist als nächstes zu erwarten, oder sind die noch existierenden organisatorischen Schwachpunkte der Unternehmen nun nicht mehr derart eklatant, so daß einfache Handlungsanweisungen in ihrer Anwendbarkeit bald ausgereizt sind und nun wieder stärker ins Detail gegangen werden muß?

Bei den hier aufgeführten Problemen werden die Ergebnisse einer vom US Department of Defense durchgeführten Studie nicht übermäßig verwundern. Diese wurde zwar bereits 1985

<b>Ergebnis der Software-Entwicklung</b> (vgl. Thome, Wirtschaftliche Informationsverarbeitung 1990, K2 S.2)	<b>Prozentsatz</b>
Software wird nicht genutzt, da nicht den Anforderungen entsprechend	46
Software wird nicht ausgeliefert, weil das Problem unlösbar ist	30
Software wird erst nach umfangreicheren Modifikationen eingesetzt	19
Software kann nach kleineren Änderungen eingesetzt werden	3
Software kann direkt eingesetzt werden	2

Abbildung 3: Ergebnis der Software-Entwicklung

durchgeführt, belegt aber dennoch eindrucksvoll die zumindest bis zu diesem Zeitpunkt vorhandenen methodischen Schwächen des Software-Engineering, die in abgeschwächter



Form auch heute noch insbesondere in der Phase der Anforderungsdefinition unterstellt werden können.

## **2.2 Software-Engineering mit Informationssystem-Architekturen und detaillierten Vorgehensmodellen**

Neuere Architekturkonzeptionen von Informationssystemen sowie die in letzter Zeit entwickelten umfassenden und detaillierten Vorgehensmodelle sind wesentliche, bereits realisierte Schritte, die zu einem Software-Engineering führen, das der Anforderung des ingenieurmäßigen Vorgehens gerecht wird. Architektur und Vorgehensmodell stehen zueinander wie Stückliste und Arbeitsplan. Während in der Architektur der strukturelle Aufbau des Endprodukts (Informationssystem) beschrieben wird, wird im Vorgehensmodell die Abfolge von Aktivitäten spezifiziert, die zur Erstellung des Endprodukts führt. In diesem Sinne dürfen Architektur und Vorgehensmodell auch nicht als voneinander unabhängige Komponenten bei der Informationssystementwicklung betrachtet werden, sondern müssen aufeinander abgestimmt werden.

### **2.2.1 Informationssystem-Architekturen am Beispiel von ARIS**

Der Grundgedanke einer Architektur wird anhand der ARIS-Architektur ("Architektur integrierter Informationssysteme") von Scheer erläutert (vgl. *Scheer, Architektur integrierter Informationssysteme 1992, andere Architekturansätze finden sich z. B. bei Amice, CIM-OSA 1991, S. 41ff. oder Martin, Informations Engineering, Planning and Analysis 1990, S.2*). Die Notwendigkeit der Definition einer Architektur eines Informationssystems liegt darin begründet, daß die als unverzichtbar erkannte Integration von Systemen zwar zu einer Verbesserung der Abläufe beiträgt, daß damit aber auch die Komplexität ihrer Erstellung, Dokumentation und Verwaltung sprunghaft ansteigt. Um den bei der Erstellung von Software anfallenden Koordinationsaufwand aller beteiligten Stellen und den Abstimmungsaufwand der genutzten Methoden zu bewältigen, muß ein theoretisch fundiertes Rahmenwerk, das dafür eine Hilfestellung geben kann, verfügbar gemacht werden.

Als Grundlage zur Ableitung der Architektur wurde zunächst ein Modell entwickelt, das Unternehmensprozesse in umfassender Komplexität beschreibt. Aus dieser Beschreibung wurden dann solche Sachverhalte, die für die Informationsverarbeitung nicht relevant sind, herausgenommen. Dabei handelt es sich insbesondere um die Aspekte der Materialtransformation. In dem derart reduzierten System wurden Typen von Objekten identifiziert und diese dann nach inhaltlichen Aspekten zu sogenannten Sichten zusammengefaßt. Unter anderem entstand dabei die Sicht der Informationstechnologie. Basierend auf der Hypothese, daß die Berücksichtigung der Informationstechnologie am Anfang der Entwicklung eines Informationssy-

stems weniger relevant ist, wurde sie als eigenständige Sicht gestrichen und wird nun im Rahmen der verbleibenden Sichten über die Einführung eines Lifecycle-Modells berücksichtigt (vgl. Scheer, *Architektur integrierter Informationssysteme 1992*, S.4-19).

Das Ergebnis des Vorgehens zeigt Abbildung 4. Dieser Konzeption folgend werden Informationssysteme in den Sichten Funktionen, Organisation und Daten beschrieben. Die Berücksichtigung der Verbindungen zwischen den Sichten wird über ein weitere, als Steuerung bezeichnete Sicht gewährleistet.



Abbildung 4: ARIS - Architektur integrierter Informationssysteme

Wesentliche Aspekte von ARIS im Sinne dieser Arbeit sind:

- **Vollständigkeit:** Durch die theoretische Fundierung ist die Vollständigkeit der Beschreibung sichergestellt. So wird in im Vergleich zu ähnlichen Ansätzen in ARIS zusätzlich eine Organisationssicht berücksichtigt, durch die eine Verteilung von Funktionen und Daten auf Organisationseinheiten und Mitarbeiter möglich wird. Dies ist ein charakteristischer Aspekt, der z. B. bei der Erstellung von Client/Server-Konzeptionen berücksichtigt werden muß. Insbesondere wird so klar, daß auch der Mensch zur Informationsverarbeitung beiträgt.
- **Komplexitätsreduktion:** Die Vollständigkeit der Betrachtung führt zu einer hohen zu berücksichtigenden Komplexität. Dieser wird durch die Bildung der Sichten entgegengewirkt. Die erstellten Sichten werden nicht isoliert betrachtet; über exakte Schnittstellendefinitionen werden auch ihre Beziehungen zueinander in der Steuerungssicht abgebildet.
- **Konzentration auf das Wesentliche:** Der verwendete Lifecycle führt zu einer gleichrangigen Berücksichtigung der fachlichen Konzeption, die zunächst von der Informationstechnologie abstrahiert. Diese Phasen sichern ein Vorgehen, bei dem sich der Entwickler eines Informationssystems immer auf das gerade Wesentliche konzentrieren kann.

- **Konkretisierung:** Unter Verwendung einer benutzernahen und formalisierten Beschreibungssprache (ERM) werden die einzelnen Komponenten der Architektur in einem Metainformationsmodell inhaltlich sehr genau definiert (vgl. *Scheer, Architektur integrierter Informationssysteme 1992, S.19ff.*).

### 2.2.2 Vorgehensmodelle am Beispiel des V-Modells

Einen weiteren wesentlichen Schritt, der zu einem methodisch nachvollziehbaren Vorgehen beim Software-Engineering führt, repräsentiert das sogenannte V-Modell, das eine sehr umfassende und detaillierte Beschreibung eines Standards zur Softwareentwicklung bei der Bundeswehr vorgibt (vgl. *KBSt, Vorgehensmodell 1992, ein anderes derartiges Vorgehensmodell findet sich z. B. bei Österle/Brenner/Hilbers, Unternehmensführung und Informationssystem 1991*).

Im Regelungsteil des V-Modells wird die Software-Entwicklung aus prozessualer Sicht beschrieben. Die gesamte Entwicklung wird dabei in Aktivitäten und Teilaktivitäten untergliedert, die sodann beschrieben werden. Aktivitäten und Teilaktivitäten werden über Produkte (das sind neben Softwarekomponenten z. B. Handbücher und Anforderungsdokumente) miteinander verknüpft. Insgesamt entsteht auf diesem Weg ein komplexes Netz, das angibt, welche Produkte in welchem Zustand von einer Aktivität benötigt werden, welche Aufgaben in einer Aktivität durchgeführt werden müssen, und welche Produkte in welchem Zustand dadurch entstehen. Neben der Softwareerstellung werden auch begleitende Aktivitäten wie die Qualitätssicherung, Konfigurationsverwaltung und das Projektmanagement berücksichtigt, so daß dadurch insgesamt vier Submodelle spezifiziert werden.

Weitere Bestandteile des V-Modells neben dem Regelungsteil sind im Anhang zu finden. Es handelt sich dabei um vertiefende Erläuterungen zur Anwendung des V-Modells, Erläuterungen zu den Produkten und bereichsspezifische Regelungen.

Da der Anspruch des V-Modells auf Allgemeinheit sehr hoch ist, muß es in einem "Tailoring" genannten Prozeß im Vorfeld an die spezielle Anwendung angepaßt werden. Ziel des Tailorings ist es, genau die Aktivitäten auszuführen und die Produkte zu erstellen, die notwendig für die Durchführung des Projekts sind.

Überaus interessant ist die Definition von Rollen. Es handelt sich dabei um fiktive Funktionsträger wie z. B. System-Analytiker oder Programmierer, denen in Matrizen Aktivitäten zugeordnet werden. Gemäß dieser Matrix kann dann eine Zuordnung von Funktionen zu Personen durchgeführt werden, indem für die Personen die Rollen angegeben werden, die sie zu erfüllen haben.

Das V-Modell ist die erste Ebene einer Drei-Ebenen-Architektur, deren zweite Ebene sich mit den bei der Software-Entwicklung verwendeten Methoden befaßt und deren dritte Ebene

funktionale Anforderungen an Werkzeuge definiert, die zur Unterstützung des Software-Erstellungsprozesses herangezogen werden können.

Bestehende Ansätze von theoretisch fundierten Architekturen wie auch von detaillierteren Vorgehensmodellen repräsentieren einen wesentlichen Schritt zu einem Software-Engineering nach dem hier vertretenen Verständnis. Sie stellen allerdings eindeutig die Sollkonzeption eines zu entwickelnden Systems in den Vordergrund. Als ungelöstes Problem muß eine angemessene Durchführung der Anforderungsanalyse und Umsetzung der Ergebnisse in der Sollkonzeption angesehen werden.

### **3 Leistungsketten als Ausgangspunkt der Entwicklung von Informationssystemen**

Zur Definition des Handlungsspielraums, in dem man sich bei der Entwicklung von Informationssystemen bewegen darf, können grundsätzlich zwei Ansätze herangezogen werden. Einerseits kann man den Spielraum direkt definieren, indem man ein Ausgangsmodell in der Entwicklungssprache formuliert und zusätzlich einen Satz von Operationen angibt, durch den das Ausgangsmodell in das individuelle Modell des Informationssystems überführt wird. Denkbar ist z. B. die Beschreibung eines Maximalsystems, daß durch reine Streichungen auf das individuelle Modell reduziert wird. Konzeptionen, die Informationssysteme auf der Grundlage von Referenzmodellen entwickeln, bewegen sich in diese Richtung (vgl. *Scheer, Wirtschaftsinformatik 1994, Hars, Referenzdatenmodelle 1994 oder Remme/Allweyer/Scheer, Implementing Organizational Structures in Process Industry Supported by Tool-Based Reference Models 1994*). Einem anderen Ansatz folgend bleibt das Informationssystem selbst vorerst unberücksichtigt. So wird in dem hier verfolgten Ansatz zunächst ein mit dem Informationssystem in Zusammenhang stehender Sachverhalt erfaßt, dann werden Transformationsregeln für die Überführung dieses Sachverhalts in ein Informationssystem angegeben.

#### **3.1 Architektur eines Materialtransformationssystems**

Ein Unternehmen ist ein System, daß aus Gründen seiner hohen Komplexität nicht in voller Detaillierung geplant und geregelt werden kann. Es ist üblich, derartige Systeme in interdependent miteinander verbundene Subsysteme einzuteilen, die in Abhängigkeit von der Zielsetzung nach verschiedenen Kriterien gebildet werden können. Das Gesamtsystem soll derart zerlegt werden, daß die dabei entstehenden Subsysteme möglichst wenige Schnittstellen untereinander aufweisen. Diese lose gekoppelten Systeme weisen dann eine Makro- und eine Mikrostruktur auf. Die Komplexität ist dadurch reduziert, daß auf der Mikroebene nur ein Ausschnitt betrachtet wird, dafür dessen Objekte aber im Detail. Auf der Makroebene wird das Gesamtsystem betrachtet, dafür aber auf höherem Aggregationsgrad. Für das Management der Informationsverarbeitung erscheint es unmittelbar einsichtig, das Gesamtsystem in ein informations- und ein materialtransformierendes Subsystem zu unterteilen (vgl. Abbildung 5).

Im Sinne einer guten Schnittstelle zwischen dem material- und dem informationstransformierenden System ist es vorteilhaft, wenn diese beiden Systeme möglichst gleiche strukturelle Merkmale aufweisen. Daher wird, basierend auf dem umfassenden Ausgangssachverhalt zur Entwicklung der ARIS-Architektur und auf dem gleichen Weg, die Architektur eines Materialtransformationssystems hergeleitet. Das heißt, zunächst wird der Ausgangssachverhalt um für die Materialtransformation irrelevante Tatbestände reduziert, dann werden die einzelnen

Elemente der Materialtransformation zu umfassenderen Sichten zusammengefaßt und ein Phasenkonzept als Ersatz für eine eigenständige Sicht der Technologie eingeführt.

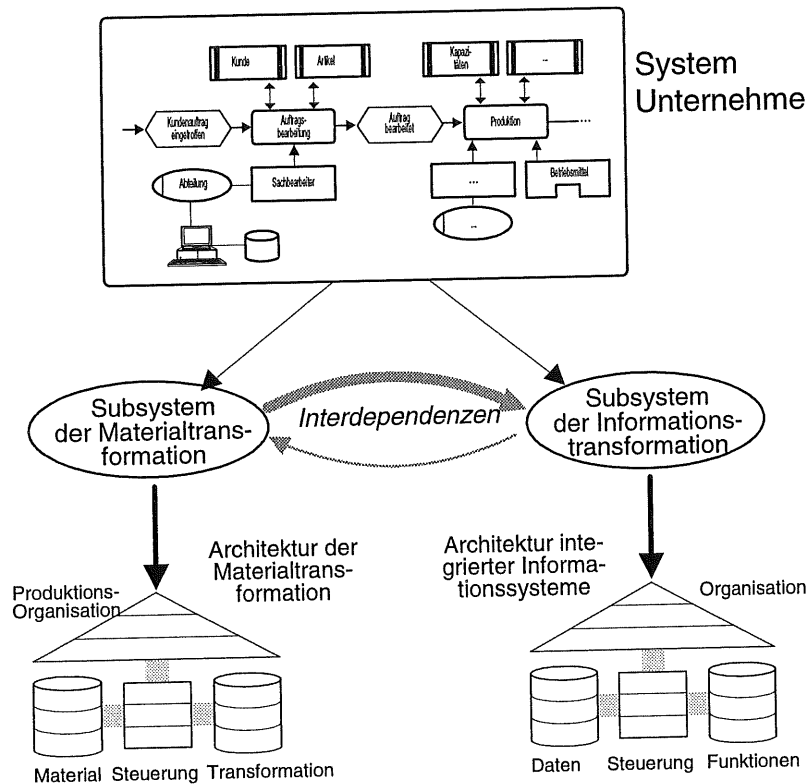


Abbildung 5: Subsysteme des Systems Unternehmen

Es ergeben sich folgende Sichten:

**Transformation:** Die Sicht der Transformation beschreibt insbesondere die Umwandlung von Input- in Outputmaterialien und die Abfolge der Transformationsfunktionen.

**Produktionsorganisation:** Die Produktionsorganisation definiert hauptsächlich Personal, das für die Produktion eingesetzt wird, Betriebsmittel, Werkzeuge, Zuordnungs- oder hierarchische Anordnungsbeziehungen zwischen diesen Objekten und weitere Verknüpfungen z. B. über Verrohrungen.

**Material:** In der Materialsicht werden vorwiegend Materialien und Beziehungen zwischen Materialien (z. B. Stücklisten) betrachtet.

**Steuerung:** In der Steuerungssicht werden die zuvor beschriebenen Sichten miteinander verknüpft.

Durch ein dreistufiges Phasenkonzept wird die Produktionstechnologie schrittweise in die Betrachtung mit einbezogen.

**Fachkonzept:** In einer fachkonzeptionellen Phase wird die Materialtransformation zunächst weitgehend unabhängig von speziellen Betriebsmitteln, Personen etc. entwickelt. Es werden grundsätzliche Entscheidungen zu Produktion und Produkten abgebildet.

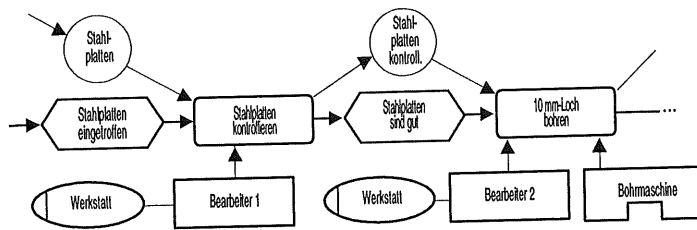


Abbildung 6: Materialflußdarstellung in Anlehnung an die EPK

det. Da derartige Sachverhalte Auswirkungen auf das ganze Unternehmen haben, somit insbesondere hier Abstimmungsmaßnahmen mit anderen Subsystemen durchgeführt werden müssen, muß an die verwendeten Beschreibungssprachen die Anforderung gestellt werden, daß sie im hohen Maß intuitiv verständlich sind, da auch solche Mitarbeiter damit umgehen müssen, die an der Materialflußgestaltung primär nicht beteiligt sind. In dieser Phase werden grobe Informationen zu Kapazitätsvorgaben, Durchlaufzeiten, Kosten, Personalanforderungen, Standorten etc. spezifiziert. Als weitere wesentliche Anforderung ist wiederum zu formulieren, daß eine gute Schnittstelle zu der Informationssystementwicklung bestehen soll. Dies ist gewährleistet, wenn die Beschreibungssprachen gemeinsame Objekte aufweisen. So wird hier ebenfalls die Methode EPK, erweitert um materialtransformationsrelevante Sachverhalte, verwendet, wie sie in Abbildung 6 dargestellt ist. Weitere Materialflußabbildungen auf vergleichbarem Abstraktionsniveau finden sich unter anderem bei *VDI, Produktionslogistik 1991, S. 34ff.* oder auch *Wiendahl, Analyse und Neuordnung der Fabrik 1990, S. 152 ff.*

**Design-Detaillierung:** In der Design-Detaillierung werden die Modelle des Fachkonzepts so weit verfeinert, daß die Ergebnisse als Ausgangspunkt für Anschaffung bzw. Konstruktion von Betriebsmitteln, Einstellung von Personal, Änderung des Fabriklayouts etc. verwendet werden können. Ergebnis sind demnach Typen von Betriebsmitteln und Personal, detaillierte Angaben zu Kapazitäten, Anzahl und Art von Transportmitteln, räumlicher Position etc. Auf dieser Ebene werden beispielsweise Beschreibungssprachen wie Petri-Netze verwendet, die sich für Simulationen des Materialflusses eignen (vgl. z. B. *Wiendahl, Analyse und Neuordnung der Fabrik 1990, S. 74 ff.*). Zu anderen Beschreibungssprachen siehe u. a. *VDI, Produktionslogistik 1991, S. 188 ff.*

**Implementierung:** In der diesem Phasenkonzept folgenden letzten Phase werden die Ergebnisse der Design-Detaillierung realisiert. Fallweise kann grade dieser Schritt

überaus aufwendig sein und muß in weitere untergeordnete aufgebrochen werden. Dies ist beispielsweise dann der Fall, wenn ein Materialflußsystem Betriebsmittel aufweist, die in Eigenkonstruktion erstellt werden müssen.

Die Architektur der Materialtransformation gibt hier zunächst nur ein Rahmenwerk vor, wie Daten über die Materialtransformation derart erfaßt werden können, daß sie sinnvoller Ausgangspunkt der Informationssystementwicklung sind. Das Materialtransformationssystem wird also aus der Sicht des Systems der Informationstransformation betrachtet. Der hier vorgestellten Architektur folgend kann allerdings auch eine Materialtransformationssystementwicklung durchgeführt werden. Den Sichten Transformation, Produktionsorganisation und Material können jeweils die für Produktionssysteme zentralen Begriffe Arbeitsplan, Fabriklayout und Stückliste zugeordnet werden. Deren vielfältige Verknüpfungen werden dann in der Steuerung formuliert. Ähnliche Lifecycle für die Erstellung von Produktionssystemen werden z.B. bei Tempelmeier/Kuhn verwendet (vgl. *Tempelmeier/Kuhn, Flexible Fertigungssysteme 1993, S.24*).

Das System der Materialtransformation gibt bereits auf fachlicher Ebene dem System der Informationsverarbeitung folgende Restriktionen vor:

**Daten:** In der Produktion ist der Ursprung einer erheblichen Menge von produktionsbeschreibenden Daten zu sehen, die von dem betrieblichen Informationssystem zu verarbeiten sind. Das Materialflußsystem bestimmt die zeitliche und örtliche Entstehung von Daten, ihren logischen Aufbau und die Bedingungen, unter denen sie entstehen.

**Organisation:** Im Materialflußsystem werden bereits eine Reihe von Stellen, Beziehungen zwischen Stellen, Zuordnungen von Stellen zu Standorten bzw. Betriebsmitteln etc. festgelegt. Da im Regelfall die Stellen nicht eindeutig dem Materialtransformationssystem- oder Informationstransformationssystem zugeordnet werden können, das heißt in der Produktion eingesetzte Mitarbeiter werden auch informationsverarbeitungsrelevante Aufgaben übernehmen, wird damit bereits eine Vorgabe für die Organisation des Informationssystems festgelegt.

**Funktionen:** Die Produktion determiniert örtlich und zeitlich den Ursprung einer Vielzahl von Ereignissen, die Funktionen der Informationsverarbeitung anstossen. Die Interdependenzen werden besonders eng, wenn Material- und Informationstransformation in einem System untrennbar miteinander verknüpft werden. Dies ist zum Beispiel bei flexiblen Fertigungssystemen der Fall.

Um eine solche Grundstruktur als Ausgangspunkt für das Informationssystemmanagement zu rechtfertigen, bedarf es einiger Prämissen, da den Interdependenzen zwischen den Subsystemen nicht in gleichem Maß Rechnung getragen wird. Die grundlegende Prämisse besagt, daß das Subsystem der Materialtransformation im Vergleich zum Subsystem der Informationstransformation relevanter und auch beständiger ist. Dies hätte ein Ungleichgewicht der einzelnen Interdependenzen zur Folge. Resultierend würde das System der Informationstransformation eher durch das der Materialtransformation determiniert als umgedreht.



Im folgenden soll begründet werden, das dieser Sachverhalt in vielen Unternehmen zutrifft.

**Das Subsystem der Materialtransformation ist beständiger:** Die bei der Informationsverarbeitung eingesetzte Technik ist aufgrund der vielen Innovationen (allein schon im Chip-Bereich, Datenspeichertechnik etc.) sehr schnell veraltet. Viele Komponenten sind bereits nach ca. zwei Jahren nicht mehr Stand der Technik. Im Gegensatz dazu ist die Technik, die für die Materialtransformation eingesetzt wird, dauerhafter einzusetzen (bis zu Jahrzehnten bei Universalmaschinen, Waggonen; finanzintensive Produktivmittel wie Kessel werden zum Teil wiederverwendet etc.). Auch ist die Logik der Materialtransformation, z. B. bei technikenabhängigen Produktionszusammenhängen wie dem Aufbau eines Produkts oder das innerhalb eines Produktionsrezepts verwendete Verfahren überaus stabil und können über Jahrzehnte hinweg gültig sein.

**Das Subsystem der Materialtransformation ist relevanter:** Aus der Sicht der Umwelt stellen das Produkt und bestimmte produktbegleitende Dienste die eigentliche Unternehmensleistung dar. Die Materialtransformation realisiert somit die Sachziele der Unternehmung (vgl. *Ferstl/Sinz, Grundlagen der Wirtschaftsinformatik 1994, S. 29*). Bei Betrachtung der Unternehmensfunktionen kann eine Reihe von Funktionen identifiziert werden, die von primärer Bedeutung für die Erstellung und Verwertung des Produkts sind. Die restlichen unternehmerischen Aufgaben, dazu gehört unter anderem das Informationsmanagement, dienen der Unterstützung dieser primären Funktionen (Vgl. dazu auch *die Aufgaben der Informationsprozesse bei Zäpfel, Taktisches Produktions-Management 1989, S. 99*).

Das Subsystem der Materialtransformation ist auch deswegen relevanter, weil sich die Informationstechnologie immer stärker einem Idealzustand nähert, in dem auf jede Information zu jedem Zeitpunkt überall in beliebiger Aufbereitungsform ökonomisch sinnvoll zugegriffen werden kann (vgl. *Scott Morton, The Corporation of the 1990s 1991, S. 13*). Das bedeutet, daß bei variierenden Konzeptionen von Produktionssystemen unterstellt werden kann, daß die Informationsversorgung grundsätzlich gewährleistet ist und daß die durch die Produktionskonzeption beeinflussten Kosten nicht erheblich voneinander abweichen.

Wenn diese Prämissen zutreffen, kann zunächst weitgehend unabhängig von den eingesetzten Informationssystemen ein System der Materialtransformation entwickelt und optimiert werden. Im Anschluß daran werden unterstützende Systeme wie das Informationssystem erstellt, die nur den Schnittstellenanforderungen des Materialtransformationssystems gehorchen müssen. Umfangreichere Rückkopplungen aus dem Informationssystem sind nicht zu erwarten.

### 3.2 Erweiterung der Materialtransformation zu Leistungskettensystemen

Neben der Beschreibung der Materialtransformation kann noch eine ganze Reihe weiterer Sachverhalte mit in einen Beschreibungsrahmen für ein Informationssystem aufgenommen werden. In Abbildung 7 sind zunächst die betrieblichen Subsysteme der Informations- und Materialtransformation dargestellt. Im Rahmen der Informationssystemerstellung kann nur das Subsystem der Informationstransformation beeinflusst werden. Es ist allerdings sinnvoll, dieses System nochmals in Subsysteme zu unterteilen, deren eines als variabel und deren anderes als fix einzustufen ist. Diese Unterteilung basiert auf der Vorstellung, daß ein Teil des Informationstransformationssystems tatsächlich nicht im Rahmen des Informationssystemmanagements beeinflusst werden kann oder soll, natürlich aber dennoch berücksichtigt werden muß. Dabei handelt es sich beispielsweise um grundsätzliche Aussagen über die ablaufenden Prozesse. Andererseits existiert ein Teilbereich im Subsystem der Materialtransformation, der zwar wie das gesamte Subsystem innerhalb der Informationstransformation nicht beeinflusst werden kann, jedoch relevant für das Subsystem der Informationstransformation ist. Die gesamte Organisation des Unternehmens wird also für die Entwicklung von Informationssystemen in drei Teile untergliedert. Es handelt sich dabei um einen Teil, der weitgehend irrelevant ist, einen Teil, der relevant aber nicht beeinflussbar ist und einen Teil, der im Rahmen der Entwicklung von Informationssystemen gestaltet werden muß.

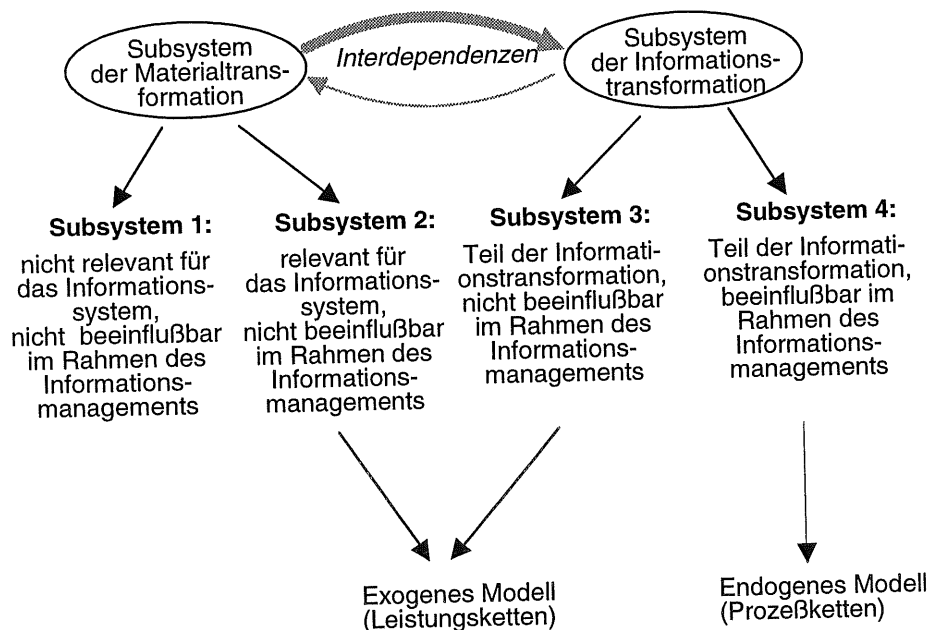


Abbildung 7: Komponenten des Leistungskettensystems

Fixer Teil des Informationssystems und relevanter, aber nicht beeinflussbarer Teil der Materialtransformation sollen Grundlage bzw. Rahmen der Gestaltung für Informationssysteme

sein. Sie repräsentieren das, was in ökonomischen Modellwelten üblicherweise als exogene Parameter bezeichnet werden, das heißt unabhängige Parameter, die außerhalb des betrachteten Modellzusammenhangs bestimmt werden. In Analogie zu diesen Parametern bei Berücksichtigung des komplexeren Sachverhalts, der durch diese Modelle beschrieben wird, soll es hier "exogenes Modell" genannt werden.

Schwerpunkt im exogenem Modell ist die Darstellung von Leistungsketten:

*Eine Leistungskette ist eine komplexe Abfolge von Aktivitäten, die unmittelbar der betrieblichen Leistungserstellung und -verwertung dienen. Zentraler Betrachtungsgegenstand von Leistungsketten sind Objekte, die sich entlang der Leistungskette durch ein Unternehmen bewegen und von den einzelnen Aktivitäten transformiert werden. In Industriebetrieben werden als Objekte insbesondere das Material und weitere wesentliche Objekte, insbesondere solche, die den Materialfluß begründen, entscheidende Wirkung auf ihn ausüben oder ihn beenden (z. B. Dokumente wie Aufträge), berücksichtigt.*

Insgesamt ist also die Leistungskette ein System von Objekten, die durch die Funktionen eines Unternehmens "fließen" und dabei transformiert werden. Abbildung 8 illustriert diesen Gedanken. Objekte wie Angebot, Material, Auftrag etc. bewegen sich durch das Unternehmen und führen in äußerst verflochtenen Abhängigkeiten zu gegenseitigen Transformationen.

Die Abbildung berücksichtigt, daß Transformationen auch betriebsextern durchgeführt werden. So wird der Beschaffungsauftrag betriebsextern in einen Lieferschein und Fremdbezugsteile transformiert. Dieser Betrachtung folgend ist es möglich, eine sehr geringe Anzahl von "originären Objekten" zu definieren, die betriebliche Aktivitäten starten, ohne selber mittelbar oder unmittelbar Ergebnis eigener betrieblicher Aktivitäten zu sein. Die originären Objekte rekrutieren sich insbesondere aus den eigenen Angeboten, Kundenaufträgen und gewissen, relativ unstrukturierten Informationen.

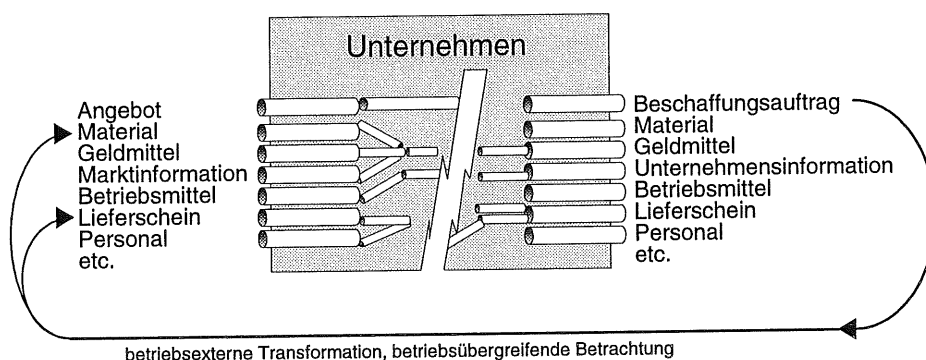


Abbildung 8: Fluß von Objekten durch ein Unternehmen

### 3.3 Ableitung von Prozeßketten aus den Leistungsketten

Abbildung 9 verdeutlicht die Umsetzung der Leistungskette in Prozeßketten, die der Beschreibung von Informationssystemen dienen. Bei der Leistungskette steht der Objektfluß im Vordergrund. Aus den Objekteigenschaftsänderungen werden grundlegende Ereignisse der Prozeßkette abgeleitet. Die Ableitungsbeziehungen, dargestellt durch gestrichelte Linien, beziehen sich insbesondere auf Materialien, können allerdings auch Betriebsmittel, Organisationseinheiten etc. betreffen (vgl. dazu auch Remme, *Prozeßorientierung in Krankenhäusern 1994*).

Die zur Darstellung der Leistungskette verwendete Beschreibungssprache ist nicht identisch mit der EPK, die zur Darstellung von Prozeßketten verwendet wird. Dieser Eindruck könnte schon deswegen entstehen, da sich die verwendeten Symbole und Begriffe zum Teil gleichen. Das liegt daran, daß man sprachlich nicht nach Material- und Informationstransformation differenziert. Abbildung 9 läßt Unterschiede zwischen den Beschreibungssprachen jedoch bereits erahnen. In Leistungsketten müssen oft Handlungsalternativen definiert werden, so gibt die Position der "Kontrolle" in der Leistungskette an, daß diese Funktion irgendwann zwischen Eingang und Abtransport des Materials ausgeführt werden kann. Solche Funktionen werden dann in der Prozeßkette "sequentialisiert".

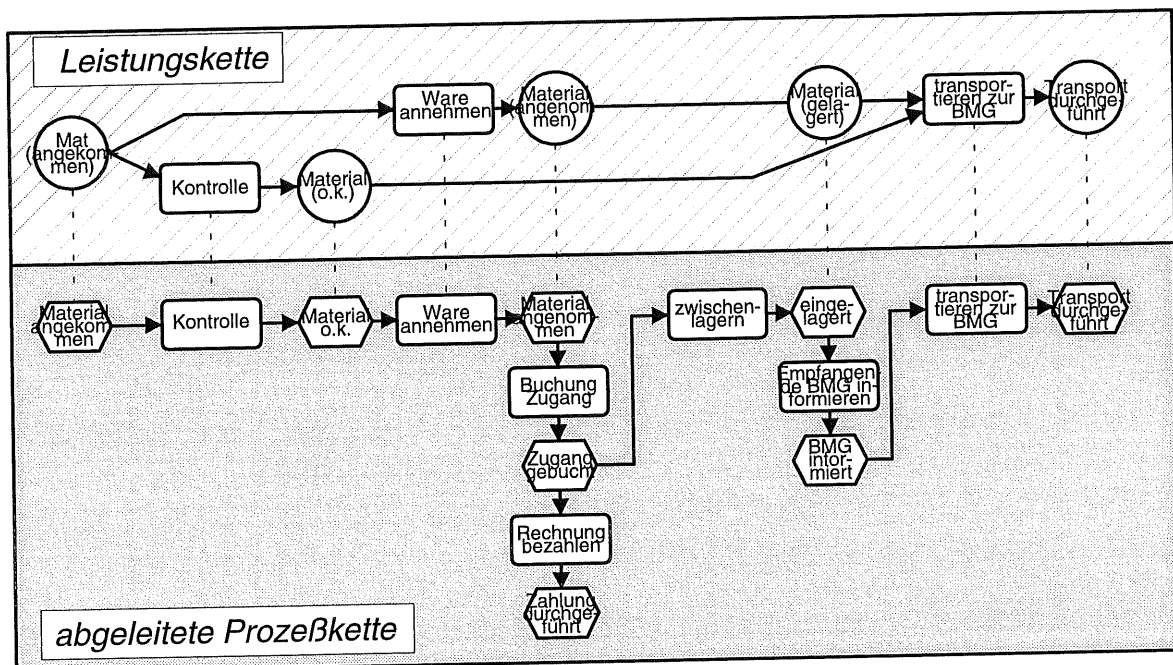


Abbildung 9: Leistungskette und abgeleitete Prozeßkette

## 4 Abschließende Betrachtung

Die Konzeption der Leistungsketten führt zu einer Verbesserung der Schnittstelle zwischen Informationssystemgestaltung und umgebenden Systemen und bewirkt somit, daß Software eher den an sie gestellten Anforderungen gerecht wird. Sie repräsentiert somit eine konsequente Fortführung der bisherigen Entwicklung des Software-Engineering.

Bis vor wenigen Jahren lag der Schwerpunkt der Informationssystementwicklung in der Implementierung. Die integrierenden Ansätze Anfang der 80er Jahre führten zu einer starken Zunahme der Komplexität von Informationssystemen. Daher fand eine Verlagerung des Schwerpunkts in die der Implementierung vorgelagerten Phasen statt, die die fachliche Konzeption des Informationssystems betrachten. Der hier vorgestellte Ansatz ist ein Plädoyer dafür, den Schwerpunkt nochmals in Richtung einer Ist-Analyse zu verlagern und hier insbesondere im Detail die Anforderungen solcher Systeme zu erfassen, die mit dem Informationssystem unterstützt werden sollen.

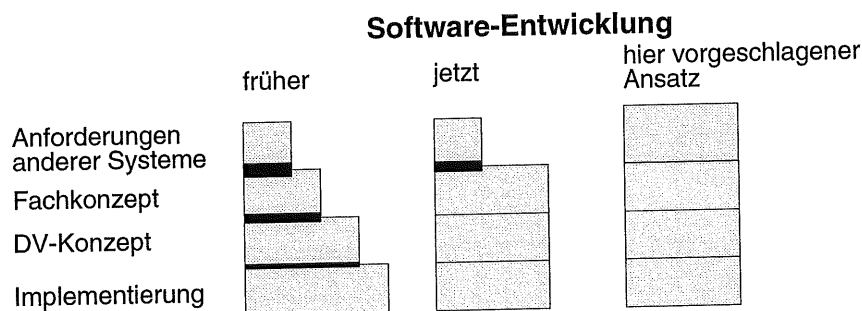


Abbildung 10: Entwicklung der Verteilung von Ressourcen auf die Phasen des Software-Lifecycles

Abbildung 10 stellt diese Entwicklungstendenz anhand eines 4-Phasen-Lifecycles dar. In der "Früher"-Situation liegt der Schwerpunkt in der Implementierung, ausgedrückt durch die besondere Länge des entsprechenden Rechtecks. Die Dicke der Balken zwischen den Phasen ist ein Maß für den Transformationsaufwand zwischen den Lifecycle-Phasen. Die Entwicklung zur aktuellen "Jetzt"-Situation ist durch die Verlagerung der Ressourcen zu einer Gleichberechtigung von der fachlichen Konzeption bis zur Implementierungsphase gekennzeichnet. Eine sinnvolle Methodenauswahl bewirkte eine Reduktion des Transformationsaufwands zwischen den Phasen. Der hier vorgeschlagene Ansatz führt dazu, daß nun nochmals eine Nivellierung bewirkt wird, so daß nun auch die Anforderungen anderer Systeme im Detail mit in die Betrachtung aufgenommen werden und die Transformation dieser Anforderungen in eine fachkonzeptionelle Sollkonzeption verbessert wird.

Folgende besonderen Vorteile verspricht die Berücksichtigung der Leistungsketten:

- **Transparente Entwicklungsmöglichkeiten:** Ein verbessertes ingenieurmäßiges Vorgehen wird unterstützt, indem der Handlungsspielraum bei der Systemkonzeption transparenter

wird. Die Leistungskette ist eine Restriktion. Wird diese von einem entwickelten System berücksichtigt, dann ist gewährleistet, daß das System die gewünschte Aufgabe grundsätzlich erfüllt.

- **Verminderung des Modifikationsaufwands:** Tritt eine für das Informationssystem relevante Wandlung des Unternehmensumfelds ein, so wird sich diese Änderung in der Regel in den Leistungsketten widerspiegeln. Bleibt nach Entwicklung eines Informationssystems die Ableitungsbeziehung zwischen Leistungskette und daraus induzierter Prozeßkette nachvollziehbar, dann besteht die Möglichkeit, bei Änderungen in der Leistungskette, die daraus resultierende Modifikation der Prozeßketten sehr klar abzugrenzen. Dies führt zu einer Verminderung des Anpassungsaufwands des Informationssystems.
- **Vernetzt-dezentralisiertes Informationsmanagement:** Leistungsketten können im Rahmen eines vernetzt-dezentralisierten Informationsmanagements als Vorgabe des unternehmenszentralen Informationsmanagements an dezentrale Entwicklungseinheiten verwendet werden. Teilprozeßketten könne dezentral ausgetauscht werden, wenn sie den Anforderungen genügen, die die Leistungsketten an sie richten. Andererseits besteht die Möglichkeit, Teilprozeßketten dann an unterschiedlichen Orten einzusetzen (Wiederverwendung von Teilprozessen), wenn die entsprechenden Leistungskettenabschnitte identisch sind (vgl. auch Kruse/Scheer, *Dezentrale Prozeßkoordination in Planungsinselfn 1994*).
- **Standardisierung von Abläufen ohne Flexibilitätseinbußen:** Es ist wahrscheinlich, daß Leistungskettenabschnitte nicht nur in einem einzigen Unternehmen, sondern mehrfach vorkommen. Anhand der Leistungsketten können das Leistungsangebot von Standardsoftwaremodulen und Anforderungen dieser Abschnitte (Daten, Funktionen, Regeln, auslösende Ereignisse) an diese Module formuliert werden.
- **Outsourcing:** Wie zuvor beschrieben, können Leistungsketten als Rahmenvorgabe an vernetzt-dezentralisierte Einheiten genutzt werden. Im gleichen Maß können sie auch als Rahmenvorgabe für unternehmensexterne Organisationseinheiten genutzt werden, zu denen betriebliche Funktionen ausgelagert werden sollen.

## 5 Literaturverzeichnis

- Amice, CIM-OSA 1991  
ESPRIT Consortium AMICE: CIMOSA: Open System Architecture for CIM, Berlin et al. 1991
- Balzert, H.: Die Entwicklung von Software-Systemen 1982  
Die Entwicklung von Software-Systemen - Prinzipien, Methoden, Sprachen, Werkzeuge, Mannheim 1982
- Bertram, H.; Blönnigen, P.; Bröhl, A.-P.: CASE in der Praxis 1993  
CASE in der Praxis: Softwareentwicklungsumgebungen für Informationssysteme, Berlin et al. 1993
- Brockhaus-Enzyklopädie 1991  
Brockhaus-Enzyklopädie, Bd. 14, 19. Aufl., Mannheim 1991
- Ferstl, O.K.; Sinz, E.J.: Grundlagen der Wirtschaftsinformatik 1994  
Grundlagen der Wirtschaftsinformatik, München-Wien-Oldenbourg 1994
- Hammer, M.; Champy, J.: Business Reengineering 1994  
Business Reengineering: die Radikalkur für das Unternehmen, 2. Aufl., Frankfurt-New York 1994
- Hars, A.: Referenzdatenmodelle 1994  
Referenzdatenmodelle - Grundlagen effizienter Datenmodellierung, Wiesbaden 1994
- KBSt, Vorgehensmodell 1992  
Koordinierungs- und Beratungsstelle der Bundesregierung für Informationsverarbeitung in der Bundesverwaltung: Planung und Durchführung von IT-Vorhaben - Das Vorgehensmodell, in: Bröhl, A.-P.; Dröschel, W.: Das V-Modell: der Standard für die Software-Entwicklung mit Praxisleitfaden, München-Wien-Oldenbourg 1993
- Kruse, C.; Scheer, A.-W.: Dezentrale Prozeßkoordination in Planungsinself 1994  
Dezentrale Prozeßkoordination in Planungsinself, in: Information Management, 9(1994)3, S. 6-11
- Martin, J.: Informations Engineering 1990  
Information Engineering, Book II: Planning and Analysis, Englewood Cliffs 1990
- Österle, H.; Brenner, W.; Hilbers, K.: Unternehmensführung und Informationssystem 1991  
Unternehmensführung und Informationssystem: Der Ansatz des St. Galler Informationssystem-Managements, Stuttgart 1991
- Pahl, G.; Beitz, W.: Konstruktionslehre 1986  
Konstruktionslehre, 2. Aufl., Berlin et al. 1986
- Pahl, G.: Grundlagen der Konstruktionstechnik 1990  
Grundlagen der Konstruktionstechnik, in: Beitz, W.; Küttner, K.-H.(Hrsg.): Dubbel - Taschenbuch für den Maschinenbau, 17. Aufl., Berlin et al. 1990
- Remme, M.: Prozeßorientierung in Krankenhäusern 1994  
Prozeßorientierung in Krankenhäusern, in: Management&Computer, 2(1994)3, S. 228-232

- Remme, M.; Allweyer, T.; Scheer, A.-W.: Implementing Organizational Structures in Process Industry Supported by Tool-Based Reference Models 1994  
 Implementing Organizational Structures in Process Industry Supported by Tool-Based Reference Models, in: Boucher, Elsayed, Jafari(Hrsg.): Computer Integrated Manufacturing in Process Industries, Proceedings of the CIMPRO #94 Conference, East Brunswick NJ, 25.-26. April 1994
- Scheer, A.-W.: Architektur integrierter Informationssysteme 1992  
 Architektur integrierter Informationssysteme, 2. Aufl., Berlin et al. 1992
- Scheer, A.-W.: Wirtschaftsinformatik 1994  
 Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 5. Aufl., Berlin et al. 1994
- Scott Morton, M.S.: The Corporation of the 1990s 1991  
 The Corporation of the 1990s: Information Technology and Organizational Transformations, Oxford et al. 1991
- Stahlknecht, P.: Einführung in die Wirtschaftsinformatik 1991  
 Einführung in die Wirtschaftsinformatik 5. Aufl., Berlin et al. 1991
- Tempelmeier, H.; Kuhn, H.: Flexible Fertigungssysteme 1993  
 Flexible Fertigungssysteme: Entscheidungsunterstützung für Konfiguration und Betrieb, Berlin et al. 1993
- Thome, R.: Wirtschaftliche Informationsverarbeitung 1990  
 Wirtschaftliche Informationsverarbeitung, München 1990
- VDI, Produktionslogistik 1991  
 VDI-Gemeinschaftsausschuß CIM; VDI-Gesellschaft Fördertechnik Materialfluß Logistik (Hrsg.): Rechnerintegrierte Konstruktion und Produktion, Bd. 5: Produktionslogistik, Düsseldorf 1991.
- Wiendahl, H.-P. (Hrsg.): Analyse und Neuordnung der Fabrik 1990  
 Analyse und Neuordnung der Fabrik, Berlin et al. 1990.
- Zäpfel, G.: Taktisches Produktions-Management 1989  
 Taktisches Produktions-Management, Berlin-New York 1989



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: Disposition- 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 Kostenrechnung, 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<sup>[1]</sup>, 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: URMELE, 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), Teil 1: Struktur der Modellierungsmethode - Dezember 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 Planungsinseln: Ein neuer Lösungsansatz für die Produktionsplanung, November 1991
- Heft 87: M. Nüttgens, G. Keller, 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 Prozeßketten (EPK)", Januar 1992 (noch nicht veröffentlicht)
- Heft 90: C. Berkau, A.-W. Scheer: VOKAL (System zur Vorgangskettendarstellung), Teil 2: VKD-Modellierung mit Vokal, Dezember 1991 (wird nicht verlegt)
- 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
- Heft 96: P. Loos: Die Semantik eines erweiterten Entity-Relationship-Modells und die Überführung in SQL-Datenbanken, November 1992
- Heft 97: Chr. Kruse, M. Gregor: Integrierte Simulationsmodellierung in der Fertigungssteuerung am Beispiel des CIM-TTZ Saarbrücken, Dezember 1992
- Heft 98: R. Heib: Konzeption für ein computergestütztes IS-Controlling, Dezember 1992
- Heft 99: H. Heß: Gestaltungsrichtlinien zur objektorientierten Modellierung, Dezember 1992
- Heft 100: P. Loos: Representation of Data Structures Using the Entity Relationship Model and the Transformation in Relational Databases, January 1993
- Heft 101: W. Hoffmann, J. Kirsch, A.-W. Scheer: Modellierung mit Ereignisgesteuerten Prozeßketten (Methodenbuch, Stand: Dezember 1992), Januar 1993
- Heft 102: P. Loos: Konzeption einer graphischen Rezeptverwaltung und deren Integration in eine CIP-Umgebung - Teil 1, Juni 1993
- Heft 103: wird noch nicht verlegt
- Heft 104: A. Traut; T. Geib; A.-W. Scheer: Sichtgeführter Montagevorgang - Planung, Realisierung, Prozeßmodell, Juni 1993
- Heft 105: A. Hars; V. Zimmermann; A.-W. Scheer: Entwicklungslinien für die computergestützte Modellierung von Aufbau- und Ablauforganisation, Dezember 1993
- Heft 106: W. Hoffmann; R. Wein; A.-W. Scheer: Konzeption eines Steuerungsmodells für Informationssysteme - Basis für die Real-Time-Erweiterung der EPK (rEPK), Dezember 1993
- Heft 107: R. Chen, A.-W. Scheer: Modellierung von Prozeßketten mittels Petri-Netz-Theorie, Februar 1994
- Heft 108: J. Galler, A.-W. Scheer: Workflow-Management: Die ARIS-Architektur als Basis eines multimedialen Workflow-Systems, Mai 1994
- Heft 109: Th. Allweyer, P. Loos, A.-W. Scheer: An Empirical Study on Scheduling in the Process Industries, July 1994
- Heft 110: M. Remme, A.-W. Scheer: Motivation und Konzeption eines leistungsketteninduzierten Informationssystemmanagements, September 1994