

Zweitveröffentlichung



Sinz, Elmar J.; Popp, Karl Michael

Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Datum der Zweitveröffentlichung: 20.09.2024

Akzeptiertes Manuskript (Postprint), Konferenzveröffentlichung

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-982016

Erstveröffentlichung

Sinz, Elmar J.; Popp, Karl Michael (1993): „Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt“. In: Karl Kurbel (Hrsg.), Wirtschaftsinformatik '93 : innovative Anwendungen, Technologie, Integration, Heidelberg: Physica-Verlag, S. 108–121, doi: 10.1007/978-3-642-52400-4_10.

Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt das deutsche Urheberrecht.

Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Elmar J. Sinz
Karl Michael Popp

Lehrstuhl für Wirtschaftsinformatik,
insbes. Systementwicklung und Datenbankanwendung,
Otto-Friedrich-Universität Bamberg
Feldkirchenstraße 21, D-8600 Bamberg

Zusammenfassung

Bei der klassischen Anforderungsanalyse wird ein betriebliches Anwendungssystem in Form von unterschiedlichen fachlichen Sichten definiert, die unmittelbar aus der betrieblichen Realität abgeleitet werden. Diese Vorgehensweise führt zu einer Reihe von Problemen bei der Abstimmung der Sichten. Der Modellierungsansatz des Semantischen Objektmodells (SOM) sieht dagegen eine zweistufige Anforderungsanalyse vor. In der ersten Stufe wird der relevante Ausschnitt der betrieblichen Realität, die Diskurswelt, in einem integrierten Modell beschrieben. Dieses Modell der Diskurswelt bildet den Ausgangspunkt für die anschließende fachliche Definition des Anwendungssystems. Im vorliegenden Beitrag wird ein Teilaspekt dieses Modellierungsansatzes beschrieben: Die rechnergestützte Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der Diskurswelt. Diese Form der Modellbildung vermeidet die genannten Abstimmungsprobleme, steigert die Qualität des konzeptuellen Schemas und ist geeignet, die bisherige Vorgehensweise der Datenmodellierung zu ersetzen.

1 Einführung

Die zunehmenden Anforderungen an die Qualität und die Flexibilität betrieblicher Anwendungssysteme bei gleichzeitig steigendem Kostendruck führen zu immer neuen Herausforderungen an die Methoden und Werkzeuge zur Anwendungsentwicklung. Besonders kritisch sind nach wie vor die frühen Phasen der Anwendungsentwicklung, deren Schwerpunkt die Anforderungsanalyse und -definition aus fachlicher Sicht ist. Hier, an

der Nahtstelle zwischen betrieblicher Realität und Anwendungssystem, ist insbesondere die Wirtschaftsinformatik aufgerufen, mit methodischen Beiträgen den genannten Anforderungen zu begegnen.

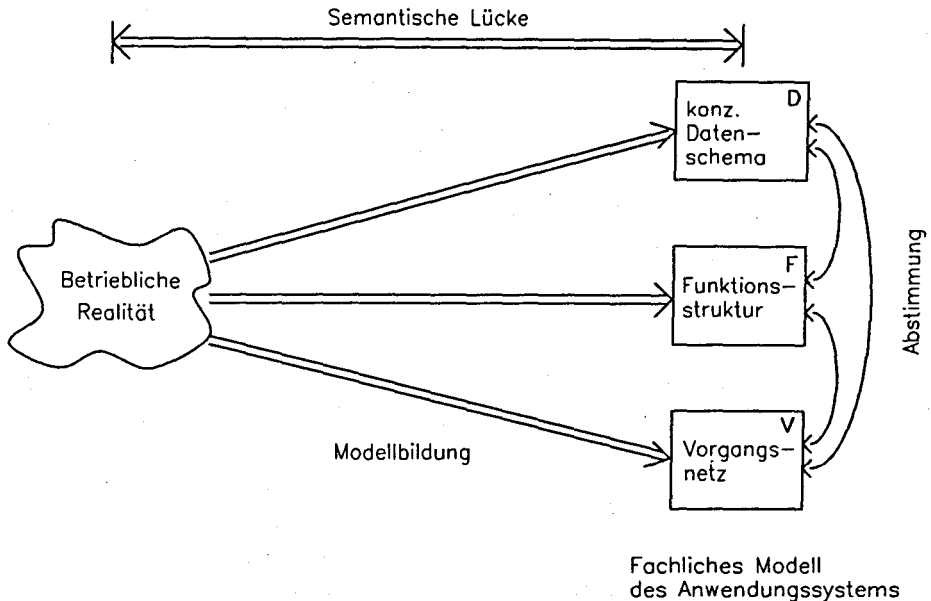


Bild 1: Klassische Anforderungsanalyse

Aufgabe der Anforderungsanalyse ist die Abbildung eines relevanten Ausschnitts der betrieblichen Realität, im folgenden als Diskurswelt bezeichnet, in die fachliche Anforderungsdefinition eines Anwendungssystems. Der methodische State-of-the-Art dieser Modellbildung ist in Bild 1 dargestellt. Danach wird die Anforderungsdefinition in mehreren getrennten fachlichen Sichten auf das Anwendungssystem beschrieben. Die wichtigsten Sichten sind

- die Datensicht in Form eines konzeptuellen Datenschemas (z.B. beschrieben im Entity-Relationship-Modell (ERM)),
- die Funktionssicht in Form einer Funktionsstruktur (z.B. beschrieben mit Hilfe der Strukturierten Analyse (SA)) und
- in Abhängigkeit vom verwendeten Modellierungsansatz eine Vorgangssicht oder Prozeßsicht, eine Steuerungssicht [Sche91] oder eine Kommunikationssicht.

Diese Form der Modellbildung der fachlichen Sichten eines Anwendungssystems besitzt u.a. folgende methodische Schwächen:

- Da jede Sicht unmittelbar aus der betrieblichen Realität erfaßt wird, besteht die Gefahr, daß die einzelnen Sichten mit unterschiedlichen Schwerpunkten, Detaillierungsgraden und "subjektiven Verzerrungen" modelliert werden.
- Die Abstimmung zwischen den getrennt modellierten Sichten ist problematisch.
- Es besteht eine große semantische Lücke zwischen der betrieblichen Realität und den einzelnen Sichten. Das zur Beschreibung der Sichten verwendete Begriffssystem ist näher an der Informatik als an der Betriebswirtschaftslehre orientiert.
- Die Abstimmung der Anforderungsdefinition mit der betrieblichen Organisation und mit dem vom Anwendungssystem zu steuernden betrieblichen Leistungssystem ist problematisch.

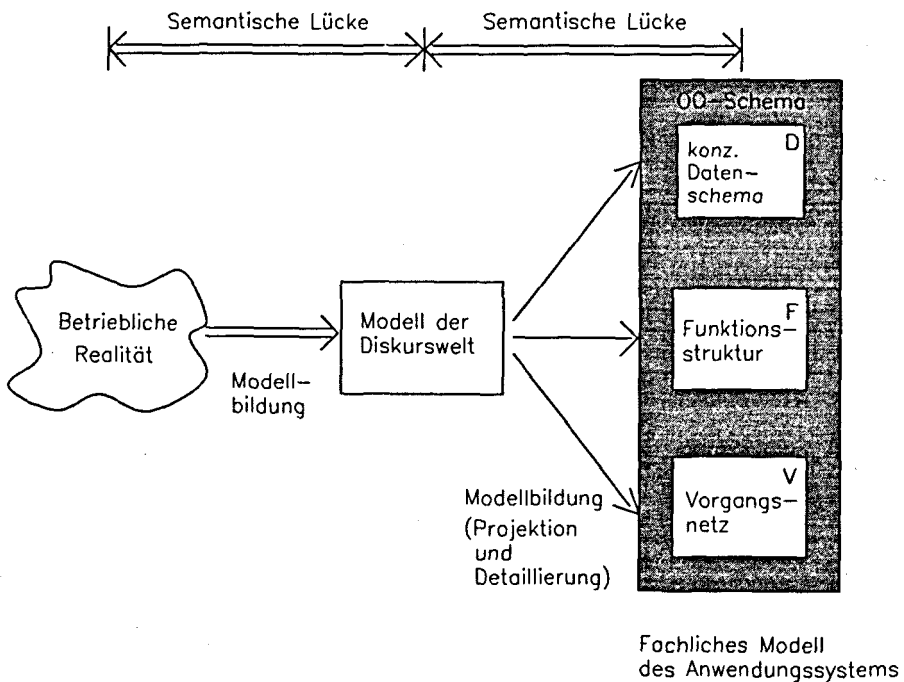


Bild 2: Anforderungsanalyse im SOM-Ansatz

Zur Vermeidung der beschriebenen Probleme unterstützt der Modellierungsansatz des Semantischen Objektmodells (SOM) [FeSi90, FeSi91] in der Anforderungsanalyse eine zweistufige Modellbildung. In der ersten Stufe wird ein explizites, geschlossenes Modell der Diskurswelt aufgestellt. Dieses dient in der zweiten Stufe bei konventioneller Anwendungsentwicklung als Basis für die Ableitung der herkömmlichen Sichten (Funktions-, Daten- und Vorgangssicht). Bei objektorientierter Anwendungsentwicklung wird daraus die fachliche Sicht des Anwendungssystems in geschlossener Form als Objektschema (oo-Schema) abgeleitet (Bild 2).

Auf diese Weise wird die in einem Modellierungsschritt zu überwindende semantische Lücke reduziert. Das Modell der Diskurswelt wird unter Verwendung eines betriebswirtschaftlichen Begriffssystems formuliert und stellt die Steuerung des betrieblichen Leistungssystems durch das betriebliche Informationssystem in den Mittelpunkt. Dabei wird sowohl der automatisierte Teil des Informationssystems, das Anwendungssystem, als auch der nicht automatisierte Teil abgedeckt. Das Diskursweltmodell ist dadurch mit der betrieblichen Realität leicht abstimmbare.

Der vorliegende Beitrag behandelt einen Aspekt des zweiten Modellierungsschrittes: Die Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Diskursweltmodell. Dieser Aspekt ist gleichermaßen für die konventionelle wie für die objektorientierte Anwendungsentwicklung relevant, da die statische Struktur des konzeptuellen Objektschemas im wesentlichen der des konzeptuellen Datenschemas entspricht.

2 Modellierung der betrieblichen Diskurswelt

Die Modellierung der Diskurswelt im SOM-Ansatz kann hier nur sehr knapp und im Hinblick auf die in diesem Beitrag behandelte Fragestellung vereinfacht dargestellt werden. Zu einer ausführlichen Darstellung siehe [FeSi91] und [FeSi92].

Betriebliche Objekte (Diskursweltobjekte) interagieren untereinander und mit Objekten der Umwelt (Umweltobjekte) durch Leistungsflüsse und Steuerflüsse. Leistungsflüsse (L-Flüsse) sind Flüsse von Gütern, Dienstleistungen und Zahlungen. Steuerflüsse (S-Flüsse) dienen zur Steuerung der Leistungsflüsse. Diskursweltobjekte, die mit L-Flüssen und S-Flüssen verknüpft sind, heißen SL-Objekte. Diskursweltobjekte, die nur mit S-Flüssen verknüpft sind, heißen S-Objekte. Flüsse werden durch Aufgaben realisiert, welche den

Diskursweltobjekten zugeordnet werden. Die Durchführung von Aufgaben wird durch Ereignisse gesteuert, die wiederum mit den Flüssen korrespondieren.

Ausgehend von einer globalen Abgrenzung wird die Diskurswelt im SOM-Ansatz sukzessive zerlegt. Wichtige Zerlegungskriterien sind (1) die Aufdeckung der Steuerung von SL-Objekten (Zerlegung eines SL-Objekts in S- und SL-Objekte), (2) die Homogenisierung der von einem Diskursweltobjekt behandelten Leistungsflußarten (Zerlegung eines SL-Objekts in SL-Objekte) sowie (3) die Differenzierung der Steuerung (Zerlegung eines S-Objekts in S-Objekte).

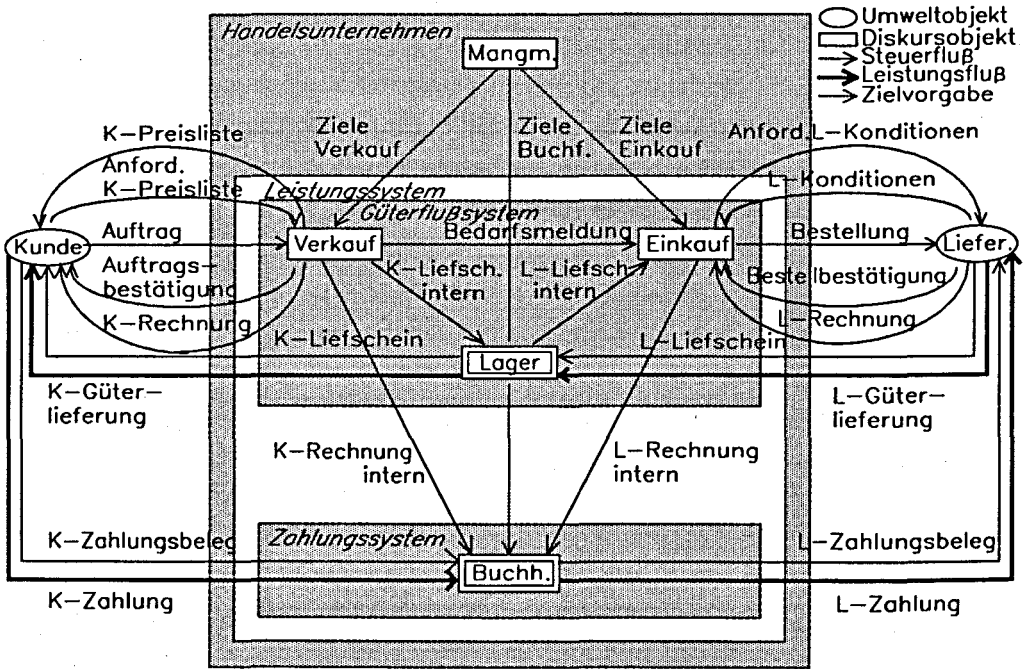


Bild 3: Interaktionsdiagramm des Handelsunternehmens

Jede Zerlegung wird in Form eines Interaktionsdiagramms beschrieben, das die Diskursweltobjekte und Umweltobjekte sowie die zugehörigen L-Flüsse und S-Flüsse darstellt. (Die ebenfalls modellierten Zielvorgaben werden im folgenden nicht berücksichtigt.) Bild 3 zeigt das Interaktionsdiagramm für das im folgenden verwendete Beispiel, ein stark vereinfachtes Handelsunternehmen:

- Objektzerlegung: Ausgangspunkt ist das SL-Objekt *Handelsunternehmen* und die beiden Umweltobjekte *Kunde* und *Lieferant*. *Handelsunternehmen* wird in das S-

Objekt *Management* und das SL-Objekt *Leistungssystem* zerlegt (Kriterium 1). Anschließend wird *Leistungssystem* in die beiden SL-Objekte *Güterflußsystem* und *Zahlungssystem* zerlegt (Kriterium 2). *Güterflußsystem* wird schließlich in die beiden S-Objekte *Verkauf* und *Einkauf* sowie das gemeinsame SL-Objekt *Lager* zerlegt (Kriterien 3 und 1). *Zahlungssystem* besteht ausschließlich aus dem SL-Objekt *Buchhaltung*.

- Leistungsflüsse: Güterflüsse verlaufen von *Lieferant* zu *Lager* und von dort zu *Kunde*. Zahlungsflüsse verlaufen in umgekehrter Richtung von *Kunde* zu *Buchhaltung* und von dort zu *Lieferant*.
- Steuerung der Leistungsflüsse: Jeder L-Fluß wird durch einen gegenläufigen S-Fluß ausgelöst und durch einen gleichläufigen S-Fluß begleitet. Zum Beispiel wird der L-Fluß *K-Güterlieferung* von *Lager* an *Kunde* durch den S-Fluß *Auftrag* von *Kunde* an *Verkauf* ausgelöst und durch *K-Lieferschein* von *Lager* an *Kunde* begleitet. Weitere S-Flüsse dienen der Vorbereitung und Nachbereitung von L-Flüssen sowie der innerbetrieblichen Steuerung.

Zu jedem Interaktionsdiagramm wird ein korrespondierendes Petri-Netz (siehe z.B. [Star90]) erstellt, dessen Übergänge die zu den Diskursweltobjekten gehörigen Aufgaben und dessen Zustände die Vor- und Nachereignisse der Aufgaben darstellen. Im Petri-Netz in Bild 4 sind nur Aufgaben dargestellt, die S-Flüsse realisieren. Jedes doppelumrandete Ereignis des Petri-Netzes korrespondiert mit einem S-Fluß des Interaktionsdiagramms.

Die Beschriftung der Petri-Netz-Kanten erfolgt als Tupel (Kantengewicht, Kantentyp). Reset-Kanten (Kantentyp R) [Hein80] entfernen beim Schalten alle Marken eines Vorzustandes, Copy-Kanten (Kantentyp C) entfernen keine Marke. Normal-Kanten (Kantentyp N) entfernen Marken in der Anzahl des Kantengewichts. Nicht beschriftete Kanten besitzen das Kantengewicht 1 und den Kantentyp N.

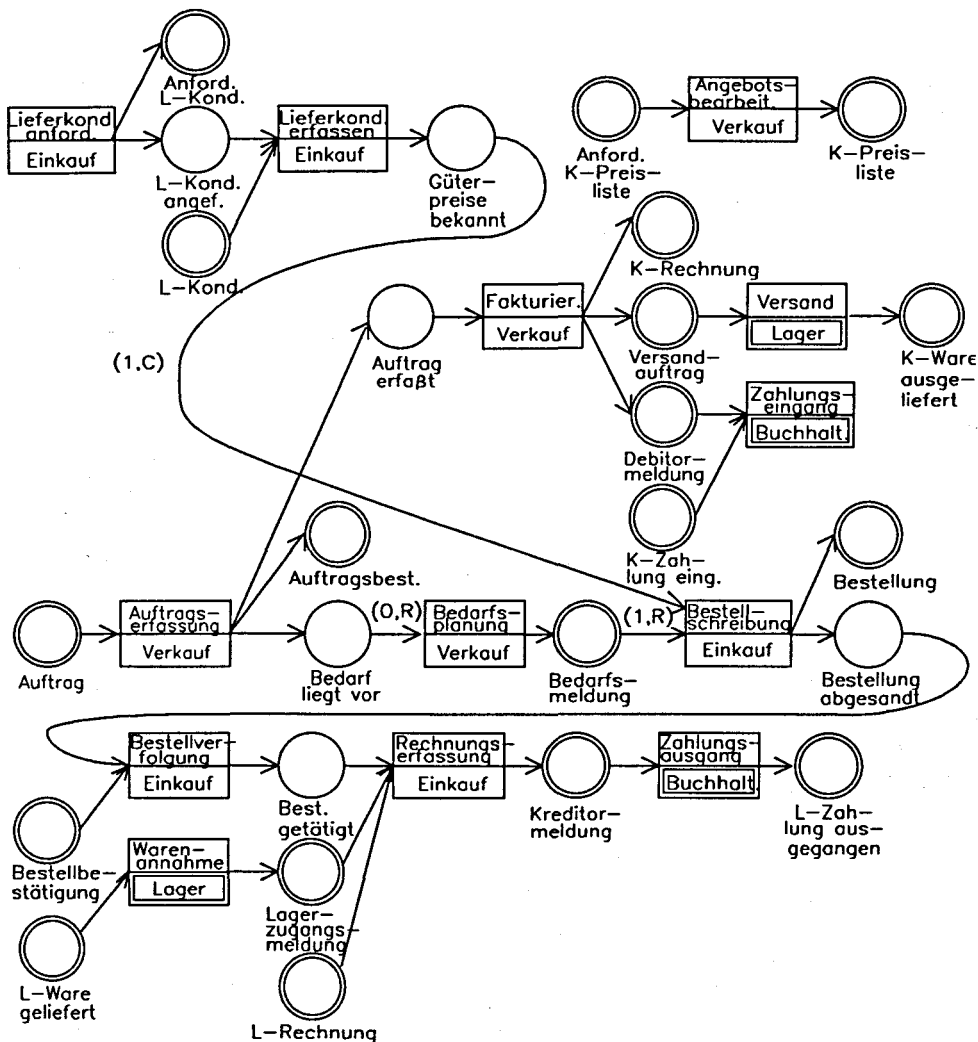


Bild 4: Petri-Netz des Handelsunternehmens

3 Modellierung des konzeptuellen Daten- bzw. Objektschemas

Konzeptuelle Datenschemata werden im folgenden im Strukturierten Entity-Relationship-Modell (SERM) [Sinz88, Sinz92] modelliert. In der graphischen Darstellung werden Datenobjekttypen als Knoten und Beziehungen zwischen Datenobjekttypen als Kanten dargestellt. Für die Komplexitätsgrade (0,1), (0,*), (1,1) und (1,*) der Beziehungen stehen spezielle Kantensymbole zur Verfügung (Bild 5).

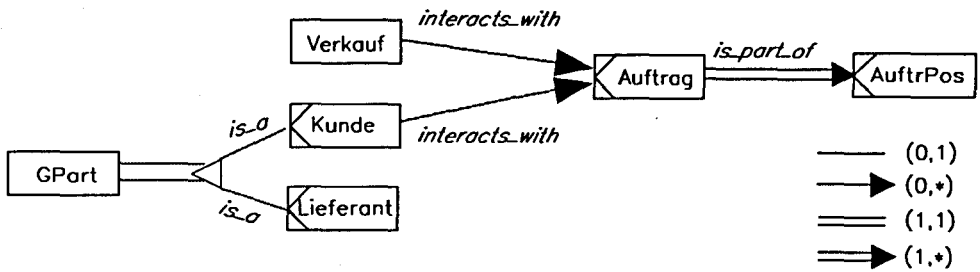


Bild 5: Ausschnitt eines konzeptuellen Daten- bzw. Objektschemas

Datenobjekttypen werden im SERM nach dem Grad ihrer Existenzabhängigkeit geordnet. In jeder Beziehung besteht eine Existenzabhängigkeit des rechts angeordneten Datenobjekttyps vom links angeordneten. In Bild 5 hängt z.B. *Auftrag* von *Kunde* ab. Ein Kunde steht mit null bis beliebig vielen Aufträgen in Beziehung (Komplexitätsgrad (0,*)), umgekehrt bezieht sich jeder Auftrag auf genau einen Kunden. Da jede Kante gerichtet von links nach rechts interpretiert wird, besitzt ein konzeptuelles Datenschema im SERM eine quasi-hierarchische Struktur (gerichteter, azyklischer Graph).

Ein konzeptuelles Objektschema gemäß SOM stellt aus der Sicht von SERM eine Erweiterung des konzeptuellen Datenschemas um folgende Punkte dar:

- Jeder Objekttyp wird durch seine (Daten-) Attribute und zusätzlich durch Methoden (Operatoren) und Nachrichtendefinitionen spezifiziert.
- Jede Beziehung zwischen Objekttypen wird einerseits als Referenz und zusätzlich als *interacts_with*-, *is_a*- oder *is_part_of*-Beziehung interpretiert [FeSi90]. *interacts_with*-Beziehungen dienen zur Modellierung von Interaktionskanälen für den Nachrichtentransport zwischen Objekten. Mit Hilfe von *is_a*-Beziehungen wird die Generalisierung von Objekttypen modelliert. Die *is_part_of*-Beziehung wird zur Modellierung komplexer, strukturierter Objekte verwendet. Bild 5 enthält Beispiele für die genannten Beziehungsarten.

In diesem Beitrag steht die Struktur konzeptueller Schemata in Form von (Daten-) Objekttypen und ihren Beziehungen im Vordergrund. Die nachstehenden Ausführungen beziehen sich daher gleichermaßen auf konzeptuelle Objektschemata und Datenschemata. Bei letzteren entfällt lediglich die Unterscheidung zwischen *interacts_with*-, *is_a*- und *is_part_of*-Beziehungen.

4 Abbildung zwischen Diskursweltmodell und konzeptuellem Schema

Die Abgrenzung zwischen dem Diskursweltmodell und dem konzeptuellen Schema ist in Bild 6 dargestellt. Das Anwendungssystem ist der automatisierte Teil des Informationssystems der Diskurswelt, das konzeptuelle Schema ist wiederum ein Teil der fachlichen Definition des Anwendungssystems (siehe auch Bild 2).

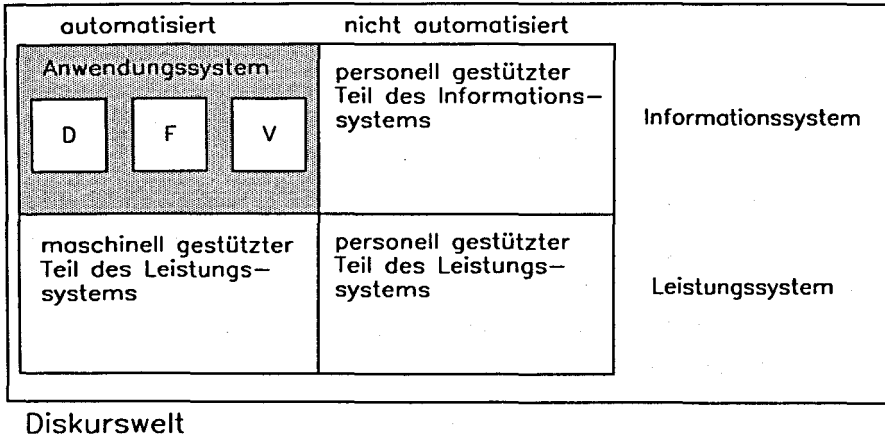


Bild 6: Struktur des Diskursweltmodells

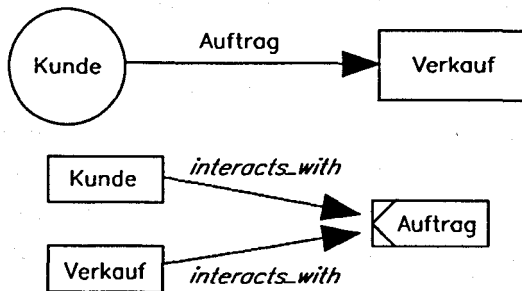


Bild 7: Beziehung zwischen Diskursweltmodell und konzeptuellem Schema

Das Modell der Diskurswelt wird im SOM-Ansatz durch ein Interaktionsdiagramm und ein Petri-Netz dargestellt. Zwischen dem Interaktionsdiagramm und dem konzeptuellen Schema besteht der in Bild 7 dargestellte Zusammenhang. Zum Beispiel enthält das Interaktionsdiagramm einen S-Fluß *Auftrag* zwischen *Kunde* und *Verkauf*. Dies wird im konzeptuellen Schema durch die Objekttypen *Kunde* und *Verkauf* modelliert, die mit dem

Objekttyp *Auftrag* über *interacts_with*-Beziehungen verbunden sind. Dabei besteht eine Existenzabhängigkeit des Objekttyps *Auftrag* von *Kunde* und *Verkauf*. Zusätzlich werden weitere Existenzabhängigkeiten zwischen Objekttypen aus dem Petri-Netz abgeleitet.

Die Abbildung zwischen dem Diskursweltmodell und dem konzeptuellen Schema wird in drei Schritten durchgeführt:

1. Im ersten Schritt wird die Transformation zwischen dem Diskursweltmodell und dem konzeptuellen Schema formal spezifiziert (siehe Abschnitt 4.1).
2. Im zweiten Schritt wird die Grobstruktur des konzeptuellen Schemas rechnergestützt abgeleitet (siehe Abschnitt 4.2).
3. Im dritten Schritt wird das konzeptuelle Schema weiter detailliert (siehe Abschnitt 4.3).

4.1 Spezifikation der Transformation

Die Spezifikation der Transformation besteht aus zwei Teilen:

- Festlegung der im Anwendungssystem zu speichernden S-Flüsse. Für diese S-Flüsse sind im konzeptuellen Schema Objekttypen mit entsprechenden Attributen anzulegen. Im Beispiel sollen die S-Flüsse *Anforderung L-Konditionen* und *Anforderung K-Preisliste* nicht im Anwendungssystem gespeichert werden.
- Festlegung derjenigen S-Flüsse, die im konzeptuellen Schema zusammengefaßt werden sollen. Dies ist z.B. dann der Fall, wenn ein betriebliches Dokument nacheinander auf unterschiedlichen S-Flüssen weitergereicht wird. Dieses Dokument wird im konzeptuellen Schema als ein Objekttyp modelliert:

<i>Auftrag</i>	faßt zusammen	<i>Auftrag</i> <i>Auftragsbestätigung</i>
<i>Bestellung</i>	faßt zusammen	<i>Bestellung</i> <i>Bestellbestätigung</i>
<i>K-Rechnung</i>	faßt zusammen	<i>K-Lieferschein intern</i> <i>K-Lieferschein</i> <i>K-Rechnung</i> <i>K-Rechnung intern</i>
<i>L-Rechnung</i>	faßt zusammen	<i>L-Rechnung</i> <i>L-Rechnung intern</i>

L-LiefSchein

faßt zusammen

*L-Lieferschein**L-Lieferschein intern*

Zum Beispiel werden die S-Flüsse *Bestellung* und *Bestellbestätigung* zusammengefaßt und in Form des Objekttyps *Bestellung* modelliert.

4.2 Ableitung der Grobstruktur des konzeptuellen Schemas

Ein Diskursweltobjekt bzw. ein Umweltobjekt wird in einen Objekttyp des konzeptuellen Schemas abgebildet, wenn es als Sender oder Empfänger eines in das Anwendungssystem zu übernehmenden S-Flusses auftritt. Jeder dieser S-Flüsse wird ebenfalls durch einen konzeptuellen Objekttyp realisiert und mit denjenigen Objekttypen durch *interacts_with*-Beziehungen verknüpft, die sein Sender- und Empfängerobjekt im konzeptuellen Schema repräsentieren. In Bild 8 wird z.B. der S-Fluß *K-Preisliste* zwischen *Kunde* und *Verkauf* (Bild 3) durch den Objekttyp *K-Preisliste* realisiert, der mit den Objekttypen *Kunde* und *Verkauf* durch *interacts_with*-Beziehungen verknüpft ist.

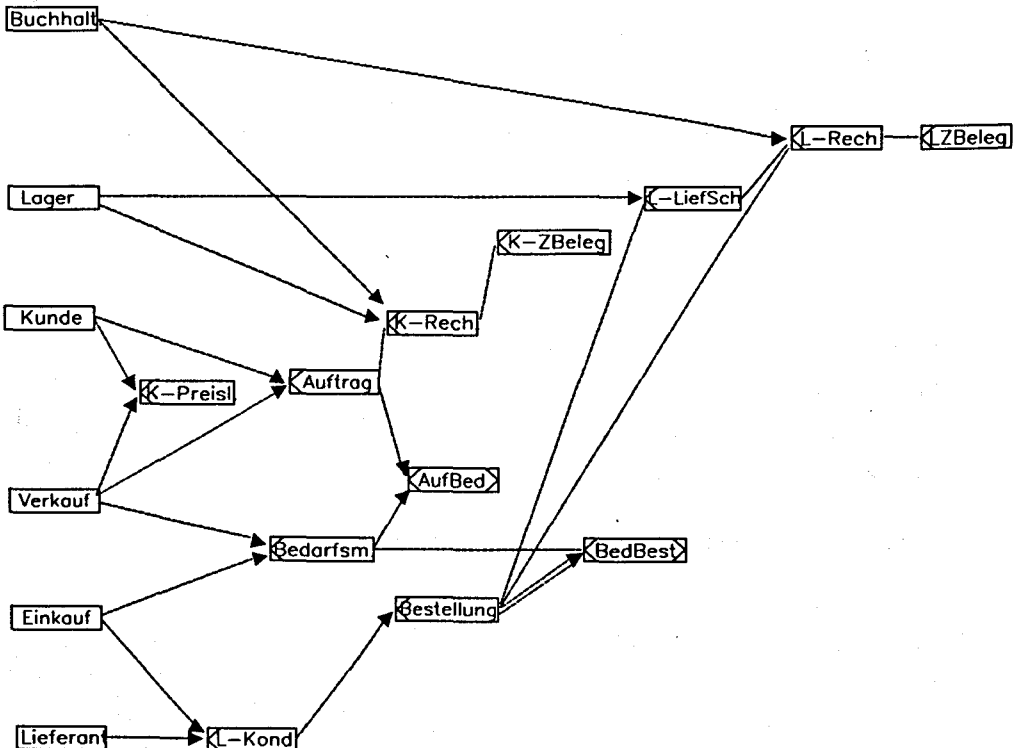


Bild 8: Grobstruktur des konzeptuellen Schemas

Zusätzlich werden die strukturbestimmenden Existenzabhängigkeiten zwischen Objekttypen aus dem Petri-Netz abgeleitet. Zum Beispiel resultiert in Bild 8 die Existenzabhängigkeit zwischen den Objekttypen *Auftrag* und *K-Rechnung* aus der Verknüpfung der Ereignisse *Auftrag* und *K-Rechnung* über die Aufgaben *Auftrags erfassung* und *Fakturierung* im Petri-Netz. Aus dem Petri-Netz können außerdem in begrenztem Umfang Komplexitätsgrade von Beziehungen abgeleitet werden. Hierauf kann im folgenden nicht weiter eingegangen werden. Bild 8 zeigt die vollständig rechnergestützt abgeleitete Grobstruktur des konzeptuellen Schemas.

4.3 Detaillierung des konzeptuellen Schemas

Die automatisch abgeleitete Grobstruktur des konzeptuellen Schemas enthält ausschließlich Objekttypen, die durch *interacts_with*-Beziehungen verknüpft sind. Diese Grobstruktur wird nun im letzten Schritt weiter detailliert. Folgende Aktivitäten sind erforderlich:

- Zuordnung von Attributen zu den einzelnen Objekttypen.
- Zerlegung komplexer Objekttypen unter Verwendung der *is_part_of*-Beziehung sowie Anpassung der davon betroffenen *interacts_with*-Beziehungen. Aus Datensicht entspricht dies einer Normalisierung von Objekttypen.
- Analyse fachlicher Generalisierungsstrukturen über den Objekttypen und ihre Modellierung mit Hilfe von *is_a*-Beziehungen. Anpassung der davon betroffenen *interacts_with*-Beziehungen.

Bild 9 zeigt das detaillierte konzeptuelle Schema zu Bild 8. Zum Beispiel wurde der komplexe Objekttyp *Bestellung* unter Verwendung einer *is_part_of*-Beziehung in die beiden normalisierten Objekttypen *Bestellung* und *BestPos* zerlegt. Ein Beispiel für eine fachliche Generalisierungsstruktur ist die Generalisierung von *Kunde* und *Lieferant* zu *Geschäftspartner (GPart)*. Auf die Darstellung der Attribute wurde aus Platzgründen verzichtet.

Die aus der Diskurswelt abgeleiteten Existenzabhängigkeiten zwischen Objekttypen stellen Invarianten des konzeptuellen Schemas dar. Diese dürfen bei der Detaillierung nicht verletzt werden.

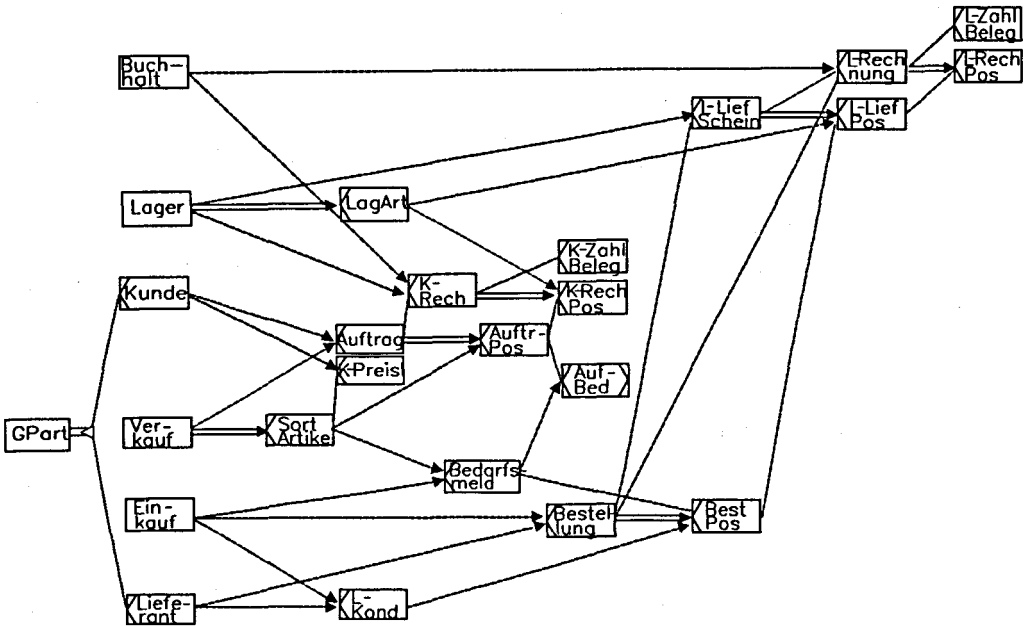


Bild 9: Detailliertes konzeptuelles Schema

5 Zusammenfassung und Bewertung

Die Anforderungsanalyse im SOM wird in zwei Stufen durchgeführt. In der ersten Stufe wird ein ganzheitliches Modell der Diskurswelt aufgestellt. Dieses dient in der zweiten Stufe als Basis für die Ableitung der fachlichen Definition des Anwendungssystems. Die obigen Ausführungen haben gezeigt, daß die Grobstruktur des konzeptuellen Schemas bereits weitgehend durch das Diskursweltmodell determiniert ist. Ein Vergleich des hier vorgeschlagenen Ansatzes mit der konventionellen Vorgehensweise läßt sich wie folgt zusammenfassen:

- Das Modell der Diskurswelt stellt eine stabile und leicht abstimmbare Plattform für die Anwendungsentwicklung dar. Durch den ganzheitlichen, an der Steuerung des Leistungssystems orientierten Modellierungsansatz lassen sich Inkonsistenzen bei der Systemerfassung leicht aufdecken. Die in einem Modellierungsschritt zu überwindende semantische Lücke wird reduziert. Die Grobstruktur des konzeptuellen Schemas ist aus dem Diskursweltmodell rechnergestützt ableitbar. Die Aufgabe der Datenmodellierung reduziert sich auf die Detaillierung des konzeptuellen Schemas.

- Im Gegensatz dazu sieht die konventionelle Vorgehensweise eine Erfassung des konzeptuellen Schemas unmittelbar aus der betrieblichen Realität vor. Diese Analyse ist mit einer Fülle von Unsicherheiten behaftet. Die Analyse erfolgt unter dem begrenzten Blickwinkel der Datensicht, Fehler bei der Analyse wirken sich aber auf das gesamte Anwendungssystem, auf die Beziehung zwischen Anwendungssystem und Organisation sowie auf die Steuerung des Leistungssystems durch das Anwendungssystem aus.

Wir sehen in der hier vorgeschlagenen Modellbildung einen Ansatz für eine spürbare Qualitätsverbesserung der fachlichen Anforderungsanalyse von Anwendungssystemen. Die bisherigen Untersuchungen mit umfangreicheren Schemata zeigen ermutigende Ergebnisse, die es rechtfertigen, weitere Forschungsbemühungen in diesen Weg zu investieren.

Literatur

- [FeSi90] Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). In: Wirtschaftsinformatik 6/90 (1990), 566 - 581
- [FeSi91] Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). In: Wirtschaftsinformatik 6/91 (1991), 477 - 491
- [FeSi92] Ferstl O.K., Sinz E.J.: Glossar zum SOM-Ansatz. Bamberger Beiträge zur Wirtschaftsinformatik Nr. 11, Bamberg 1992
- [Hein80] Heinemann B.: Teilklassen der selbst-modifizierenden Netze. Bericht Nr. 69, Fachbereich Informatik, Universität Hamburg, März 1980
- [Sche91] Scheer A.-W.: Architektur integrierter Informationssysteme. Springer, Berlin 1991
- [Sinz88] Sinz E.J.: Das Strukturierte Entity-Relationship-Modell (SER-Modell). In: Angewandte Informatik 5/88, 191 - 202
- [Sinz92] Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM). Erscheint in: Müller-Ettrich G. (Hrsg.): Fachliche Modellierung von Anwendungssystemen - Methoden, Vorgehen, Werkzeuge. Addison-Wesley, Bonn 1992
- [Star90] Starke P.H.: Analyse von Petri-Netz-Modellen. Teubner, Stuttgart 1990