

Kurs 1585 „Software Praktikum“ WS 00/01

Lehrgebiet Praktische Informatik III
Prof. Dr. Hans-Werner Six, Dr. Mario Winter

The screenshot displays a software development environment with three main windows:

- Top Window (Drawing: KontoManager):** Shows a database schema diagram. Entities include *Person* (attributes: Nachname, Anschrift, Geburtstag, Geschlecht), *Konto* (attributes: Kontonummer, Sperrmerk, Saldo), *Buchungsbeleg* (attributes: Betrag, Buchungstg, Auftragsnr), *Dauerauftrag* (attributes: Betr, Dau, Per), *Kontoinhaber* (attributes: Perf, Beant, Einkommen), *Unternehmen* (attributes: Name, Anschrift, Branche, Umsatz), *privates Konto*, *laufendes Konto*, *Spar-konto*, *Girokonto*, and *Firmen-konto*. Relationships include *erzellt* (1 Person to n Dauerauftrag), *ist zeichn-berechtigt* (n Person to m Konto), *verwaltet* (m Konto to m Buchungsbeleg), *besitzt* (n Kontoinhaber to n privates Konto), and *gehört* (1 Unternehmen to n Firmen-konto).
- Middle Window (Entwurf):** Shows a flowchart with four main nodes: *Buchungs-belege*, *Konten*, *Dauer-aufträge*, and *Privat-kunden*. Each node is associated with a 'TS' (Transaction Set) and has arrows indicating data flow between them.
- Bottom Window (Implementation):** Contains a code editor with the following text:


```

      Line: 73
      Lines: 56-74

      1 THEN
      ger^.LinkerSohn
      ger^.RechterSohn;

      utZeiger = NIL*)
      den Knoten *)
      
```

 Below the code is a control panel with buttons: *print*, *print **, *next*, *step*, *stop at*, *cont*, *stop in*, *clear*, *where*, *up*, *down*, *run*.

Labels at the bottom of the screenshot indicate the phases: **Problemanalyse und Produktdefinition** (top left), **Entwurf** (middle left), and **Implementation** (bottom center).

Softwarepraktikum WS 2000/2001

Aufgabenstellung

FernUniversität Hagen
Lehrgebiet Praktische Informatik III
Informatikzentrum
Feithstr. 140
58084 Hagen

Inhaltsverzeichnis

Inhalt	2
Vorwort	3
1 Allgemeines und Formalia	5
1.1 Genereller Ablauf	5
1.2 Von Ihnen abzuliefernde Dokumente	6
1.3 Bearbeitungshinweise	7
1.4 Termine	8
1.5 Betreuung und Sprechzeiten	8
1.6 Kriterien für das Bestehen der erste Heimphase	9
2 Produktskizze des Hilfskraft-Verwaltungssystems (HiKS)	11
3 Feststellung der operativen Anforderungen mit MSA	17
3.1 ER-Modellierung	17
3.2 Modellierung der Systemzustände	22
3.3 Datenflußmodellierung	23
3.4 Prozeß-Spezifikationen	33
3.5 Das Datenlexikon	50

Vorwort

Liebe Fernstudentin, Lieber Fernstudent,

das Softwarepraktikum ist sicherlich die zeitintensivste und privat am schwierigsten zu organisierende Veranstaltung Ihres Studiums und stellt für Sie neben den Prüfungen und der Diplomarbeit die größte Hürde auf dem Weg zum Diplom dar. Bevor Sie sich mit der Aufgabenstellung befassen, möchten wir Ihnen aus unserer Sicht noch einige Worte über die Stellung und Bedeutung des Softwarepraktikums mit auf den Weg geben, die Ihnen hoffentlich helfen, auch eventuell bevorstehende frustrierende und nervige Momente etwas motivierter zu überstehen.

Ziel des Softwarepraktikums ist es, die im Kurs 1793 „Software Engineering I“ („SE I“) vermittelten Methoden und Techniken in einem größeren Softwareprojekt konkret anzuwenden und ihre Praktikabilität zu demonstrieren. Zusätzlich geht es darum, im Team und unter Zeitdruck ein definiertes Projektziel zu erreichen, nämlich ein funktionsfähiges Softwaresystem zu realisieren, das fest vorgegebene Anforderungen erfüllt. Da die meisten von Ihnen nach dem erfolgreichen Studienabschluß in der einen oder anderen Form in der Softwarebranche tätig sein werden, wo genau diese Fertigkeiten gefragt sind, besitzt das Softwarepraktikum wahrscheinlich die größte Praxisnähe aller Veranstaltungen des Informatikstudiums. Von vielen ehemaligen Fernstudenten haben wir jedenfalls ein derartiges Feedback erhalten. Sie profitieren also langfristig davon, wenn Sie das Praktikum mit der entsprechenden Einstellung angehen und es nicht unter das Motto „Hauptsache Schein“ stellen - abgesehen davon, daß diese Einstellung im Team durchaus zu Konflikten führen kann. Wir jedenfalls werden unseren Teil dazu beitragen, daß das nächste halbe Jahr für Sie interessant und lehrreich wird.

Ihnen liegt nun die konkrete Aufgabenstellung des Praktikums im WS 2000/2001 vor. Erschrecken Sie nicht über den Umfang! Er erklärt sich daraus, daß wir im Praktikum aus Zeitgründen den Schwerpunkt auf die Entwurfs- und Implementierungsphase legen müssen und deshalb bereits sowohl eine Produktskizze als auch eine vollständige Moderne Strukturierte Analyse (MSA) des zu realisierenden Softwaresystems angefertigt haben. Die knappe Produktskizze alleine reicht nicht aus, Sie hinreichend mit der zugrundeliegenden Anwendungsdomäne (hier die Verwaltung von studentischen Hilfskräften an einem Universitätslehrstuhl) vertraut zu machen, und bei einer „realitätsnahen“ Aufgabe kommt eine MSA mit ca. 35 Seiten (davon alleine 16 Seiten Prozeßspezifikationen) schnell zusammen. Sie können sicher

sein, daß wir uns die beträchtliche Arbeit nicht machen würden, wenn wir nicht aus Erfahrung wüßten, wie sehr eine ausführliche und genaue Spezifikation sich in den Folgephasen auszahlt. Unser Arbeitsaufwand soll Ihre Chancen für ein erfolgreiches Absolvieren des SoPra's erhöhen und zugleich Ihre Arbeit erleichtern.

Abschließend noch einen kurzen Überblick über dieses Dokument: Im ersten Kapitel skizzieren wir den Ablauf des Praktikums, nennen Termine und legen fest, was Gegenstand der ersten Heimphase ist und welche Ausarbeitungen wir von Ihnen genau erwarten. Das zweite Kapitel beinhaltet die Produktskizze des zu entwickelnden Systems und im dritten Kapitel finden Sie die aus ER-Diagramm, Datenflußdiagrammen, Prozeßspezifikationen und dem Datenlexikon bestehende MSA.

Wir wünschen Ihnen bei der Bearbeitung viel Erfolg.

Ihre Praktikumsbetreuer

Kapitel 1

Allgemeines und Formalia

1.1 Genereller Ablauf

Um Ihnen noch einmal vor Augen zu führen, was im Softwarepraktikum auf Sie zukommt, und einen Eindruck davon zu geben, wie umfangreich und zeitintensiv eine Teilnahme ist, skizzieren wir zunächst kurz den Ablauf des Praktikums.

Das Softwarepraktikum beginnt für jeden Teilnehmer mit dem Anfertigen eines eigenen modularen Grob- und Feinentwurfs ausgehend von der Modernen Strukturierten Analyse des geplanten Softwaresystems, hier eines *Hilfskraft-Verwaltungssystems (HiKS)*. Gruppenarbeit ist in dieser ersten Phase weder erwünscht noch erlaubt, denn Sinn und Zweck dieses Vorgehens ist, daß **jeder** Teilnehmer

- seine Entwurfskenntnisse an dem konkreten Projekt nachweist und dazu ggf. auffrischt sowie
- sich intensiv mit dem Anwendungsgebiet vertraut machen.

Dadurch ist sichergestellt, daß alle Teilnehmer zur ersten Präsenzphase einen annähernd gleichen Kenntnisstand besitzen und aktiv an den Diskussionen teilnehmen können.

Teilnehmer, welche die Hürden des Grob- und Feinentwurfs überspringen, werden zur ersten Präsenzphase eingeladen. Teilnehmer, die einen schlechten oder unvollständigen modularen Grob- und/oder Feinentwurf abliefern, werden vom weiteren Praktikum ausgeschlossen. In der ersten Präsenzphase gilt es dann, sich in Teams von ca. 10 Personen auf einen gemeinsamen Entwurf zu einigen und diesen bis zum Implementationsentwurf auszuarbeiten, d.h. bis hin zu konsistenten, kompilierfähigen Modula-2 Definitionsmoduln. Nach Aufteilung der Moduln auf die Teammitglieder werden diese in „Heimarbeit“ implementiert und modulgetestet. In der zweiten Präsenzphase erfolgen die Integration und der Abschlußtest. Das Vervollständigen der System- und Benutzerdokumentation in einer letzten Heimphase bildet den Abschluß des Praktikums.

Nur das erfolgreiche Absolvieren jeder einzelnen genannten Phase garantiert den Gesamterfolg. Wir warnen noch einmal davor, den Arbeitsaufwand auf die leichte Schulter zu nehmen. Nehmen Sie sich parallel zum Praktikum nicht zu viel anderes vor und beginnen Sie sowohl mit dem Entwurf als auch mit der Implementation und dem Testen so früh wie möglich. Arbeiten Sie sich schnell in die Analyse ein, um ggf. auftretende inhaltliche Probleme rechtzeitig (zu den unten angegebenen Sprechzeiten) mit den Betreuern zu klären.

1.2 Von Ihnen abzuliefernde Dokumente

Als Ergebnis Ihrer Entwurfsarbeit erwarten wir von Ihnen zu den angegebenen Terminen (Abschnitt 1.4) jeweils die folgenden Dokumente:

1. Grobentwurf

- Komponentengraph des HiKS nach Transformation der MSA in einen datenstrukturorientierten Grobentwurf.
- Modulspezifikationen der folgenden Teilsysteme bzw. Funktionsmodule des Grobentwurfs einschließlich der Semantikbeschreibungen relevanter Operationen der Exportschnittstelle:
 - a.) TS „Konditionen“
 - b.) FM „Vertragsbeziehungen“
 - c.) TS „Hilfskraftbeauftragter“
 - d.) TS „Automatischer Betrieb“
- Begründungen aller Entwurfsentscheidungen, die nicht unmittelbar nachvollziehbar sind. Die Teile des Grobentwurfs, die sich durch unmittelbares Anwenden von Transformationsregeln ergeben, brauchen dagegen nicht näher erläutert werden.

2. Feinentwurf

- Komponentengraph des HiKS-Feinentwurfs. Nutzen Sie dabei die im Kurs dargestellten Möglichkeiten z.B. zum Kollabieren von Teilsystemen, um eine übersichtliche Präsentation zu erreichen.
- Spezifikationen folgender Teilsysteme des HiKS-Feinentwurfs einschließlich der Semantikbeschreibungen relevanter Operationen der Exportschnittstelle und der graphischen Darstellung aus Sicht des Entwerfers:
 - a.) TS „Konditionen“
 - b.) TS „Administrator“
 - c.) TS „Virtuelles Dateisystem“

Semantikbeschreibungen können weggelassen werden, sofern sie offensichtlich sind (z.B. der **Create**-Operation), aber nur dann. Im Zweifel spezifizieren Sie lieber zu viel als zu wenig, um Unklarheiten und Ungenauigkeiten zu vermeiden.

- ❑ Informelle Beschreibungen der Rümpfe aller Moduln des TS „Konditionen“, die komplexe Datenstrukturen verkapseln.
- ❑ Prozessspezifikationen in Pseudocode aller Operationen des TS „Konditionen“, die nicht sinnvoll durch Vor- und Nachbedingungen beschrieben werden können, sich stark auf importierten Operationen abstützen oder komplizierte Abläufe oder Algorithmen realisieren.
- ❑ Veranschaulichen Sie die Zerlegung der Operation „VertragskonditionGeprüftEinfügen“ durch Angabe der Aufrufstruktur wie in Abb. 19.3 auf Seite 338 des Kurstextes gezeigt (also mit modulweise zusammengefaßten Operationen als Knoten und gerichteten Kanten von den benutzenden zu den benutzten Operation).
- ❑ Begründungen aller Entwurfsentscheidungen des Feinentwurfs, die nicht unmittelbar nachvollziehbar sind.
- ❑ Skizzen der dem Benutzer „Hilfskraftbeauftragter“ zugeordneten Fenster der Benutzungsoberfläche und eine Zuordnung der dort angebotenen Operationen zu den exportierten Operationen der Module/Teilsysteme Ihres Feinentwurfs.

Achten Sie bei Ihrer Ausarbeitung immer darauf, ob sie uns den Eindruck vermittelt, daß Sie die Analyse des Systems richtig verstanden haben und mittels modularer Techniken sicher in einen funktionsfähigen Entwurf umsetzen können. Da es nicht DEN besten Entwurf gibt, sind Begründungen an entscheidenden Stellen besonders wichtig.

Die Ausarbeitung kann natürlich handschriftlich erfolgen. Um Ihnen jedoch die „Loseblattsammlung“ und uns Probleme mit der Lesbarkeit zu ersparen, empfehlen wir Ihnen dringend, die Möglichkeiten Ihres PC's zu nutzen und nur in Ausnahmefällen handschriftliche Anmerkungen hinzuzufügen. Denken Sie daran, daß unleserliche Teile von uns nicht korrigiert werden können. Aus Korrekturgründen müssen Sie sich bei der Darstellung der Spezifikationen und Komponentengraphen an die Formate des Kurses 1793 „Software Engineering I“ halten.

1.3 Bearbeitungshinweise

In diesem Abschnitt geben wir Ihnen einige Tips zur Behandlung der Analyse und ihrer Übertragung in den Entwurf. Sie sollten diese Hinweise sorgfältig lesen und unbedingt beherzigen, da sie Ihnen die Bearbeitung erleichtern oder aber Punkte ansprechen, auf die es uns bei der Bewertung der Entwürfe besonders ankommt.

- ❑ In den Analysedokumenten sind einige Informationen redundant, allerdings in unterschiedlichen Abstraktionsgraden aufgeführt. Maßgeblich für die **operationalen An-**

forderungen des HiKS ist stets die MSA in Kapitel 3. Bei **nicht-operationalen Anforderungen** gilt dagegen das Bedarfsdokument in Kapitel 2.

- ❑ Unvollständige Prozeßspezifikationen oder Fehler bzw. Ungenauigkeiten in der Analyse müssen von Ihnen in der MSA nicht vervollständigt werden! Größere Fehler oder Inkonsistenzen, vor denen auch wir nicht gefeit sind, teilen Sie uns jedoch bitte möglichst gesammelt mit.
- ❑ Verwenden Sie die im Kurs angegebenen Transformationsregeln für die MSA.
- ❑ Entwurfsschablonen (z.B. von Behältern) oder Komponenten, die im Kurs bereits besprochen und spezifiziert wurden, müssen nicht aufgeführt werden. Ein entsprechender Verweis reicht hier aus. Beachten Sie, daß für Daten mit explizit definierten Schlüsselfeldern keine separate ID-Verwaltung in den Behältern notwendig ist.
- ❑ Für den Feinentwurf des TS „Konditionen“ muß eine Datenstruktur entworfen werden, die Konditionen im Zeitverlauf fortführt. Dabei ist sicherzustellen, daß zu jedem Zeitpunkt von Beginn der Systemlaufzeit an genau eine Kondition gültig ist. Mit jeder neuen Kondition endet also automatisch die Gültigkeit der bis dato aktuellen Kondition. Die aus dem Kurs bekannte (sortierte) Objektliste erfüllt diese Anforderungen nicht, so daß Sie sich selbst Gedanken machen und ggf. den Entwurf eigener generischer Komponenten in Erwägung ziehen müssen.
- ❑ Datenflüsse, die von aktiven Begrenzern ausgehen bzw. an diese gesendet werden, definieren die Benutzungsoberfläche des Systems. Bei der Spezifikation der Benutzungsoberfläche sollen nur ASCII-Eingabemasken entworfen werden. Zum einen erlaubt auch die später zur Verfügung stehende Modula 2-Bibliothek nur einfache Menüs und Eingabefelder, zum anderen bildet der Entwurf der Anwendung und nicht der Benutzungsschnittstellenentwurf den Schwerpunkt dieser Praktikumsphase.

1.4 Termine

Wir erwarten den datenstrukturorientierten **Grobentwurf** des HiKS bis zum **6.11.2000** und den (ggf. um den überarbeiteten Grobentwurf ergänzten) **Feinentwurf** bis zum **18.12.2000** (genauer: jeweils die in Abschnitt 1.2 geforderten Dokumente). **Es gilt das Datum des Poststempels!**

1.5 Betreuung und Sprechzeiten

Ab Mittwoch, den 4.10.2000, stehen Ihnen während der Entwurfsphase die Betreuer jeweils mittwochs von 15.00 bis 17.00 unter der Nummer 02331-987-2103 für Rückfragen zur Verfügung. In der übrigen Zeit erreichen Sie unter dieser Nummer einen Anrufbeantworter. Geben Sie dann Datum und Uhrzeit Ihres Anrufs sowie das Stichwort „SoPra“ an, nennen Sie

Ihren Namen und wann Sie unter welcher Nummer erreichbar sind. Schildern Sie Ihr Problem, wir rufen Sie schnellstmöglich zurück.

Als Diskussionsforum des Softwarepraktikums steht Ihnen außerdem die NeWS-Gruppe `feu.informatik.sopra` zur Verfügung.

1.6 Kriterien für das Bestehen der erste Heimphase

Wie erwähnt, sollen Sie sich mit Ihren Grob- und Feinentwürfen des HiKS die Teilnahmeberechtigung an der ersten Präsenzphase erwerben. Beachten Sie, daß verspätet eingehende Entwürfe von uns ebenso wenig akzeptiert werden wie Entwürfe, welche die oben genannten Bedingungen in punkto Vollständigkeit und Lesbarkeit verletzen. Nur formal einwandfreie Entwürfe werden von uns durchgesehen und mit entsprechenden Anmerkungen an die Autoren zurückgesendet.

Die korrigierten Entwürfe werden von uns in drei Klassen eingeteilt:

- Werden Grob- und Feinentwurf ohne Auflagen akzeptiert, so ist der Teilnehmer direkt zur ersten Präsenzphase zugelassen.
- Ein Grobentwurf mit kleineren Mängeln ist vom Autor nachzubessern. Der überarbeitete Grobentwurf muß zum Einsendetermin für den Feinentwurf am 18.12.2000 nachgereicht werden.
- Kleinere Mängel im Feinentwurf müssen bis zum Beginn der ersten Präsenzphase am 8.1.2001 nachgebessert werden. Die Teilnahme an der ersten Präsenzphase hängt von der Qualität der Überarbeitung und der Erfüllung der Auflagen ab.
- Ein mangelhafter Grob- und/oder Feinentwurf, insbesondere ein Entwurf, der wiederholt modulare Konzepte verletzt, führt zum Ausschluß des Autors.

Kapitel 2

Produktskizze des Hilfskraft-Verwaltungssystems (HiKS)

Hilfskraft-Verwaltungssystem (HiKS)

Produktskizze



vorgelegt am 20.08.97
durch LG Praktische Informatik III



- I. Problembeschreibung, Projektziele
- II. Funktionsumfang, Außenverhalten
- III. Benutzerprofil
- IV. Akzeptanzkriterien
- V. Entwicklungs- Einsatz- und Wartungsumgebung, Schnittstellen, Nebenbedingungen
- VI. Lösungsstrategien
- VII. Informationsquellen, Glossar

Produktskizze	HiKS Endversion
I. Problembeschreibung, Projektziele	<p>Gegenstand des Projekts ist die automatische Verwaltung von Hilfskräften, ihrer Verträge und der zur Verfügung stehenden Etats. Primäre Ziele sind:</p> <ul style="list-style-type: none">• Informationen über Hilfskräfte und deren Verträge bereitzustellen, die den Hilfskraft-Beauftragten bei der Erstellung von Statistiken und Zeugnissen unterstützen,• Neuverträge bzw. Vertragsänderungen haushaltstechnisch und auf ihre vertragsrechtliche Korrektheit zu überprüfen,• Kostenstellen zu verwalten, Verträge auf diesen Kostenstellen automatisch zu buchen und aktuelle Kontostände für Planungszwecke zu liefern und• Weihnachtsgeldzahlungen sowie die Etatverwaltung zu automatisieren. <p>An das System soll ein Laserdrucker angeschlossen werden, so daß z.B. Informationen über eine Hilfskraft und ihre Verträge bzw. Informationen über die Kostenstellen, z.B. die (Jahres-) Abschlüsse, ausgedruckt werden können.</p>
II. Funktionsumfang, Außenverhalten	<p>Für die verschiedenen Lehrgebiete der FernUniversität ist ein Hilfskraft-Verwaltungssystem (HiKS) zu erstellen. Funktionsumfang und Außenverhalten lassen sich zunächst grob in die sechs Bereiche Personen, Verträge, Kostenstellen, Informationsbetrieb, automatischer Betrieb und Administration einteilen. Wir geben die wichtigsten Funktionalitäten an:</p> <p>a) Verträge</p> <ul style="list-style-type: none">• Vertrag auf Vollständigkeit und Vertragsbedingungen prüfen• Neueinstellung vornehmen• Vertragsdauer oder -volumen eines Vertrages ändern• Vertrag verlängern• Vertrag kündigen• Vertragskonditionen ändern• Vertragsordner führen <p>b) Personen</p> <ul style="list-style-type: none">• Registrieren der Personendaten von (potentiellen) Hilfskräften• Aufnehmen von Daten für die Einstellung• Personenkartei führen <p>c) Kostenstellen</p> <ul style="list-style-type: none">• Kostenstelle einrichten• Etat festlegen bzw. modifizieren• Noch verfügbare Mittel (einer Kostenstelle) errechnen
HiKS 2	

HiKS Endversion

Produktskizze

- Gehalts- oder Weihnachtsgeldbuchung vornehmen
- Freie Buchung vornehmen
- Buchung stornieren
- Kostenstelle sperren oder liquidieren; Sperrung aufheben
- Kostenstellenverzeichnis führen

d) Informationsbetrieb

Aus Gesprächen mit den Hilfskraftbeauftragten und Personalsachbearbeitern ergibt sich, daß u.a. folgende Arten von Anfragen unterstützt werden müssen:

- Erstelle eine aktuelle Vertragsliste für eine Hilfskraft
- Ermittle alle Verträge einer Hilfskraft innerhalb eines vorgegebenen Zeitraumes (z.B. Jahres) bzw. alle Verträge einer Hilfskraft
- Ermittle alle Hilfskräfte, deren Verträge bis zu einem bestimmten Zeitpunkt auslaufen
- Liefere alle Buchungen auf einer Kostenstelle innerhalb eines Zeitraumes
- Ermittle die insgesamt verfügbaren Haushaltsmittel auf einer oder mehrerer Kostenstellen des Lehrgebiets (in DM, Hilfskraft-Stunden oder als Anzahl damit finanzierbarer Hilfskräfte mit 19h Beschäftigungszeit pro Woche)

e) automatischer Betrieb

- Weihnachtsgelder zum Stichtag (z.B. 1.12) buchen
- Kostenstellen zum Jahres- bzw. Semesterende (je nach Art der Kostenstelle) abschließen

f) Administration

- Backup des kompletten Datenbestandes erstellen
- Backup-Version des Systems einspielen

III. Benutzerprofil

Benutzer des Hilfskraft-Verwaltungssystems sind ausschließlich Angestellte der FernUni. Diese lassen sich in drei Gruppen einteilen:

- Hilfskraft-Beauftragten obliegt die fachliche Seite der Hilfskraft-Verwaltung. Sie halten Kontakt zu interessierten Studenten und leiten Vertragsabschlüsse bzw. -veränderungen unter Beachtung der vorhandenen Etats ein.
- Personalsachbearbeitern obliegt die haushaltstechnische Seite der Hilfskraft-Verwaltung. Sie bearbeiten und überprüfen Verträge und führen die entsprechenden Buchungen durch.
- Administratoren sind für die regelmäßige Sicherung des Datenbestandes verantwortlich (Backups).

Produktskizze	HiKS Endversion
IV. Akzeptanzkriterien	<p>Primär von Interesse sind die Laufzeit einzelner Funktionen sowie die Robustheit und Zuverlässigkeit.</p> <p>Bei den bisher manuellen ausgeführten Tätigkeiten (Vertragsbearbeitung, Informationsbetrieb, Kostenstellenverwaltung) soll die durchschnittliche Bearbeitungsdauer gegenüber dem jetzigen Stand wesentlich verkürzt werden. Bei den mit der Zeit zu erwartenden großen Datenmengen (Löschen von Hilfskräften, Verträgen, Kostenstellen oder Buchungen ist nicht vorgesehen) ist besonders bei komplexen Anfragen im Informationsbetrieb auf die Laufzeit zu achten. Eine Antwortzeit proportional zur Antwortgröße ist anzustreben und z.B. durch geeignete Sortierungen und Indexstrukturen sicherzustellen.</p> <p>Der automatische Betrieb beginnt mit jedem Tagewechsel und muß vor Arbeitszeit der Benutzer beendet sein. Dieser Zeitraum stellt somit eine natürliche Laufzeitschranke dar. Ein hinreichendes Kriterium für die Laufzeit der automatischen Funktionen in Abhängigkeit von der Größe der zu bearbeitenden Daten ist wie folgt zu bestimmen:</p> <p>Da die Zeitpunkte, an denen automatische Funktionen aktiviert werden (Bsp.: Stichtage, Semester- oder Jahreswechsel), im ungünstigsten Fall zusammenfallen können, ist die Summe der maximal zu erwartenden Laufzeiten aller automatischen Funktionen zu bestimmen. Multipliziert mit einem Sicherheitsfaktor von 2 darf dieser Wert 7 Stunden nicht überschreiten.</p> <p>Im laufenden Betrieb sind alle Daten im Hauptspeicher zu halten. Bei Beendigung des Betriebs müssen alle Daten und Beziehungen auf dem Sekundärspeicher gesichert werden. Zusätzlich muß die Möglichkeit bestehen, die Systemdaten während des Betriebs explizit zu sichern und zu einer früheren Version zurückzukehren. Mindestens drei Versionen sind gleichzeitig zu verwalten.</p> <p>Ein modularer, leicht änderbarer Softwareentwurf ist zwingend, da die Berechnungen häufig geändert bzw. später verfeinert werden. Änderungen bei den Berechnungsalgorithmen von Gehältern und Weihnachtsgeld, d.h. den verschiedenen Konditionen, sind ebenfalls die Regel.</p> <p>Beim Testen sind jenen Vorgängen besondere Aufmerksamkeit zu schenken, die Veränderungen von Kontoständen der Kostenstellen verursachen. Buchungen müssen absolut korrekt bearbeitet werden.</p> <p>Implementation, Kommentare und Dokumentation haben den bekannten Regeln und Standards zu gehorchen und werden in einem Abnahme(code)review geprüft.</p>
HiKS 4	

V. Entwicklungs-, Einsatz- und Wartungsumgebung, Schnittstellen, Nebenbedingungen

Jedes Lehrgebiet soll über eine lokale Kopie verfügen, die unabhängig von den Systemen anderer Lehrgebiete arbeitet. Dazu erhält jedes Lehrgebiet einen Rechner, der mit einem graphischen Bildschirm ausgestattet ist. Das System arbeitet im Einbenutzerbetrieb.

Die Entwicklungs-, Wartungs- und Einsatzumgebung des Projektes ist ein LINUX-Betriebssystem. Die Implementierung erfolgt in Modula-2, wobei neben den Modula-2 Standardbibliotheken eine Bibliothek mit generischen Listen und eine Qualitätssicherungsbibliothek zur Auswertung von Vor- und Nachbedingungen sowie Invarianten zur Verfügung stehen. Als Werkzeug für die graphische Oberfläche dient eine Graphikbibliothek mit den wichtigsten Oberflächenobjekten in gängigem Standard (z.B. X-Windows).

Das System besitzt eine Schnittstelle zu einem Laserdrucker, dessen Fabrikat noch unbestimmt ist.

Um die Datenbeschaffung kümmert sich der Auftraggeber. Sie ist nicht Gegenstand des Projekts.

VI. Lösungsstrategien

Die Analyse erfolgt mit MSA. Das Ergebnis wird in einer modularen Grobarchitektur übertragen und anschließend verfeinert.

Beim Speichern werden nur originäre Datenobjekte als Datensätze auf den Sekundärspeicher geschrieben. Indexstrukturen werden also nicht mit gespeichert, sondern beim Laden jeweils neu aufgebaut.

VII. Informationsquellen, Glossar

...

Kapitel 3

Feststellung der operativen Anforderungen mit MSA

Zunächst wollen wir einige Absprachen bezüglich der Notation treffen: In den folgenden Erläuterungen sind die Namen von Zuständen, Prozessen etc. kursiv gedruckt und in Anführungszeichen eingeschlossen. Nummern von Zuständen in Zustandsübergangsdigrammen wird der Buchstabe „Z“, Prozessen in Datenflußdiagrammen der Buchstabe „D“ vorangestellt. Außerdem verzichten wir aus Darstellungsgründen auf die Annotation von Flüssen, die ihr Ziel bzw. ihren Ursprung in Lagern haben, da diese durch den Typ des Lagers eindeutig bestimmt sind.

Die Anforderungsermittlung unseres Hilfskraft-Verwaltungssystems (HiKS) beginnen wir mit den Systemdaten, betrachten danach die Ereignismodellierung und konstruieren zuletzt das Datenflußmodell. Die Systemdaten und die Datenflüsse sind im Datenlexikon genau spezifiziert.

3.1 ER-Modellierung

In den ER-Diagrammen dieses Abschnitts sind nicht immer alle Attribute eines Entitätstyps aufgeführt, um die Diagramme nicht zu überladen. Der genaue Aufbau eines Entitätstyps ist jedoch im Datenlexikon spezifiziert.

Im Problembereich wird zwischen Hilfskräften und Kandidaten unterschieden, also Personen, die bereits als Hilfskraft gearbeitet haben, und solchen, die erst noch Hilfskräfte werden wollen. Bei der ER-Modellierung verzichten wir jedoch zur Vereinfachung auf die Bildung entsprechender Subentitätstypen "*Kandidat*" und "*Hilfskraft*" von einem Entitätstyp "*Person*", sondern ordnen dem Entitätstyp "*Hilfskraft*" zusätzlich zu Namen, Geburtsdatum und Geschlecht auch Attribute wie "*Adresse*" oder "*Interessen*" zu. Der genaue Aufbau einer Adresse kann dem Datenlexikon entnommen werden. Wir erhalten auf diese Weise das Teildigramm in Abb. 3.1.

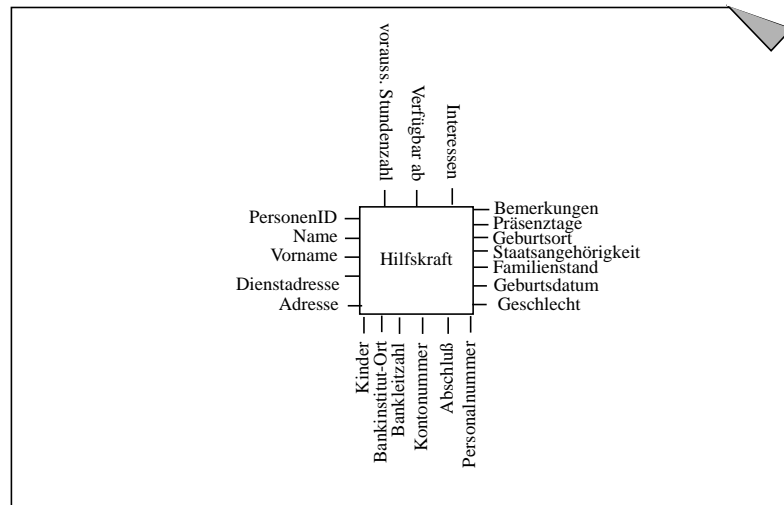


Abb. 3.1 ER-Diagramm des HiKS: Hilfskräfte.

Verträge stehen als verbindendes Element von Hilfskräften und Kostenstellen im Mittelpunkt des HiKS. Die Vertragslänge und das Vertragsvolumen bestimmen die Höhe der Gehaltsbuchungen für die Hilfskraft auf der jeweiligen Kostenstelle. Als Attribute eines Entitätstyps "Vertrag" extrahieren wir das Erstellungsdatum, Anfangs- und Endedatum, die Stundenzahl pro Woche, die Tätigkeit (Korrektor, Präsenz-Hilfskraft), eine Begründung (freier Text) sowie ein Stornodatum (Abb. 3.2).

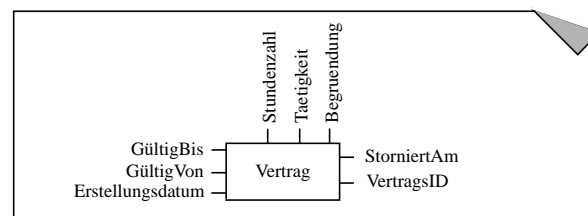


Abb. 3.2 ER-Diagramm des HiKS: Verträge.

Im Problembereich existieren zwei Arten von Kostenstellen, die sich durch die Art ihrer Buchungen, ihre „Lebensdauer“ und insbesondere die Berechnung ihrer (Anfangs-) Etats stark unterscheiden. Eine Lehrgebietskostenstelle besitzt einen festen Jahresetat. Nicht ausgegebene Beträge fallen am Jahresende an die Universität zurück und die Kostenstelle wird durch eine entsprechende Kostenstelle im Folgejahr ersetzt. Kostenstellen für Korrekturhilfskräfte sind lastabhängig, d.h. ihre Etats errechnen sich aufgrund fester Regeln aus der Anzahl der

Kursbeleger, den tatsächlichen Einsendungen und Klausuren und den Semesterwochenstunden der Veranstaltung.

Obwohl sich die Kostenstellen "*Lehrgebiet*" und "*Kurs*" stark unterscheiden, besitzen sie einige gemeinsame Attribute, die wir bei der ER-Modellierung herausdestillieren. Die KostenstellenID wird explizit vom Personalsachbearbeiter angegeben. Beide werden zu einem bestimmten Zeitpunkt eingerichtet, können zeitweilig gesperrt sein und werden schließlich irgendwann liquidiert. Wir fassen diese Attribute in einem Entitätstyp "*Kostenstelle*" zusammen und modellieren Subentitätstypen für die zwei Kostenstellenarten, welche die kostenstellenspezifischen Attribute erhalten (Abb. 3.3).

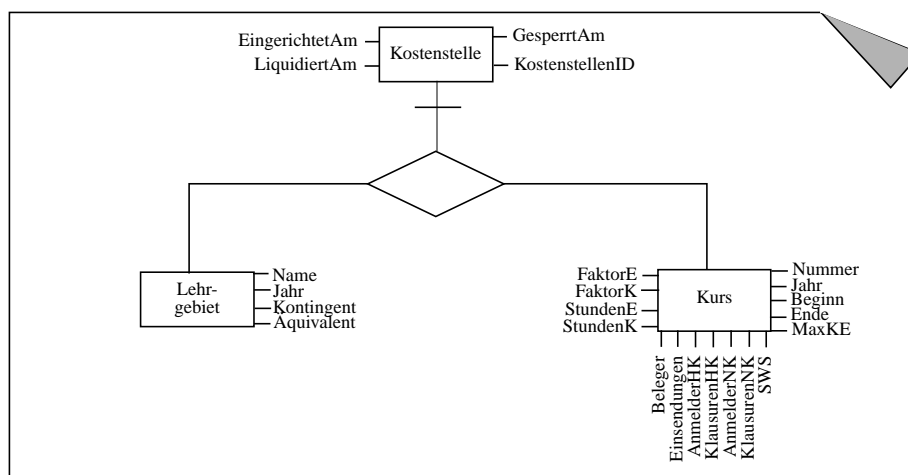


Abb. 3.3 ER-Diagramm des HiKS: Kostenstellen.

Verträge sind jeweils einer Hilfskraft und einer Kostenstelle eindeutig zugeordnet. Bei Vertragsabschluß erfolgt eine Gehaltsbuchung zu Lasten dieser Kostenstelle. Im Vertragsverlauf können weitere Buchungen aufgrund von Weihnachtsgeldzahlungen oder durch Änderung von Tarifkonditionen notwendig werden. Außerdem können auch Buchungen allgemeiner Art zu Lasten einer Kostenstelle erfolgen. Wir modellieren diese Sachverhalte durch die Einführung eines Entitätstyps "*Buchung*" und zwei Beziehungstypen "*belastet*" und "*hatBuchung*" (Abb. 3.4).

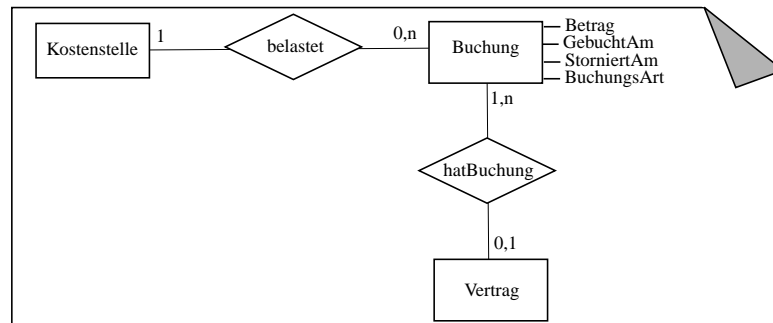


Abb. 3.4 ER-Diagramm des HiKS: Buchungen.

Wie bereits erwähnt sind Hilfskräfte und Kostenstellen über Verträge miteinander verknüpft. Es liegt also nahe, den "Vertrag" als assoziativen Entitätstyp "Vertrag" zu modellieren (Abb. 3.5).

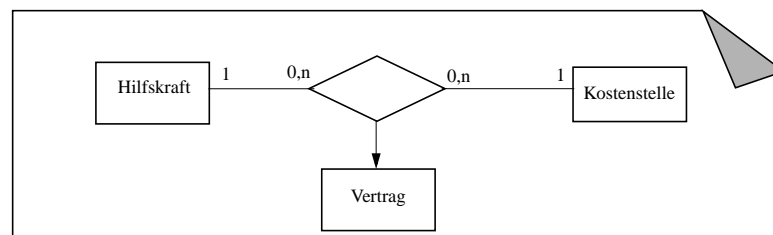


Abb. 3.5 ER-Diagramm des HiKS: Vertragsbeziehungen.

Die gesetzlichen Rahmenbedingungen, die bei Abschluß neuer Verträge und der Berechnung von Gehältern berücksichtigt werden müssen, sind in den "Tarif-", "Weihnachtsgeld-" und "Vertragskonditionen" festgelegt. Jede "Kondition" ist für einen bestimmten Zeitraum gültig. (Abb. 3.6)

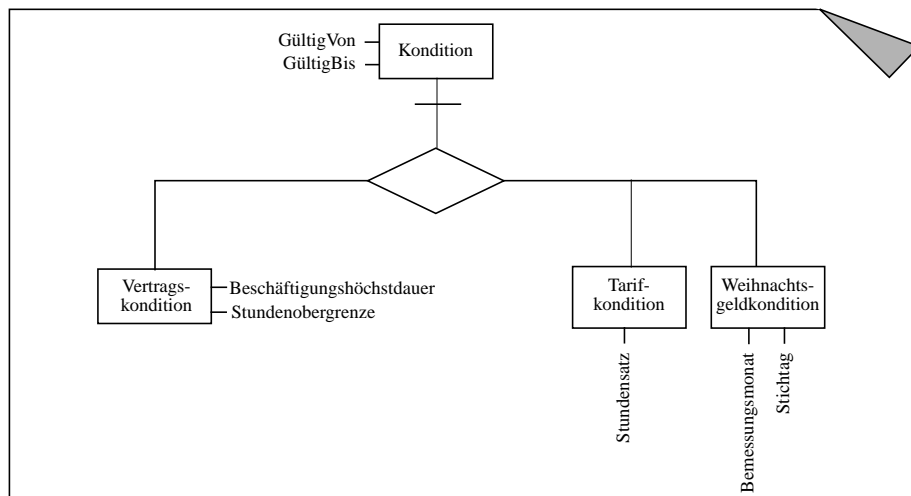


Abb. 3.6 ER-Diagramm des HiKS: Konditionen.

Die Teildiagramme aus Abb. 3.1 bis Abb. 3.6 fassen wir abschließend noch einmal in einer Übersicht zusammen (Abb. 3.7). Auf die erneute Angabe der Attribute verzichten wir hierbei.

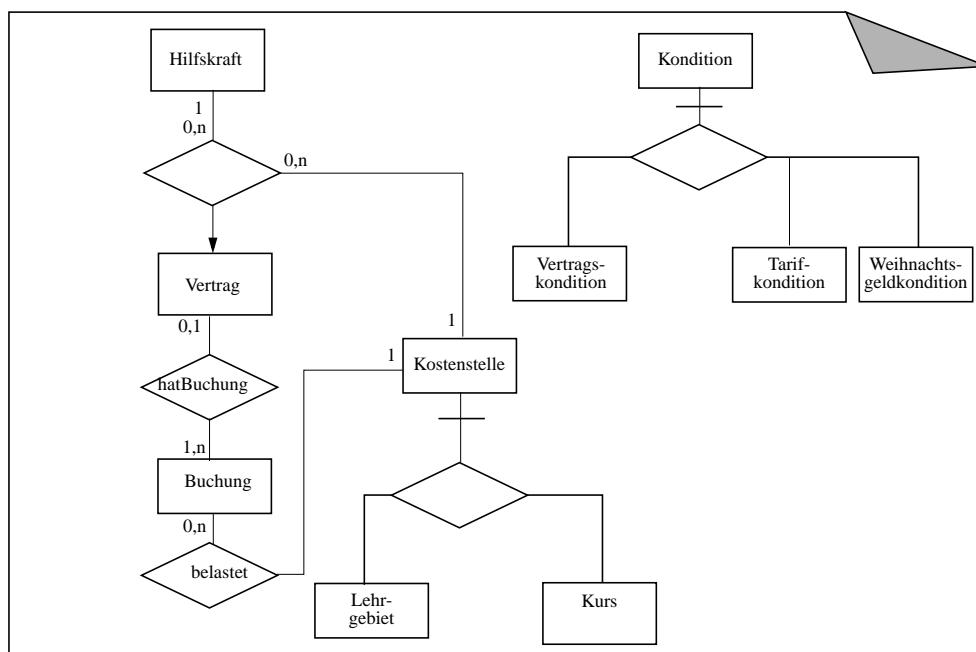


Abb. 3.7 Das ER-Diagramm des Hilfskraft-Verwaltungssystems.

3.2 Modellierung der Systemzustände

Als Zustand des Systems können wir zunächst den normalen "*Bedienerbetrieb*" identifizieren. Desweiteren muß der "*Bedienerbetrieb*" an einigen Stichtagen unterbrochen werden, um abhängig vom Datum eine Reihe periodischer Berechnungen und Buchungen (z.B. Weihnachtsgeld) durchzuführen. Wir modellieren dies, indem wir vom Bedienerbetrieb aus bei jedem Tageswechsel zunächst in den "*Automatischen Betrieb*" wechseln. Operationen, die an diesem Tag anfallen (vgl. das Buchen fälliger Daueraufträge im Kurs), werden angestoßen und nach ihre Abarbeitung der "*Automatische Betrieb*" und in Richtung "*Bedienerbetrieb*" verlassen.

Der Systemverwalter ist für die Durchführung der Datensicherung und das Zurückspielen von Daten zuständig. Diese Operationen kann er natürlich nur ausführen, wenn keine anderen Aktionen innerhalb des Systems ausgeführt werden. Wir identifizieren also einen dritten Zustand, die "*Systemverwaltung*", in den auf Anforderung des Beauftragten oder des Personalsachbearbeiters gewechselt wird. Er ist gleichzeitig Startzustand des Systems, d.h. der Systemverwalter ist für das Hochfahren des Systems zuständig. Hat der Systemverwalter seine Aufgaben beendet, wird der Zustand "*Systemverwaltung*" in Richtung "*Bedienerbetrieb*" wieder verlassen (Abb. 3.8).

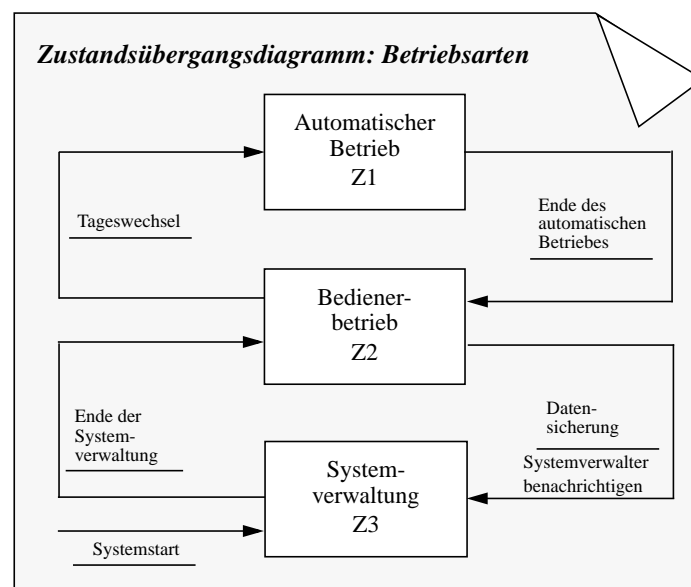


Abb. 3.8 Das rudimentäre Ereignismodell des HiKS.

Die Ereignismodellierung unseres Fallbeispiels gibt alles in allem nicht viel her. Es gibt keinen Endzustand, die Übergänge zwischen den beiden Zuständen lösen in der Regel keine Aktionen aus und es fehlen externe Ereignisse.

3.3 Datenflußmodellierung

Aus dem Ereignismodell in Abb. 3.8 leiten wir (gemäß der Balancierregeln aus Abschnitt 11.1 im Kurs) den Kontrollprozeß "Betriebsarten" ab, der die Prozesse "Bedienerbetrieb", "Automatischer Betrieb" und "Systemverwaltung" steuert. Im "Bedienerbetrieb" kommuniziert das System mit dem Beauftragten, dem Personalsachbearbeiter, der Uhr und einem Drucker. Im "Automatischen Betrieb" werden dem Drucker Formulare geschickt und das Tagesdatum von der Uhr erfragt. Innerhalb der "Systemverwaltung" veranlaßt der Systemverwalter das Speichern und Laden eines Systemzustandes. Dabei werden Daten an das Dateisystem geschickt oder von diesem geholt (Abb. 3.9).

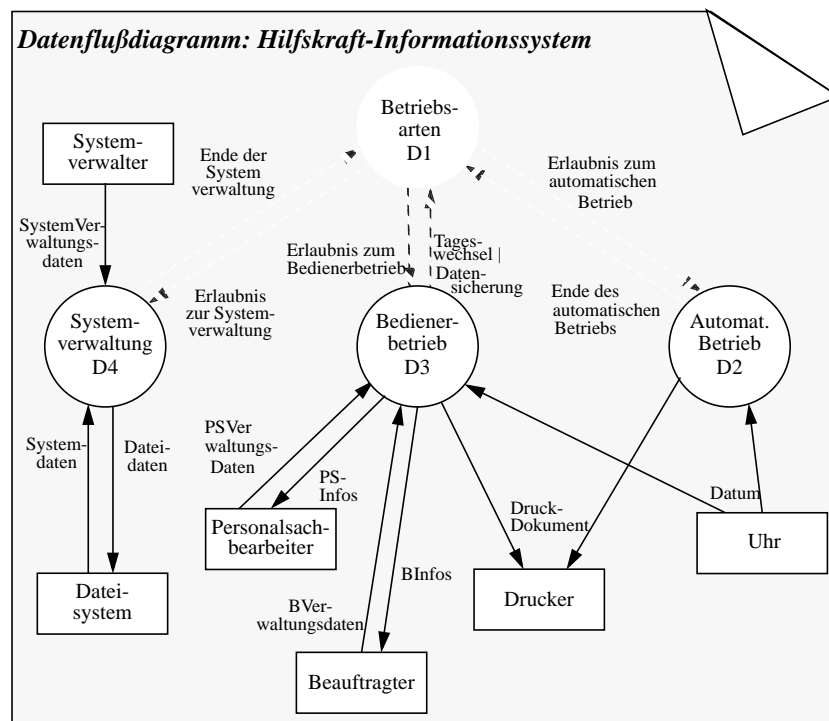


Abb. 3.9 HiKS-Datenflußmodellierung: Umsetzung des Zustandsübergangsdigramms.

Durch Zusammenfassen der Prozesse D1 - D4 erhalten wir unmittelbar das Kontextdiagramm des Datenflußmodells (Abb. 3.10).

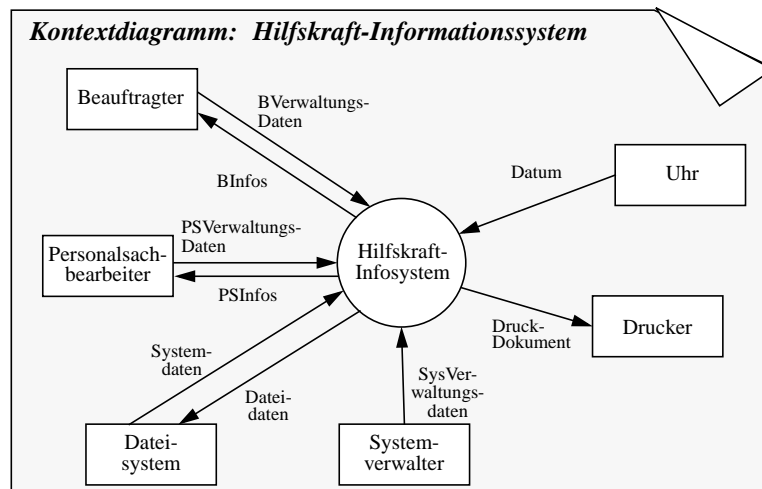


Abb. 3.10 Kontextdiagramm des HiKS.

Die beiden bisherigen Datenflußdiagramme bilden die höchsten Abstraktionsstufen des Datenflußmodells.

Im folgenden gehen wir nicht top down vor, was eine rein funktionale Zerlegung bedeuten würde, sondern wählen stattdessen den outside in-Ansatz. Wir identifizieren folgende Schnittstellen zwischen Systemfunktionalitäten und Begrenzern:

- "*Beauftragter*": Verwaltung von Hilfskraftdaten, Abschließen und Bearbeiten von Verträgen, Informationsretrieval;
- "*Personalsachbearbeiter*": Verwaltung der Kostenstellen und der Konditionen, Informationsretrieval;
- "*Systemverwalter*": Datensicherung, Zurückspielen von Daten;
- "*Uhr*": Systemdatum aktualisieren;
- "*Drucker*": Ausgabeaufbereitung;
- "*Dateisystem*": Speichern und Auslesen von Daten.

Die Funktionalitäten formulieren wir als neue Prozesse. Anschließend versuchen wir, die Prozesse in Diagrammen zu organisieren und verbindende Datenflüsse zu spezifizieren. Dabei muß auch eine eindeutige Zuordnung zu den Oberprozessen "*Automatischer Betrieb*" und "*Bedienbetrieb*" gefunden werden. Da der Prozeß "*Systemverwaltung*" lediglich die Aufgabe hat, einen Systemzustand zu sichern bzw. einen gesicherten Systemzustand zu laden, wollen wir ihn nicht weiter verfeinern.

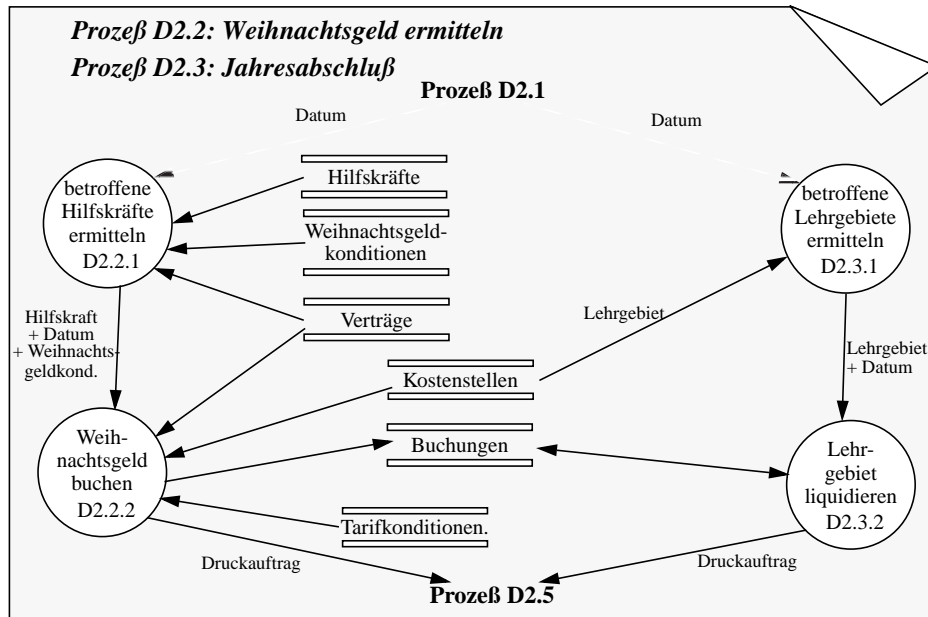


Abb. 3.12 HiKS-Datenflußmodellierung: Verfeinerung der Prozesse D2.2 und D2.3.

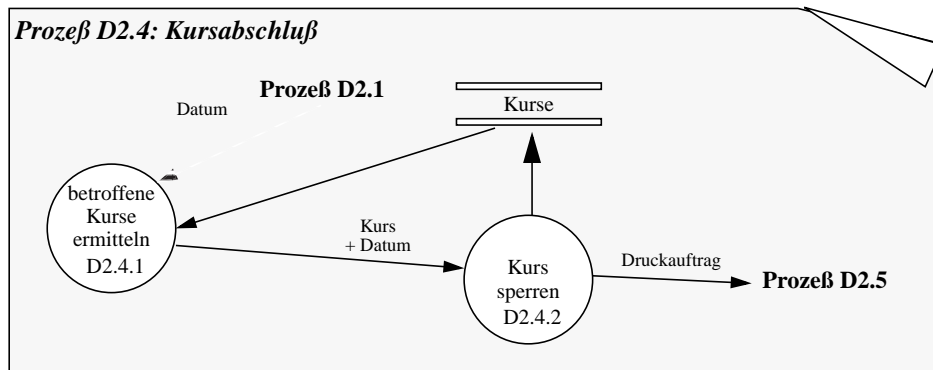


Abb. 3.13 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses D2.4.

Prozeß D3 "Bedienbetrieb". Auch der "Bedienbetrieb" kann in eine Reihe unterschiedliche und weitgehend voneinander unabhängige Teilaufgaben zerlegt werden. Aufgrund der von uns ermittelten Schnittstellen zwischen System und Begrenzern erhalten wir die Teilprozesse "Kostenstellenverwaltung", "Konditionsverwaltung", "Informationsretrieval", "Hilfskraftverwaltung", "Vertragsverwaltung" und "Datumsverwaltung". Zusätzlich führen wir

noch den Kontrollprozeß "Sicherungsanforderung" ein, der uns den Wechsel vom "Bedienbetrieb" in die "Systemverwaltung" ermöglicht. Dieser Prozeß ist ein elementarer Prozeß, wird also nicht weiter verfeinert.

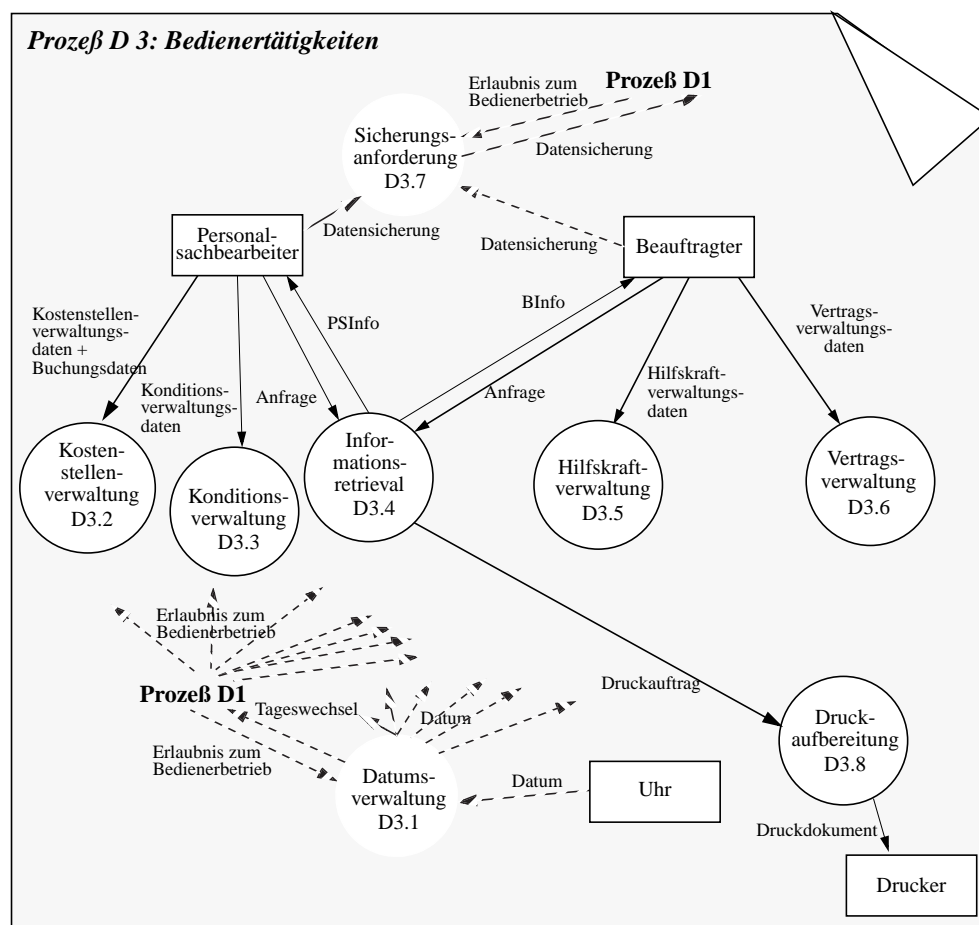


Abb. 3.14 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Bedienertätigkeiten".

Die "Datumsverwaltung" ist ebenfalls ein Kontrollprozeß, der den Übergang vom "Bedienbetrieb" in den "Automatischen Betrieb" ermöglicht und ansonsten das aktuelle Systemdatum zur Verfügung stellt. Auch dieser Prozeß wird direkt als elementarer Prozeß spezifiziert und nicht weiter verfeinert. Ebenfalls nicht weiter verfeinert werden die "Druckaufbereitung" und das "Informationsretrieval" (Abb. 3.14).

Prozeß D3.2 "Kostenstellenverwaltung". Die "Kostenstellenverwaltung" wird in vier weitere Prozesse verfeinert. Der Prozeß "Kostenstellendaten inspizieren" stellt fest, welche Ope-

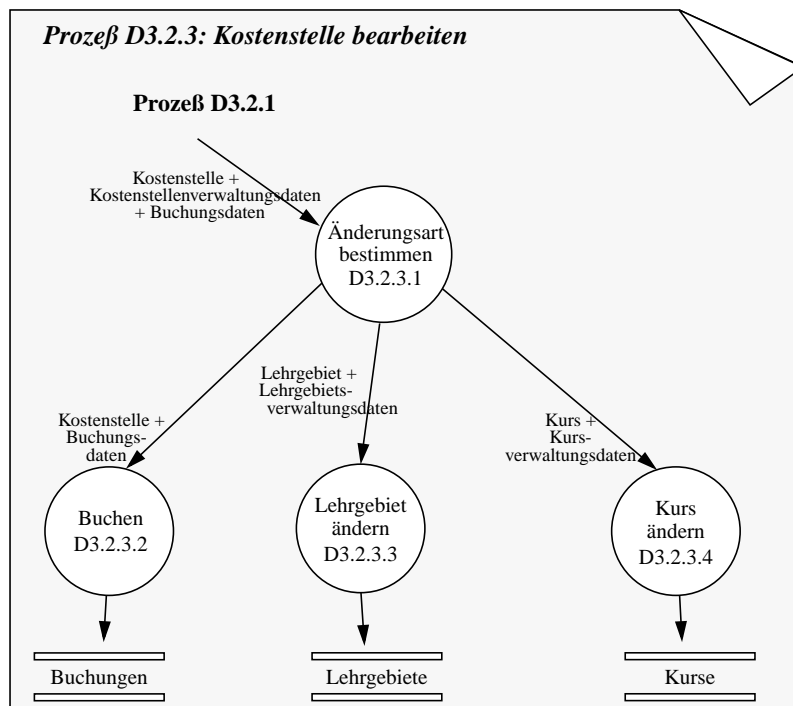


Abb. 3.16 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Kostenstelle bearbeiten".

Prozeß D3.3 "Konditionsverwaltung". Der Datenfluß "Konditionsverwaltungsdaten" wird bei der Verfeinerung des Prozesses "Konditionsverwaltung" in drei disjunkte Datenflüsse aufgeteilt, mit denen jeweils eine Konditionsart geändert werden soll (Abb. 3.17). Ändern einer geltenden Kondition bedeutet, daß ein neuer Konditionssatz erstellt und im entsprechenden Konditionslager abgelegt wird. Da Konditionen auch rückwirkend geändert werden dürfen, müssen ggf. eine Reihe von Gehaltsberechnungen neu vorgenommen bzw. korrigiert werden. Näheres kann wieder den Prozeßspezifikationen entnommen werden.

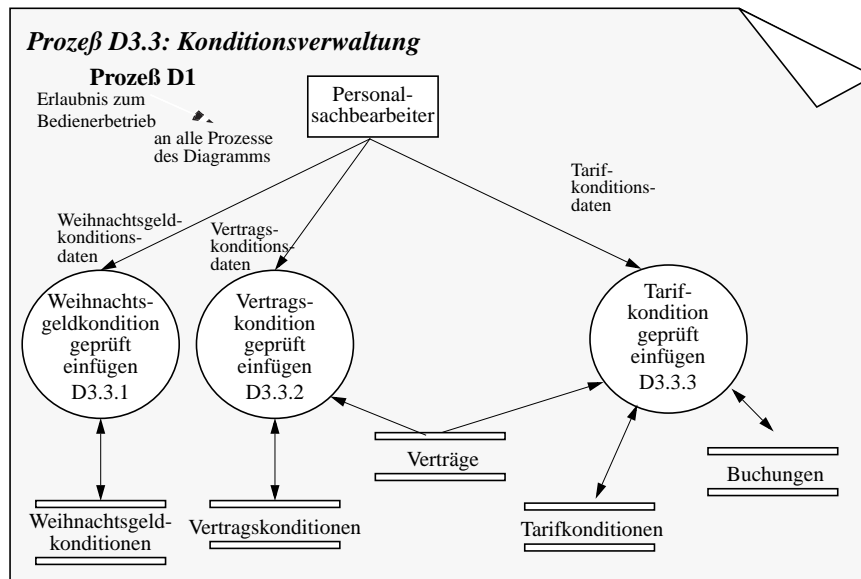


Abb. 3.17 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Konditionsverwaltung".

Prozesse D3.5 "Hilfskraftverwaltung". Innerhalb der "Hilfskraftverwaltung" kann eine neue Hilfskraft (quasi als „Kandidat“) aufgenommen und die Daten einer Hilfskraft geändert und ergänzt werden. Der Datensatz einer Hilfskraft kann nur gelöscht werden, wenn keine Verträge mit dieser Hilfskraft existieren bzw. jemals existiert haben (Abb. 3.18).

Da die Semantik dieser Prozesse klar sein sollte, haben wir auf die entsprechenden Prozeßspezifikationen verzichtet.

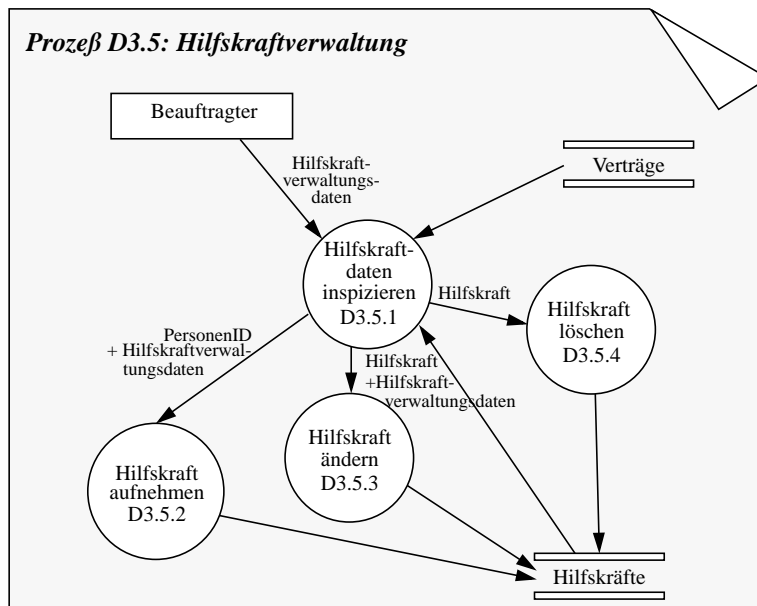


Abb. 3.18 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Hilfskraftverwaltung".

Prozeß D3.6 "Vertragsverwaltung". Der Beauftragte kann neue Verträge anlegen oder aber bestehende Verträge ändern. Wir verfeinern den Prozeß D3.6 also in die Prozesse D3.6.1 "Neuvertrag bearbeiten" und D3.6.2 "Vertrag ändern" (Abb. 3.19).

Möchte der Beauftragte einen neuen Vertrag für eine Hilfskraft abschließen (Abb. 3.20), so müssen folgende Vorbedingungen überprüft werden:

1.) Die Hilfskraft darf die Vertragskonditionen nicht verletzen. Um dies sicherzustellen, müssen alle Verträge der Hilfskraft darauf geprüft werden, ob die Stundenzahl der Hilfskraft im Zeitraum des neu einzurichtenden Vertrages nicht die maximal zulässige Stundenzahl überschreitet und ob die Gesamtarbeitszeit der Hilfskraft nicht länger als die Höchstarbeitszeit wird.

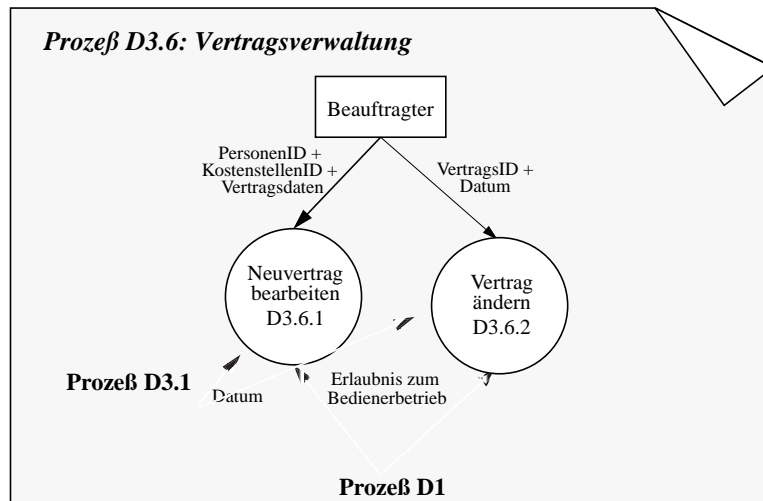


Abb. 3.19 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Vertragsverwaltung".

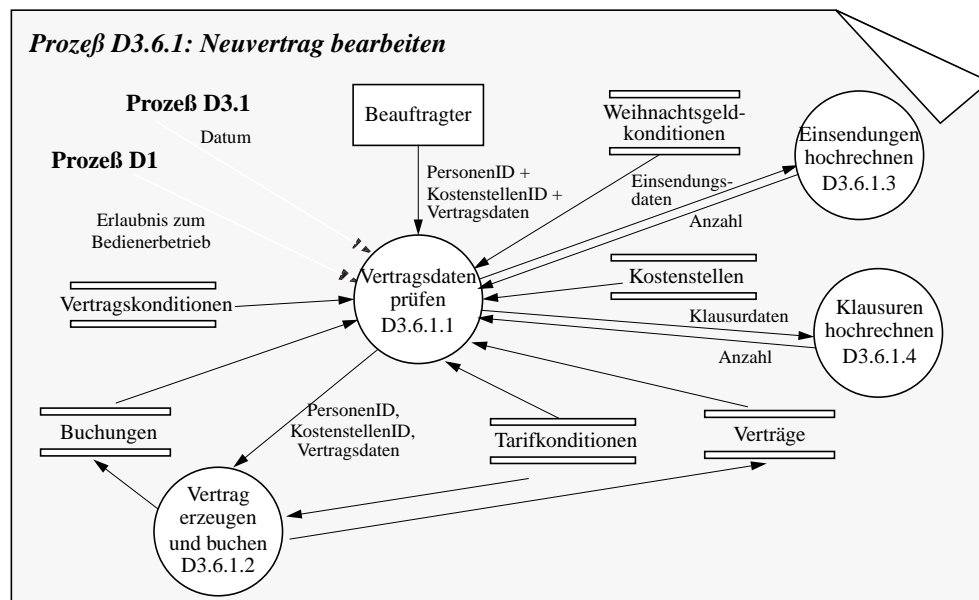


Abb. 3.20 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Neuvertrag bearbeiten".

2.) Der Etat der Kostenstelle darf nicht überzogen werden. Zusätzlich müssen die Etats aller Kostenstellen, zu denen die Hilfskraft im Jahr des neu einzurichtenden Vertrages weitere Verträge hat, eventuell mit einer zusätzlichen Weihnachtsgeldbuchung belastet werden, falls der Vertrag den Stichtag für Weihnachtsgeldzahlungen enthält.

Der Beauftragte ändert einen Vertrag (Abb. 3.21), indem er das Vertragsende vorzieht oder den Vertrag storniert. Bei einem stornierten Vertrag gilt außerdem Vertragsende = Vertragsanfang. In beiden Fällen muß die Buchungen neu vorgenommen werden.

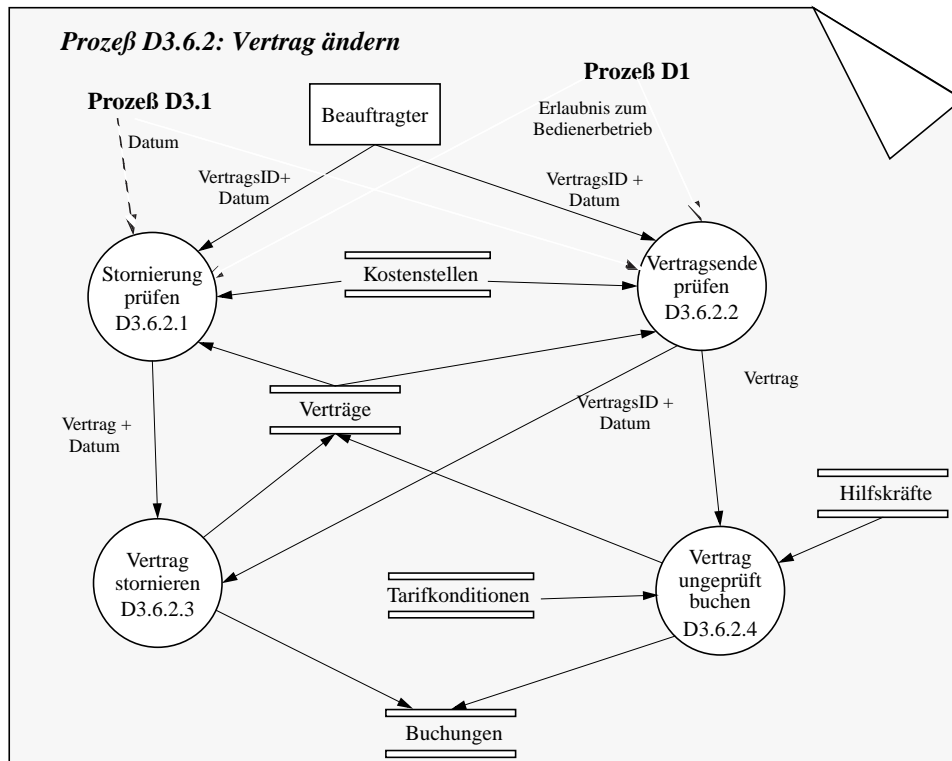


Abb. 3.21 HiKS-Datenflußmodellierung: Verfeinerung des Prozesses "Vertrag ändern".

3.4 Prozeß-Spezifikationen

In diesem Abschnitt folgen die Spezifikationen aller Prozesse der Datenflußdiagramme. Einzige Ausnahme bilden die Teilprozesse des Prozesses D3.5, deren Semantik direkt aus den Beschreibungen im letzten Abschnitt hervorgehen. Die Prozeß-Spezifikationen sind nach ihren Nummern sortiert. Wir weisen noch auf einige Besonderheiten hin, die für das Verständnis und die richtige Umsetzung in den Entwurf wichtig sind:

- Da Oberprozesse bei Spezifikationen nicht viel hergeben, haben wir uns auf die elementaren Prozesse konzentriert.
- Die Spezifikationen werden in Pseudocode angegeben, der bereits große Ähnlichkeiten mit einer Programmiersprache besitzt. Denken Sie jedoch immer daran, daß diese

operationale Art der Semantikdefinition nicht bedeutet, daß sie die Prozesse 1:1 in Prozeduren übertragen können und den suggerierten Kontrollfluß einfach übernehmen können. Die Prozesse schildern den Ablauf in der Analyse, der nicht mit dem Entwurf übereinstimmen muß und häufig auch nicht sollte. Versuchen Sie bei der Umsetzung insbesondere die noch vorhandene Redundanz zu eliminieren und die Semantik Ihrer Operationen durch Vor- und Nachbedingungen, also nicht operational, zu beschreiben.

- ❑ Wir verwenden an vielen Stellen die Punktnotation, wie sie aus Verbunddatentypen oder objektorientierten Methoden bekannt ist. Sie sollten mit dieser Schreibweise keine grundsätzlichen Probleme haben.
- ❑ Wird in Ausdrücken der Bezeichner „UNDEF“ verwendet, so bedeutet dies, daß der zugehörige Attributwert undefiniert, also (noch) nicht gesetzt ist.
- ❑ Die Operatoren „+=“, „-=“ und „*=“ stellen eine besondere Art der Zuweisung dar. Der Variablen auf der linken Seite wird ihr alter Wert addiert bzw. subtrahiert um oder multipliziert mit dem Wert des Ausdrucks auf der rechten Seite zugewiesen.
- ❑ Aus Platzgründen ersetzen an Stellen, wo keine Mißverständnisse zu befürchten sind, Einrückungen die END's bedingter Anweisungen und Schleifen.

PROZESS-SPEZIFIKATION *Datumsverwaltung (D2.1)*

Eingaben: Datum (*Uhr*)
 Erlaubnis zum automatischen Betrieb(*vom Prozeß D1*)
 Ausgaben: Datum (* an die Prozesse D2.2 - D2.4*)
 Ende des automatischen Betriebs (*an Prozeß D1*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

Vergleich des „Systemdatums“ mit dem aktuellen Datum;

WHILE „Systemdatum“ < aktuelles Datum **DO**

„Systemdatum“ := „Systemdatum“ + 1;

rufe die Prozesse 2.2 - 2.4 auf

END (*WHILE*)

END

PROZESS-SPEZIFIKATION *Systemverwaltung (D4)*

Eingaben: Systemverwaltungsdaten (*vomSystemverwalter*)
Systemdaten (*vom Dateisystem*)
Erlaubnis zur Systemverwaltung (*vom Prozeß D1, Betriebsarten*)

Ausgaben: Dateidaten (*an das Dateisystem*)
Ende der Systemverwaltung (*an Prozeß D1, Betriebsarten*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

*Durchführung von Systemsicherungen und
Laden eines gesicherten Zustandes*

END**PROZESS-SPEZIFIKATION** *betroffene Hilfskräfte ermitteln (D2.2.1)*

Eingaben: Datum (*von Prozeß D2.1*)

Ausgaben: Hilfskraft + Datum + Weihnachtsgeldkondition (*an Prozeß D2.2*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

Ermittle die am Datum gültige Weihnachtsgeldkondition WK;

IF (WK.Stichtag.Tag = Datum.Tag) and (WK.Stichtag.Monat = Datum.Monat)

THEN

FOR alle Hilfskräfte H **DO**

IF es existiert ein Vertrag V mit (V.PersonenID = H.PersonenID) and

(WK.Stichtag IN [V.GültigVon, V.GültigBis]) **THEN**

WeihnachtsgeldBuchen (H, WK, Datum)

END (*IF*)

END (*FOR*)

END (*IF*)

END

PROZESS-SPEZIFIKATION *Weihnachtsgeld buchen (D2.2.2)*

Eingaben: Hilfskraft H, Weihnachtsgeldkondition WK, Datum (*von D2.2.1*)

Ausgaben: Druckauftrag (* an D2.8*)

Vorbedingungen: ...**Nachbedingungen:** ...**BEGIN**

Monate := Anzahl Monate, die die Hilfskraft in Datum.Jahr beschaefigt war,
angefangene Monate werden voll gezaehlt;

IF Monate > 5 **THEN**

(* Geleistete Stunden im Bemessungsmonat ermitteln*)

GeleisteteStunden := 0;

FOR alle Wochen W **IN** WK.Bemessungsmonat **DO****FOR** alle Vertraege V mit V.PersonenID = H.PersonenID **DO****IF** W **IN** [V.GueltigVon, V.GueltigBis] **THEN**

GeleisteteStunden+ := V.Stundenzahl;

TK := Tarifkonditionen.get (Datum); (* Am Berechnungstag geltende Tarifkondition *)

Betrag := TK.Stundensatz*GeleisteteStunden; (*Monatsgehalt*)

Betrag := Betrag * Monate / 12; (*Weihnachtsgeld*)

Stunden := Summe Stundenzahl aller am Stichtag geltenden Vertraege der Hilfskraft ;

FOR alle Vertraege V mit V.PersonenID = H.PersonenID **DO**

(* Alle Kostenstellen der am Stichtag laufenden Vertraege anteilig belasten*)

IF WK.Stichtag **IN** [V.GueltigVon, V.GueltigBis] **THEN**

(* Vertrag laeuft am Stichtag*)

Anteil := Betrag * V.Stundenzahl / Stunden;

B := Buchung.create (V.KostenstellenID, V.VertragsID, Anteil, Datum, "WG");

Buchungen.add (B);

Druckauftrag (V.VertragsID, V.KostenstellenID, Anteil, Datum, "WG")

END (*IF*)**END**

PROZESS-SPEZIFIKATION *betroffene Lehrgebiete ermitteln (D2.3.1)*

Eingaben: Datum (*von Prozeß D2.1*)

Ausgaben: Lehrgebiet + Datum (*an Prozeß D2.3.2*)

Vorbedingungen: ...**Nachbedingungen:** ...**BEGIN****IF** (Datum.Tag = 01) and (Datum.Monat = 01) **THEN****FOR** alle Lehrgebiete L **DO****IF** L.Jahr = Datum.Jahr - 1 **THEN**

Lehrgebiet liquidieren (L, Datum)

END (*IF*)**END** (*FOR*)**END** (*IF*)**END****PROZESS-SPEZIFIKATION** *Lehrgebiet liquidieren (D2.3.2)*

Eingaben: Lehrgebiet + Datum (*von Prozeß D2.3.1*)

Ausgaben: Druckauftrag (*an Prozeß D2.5*)

Vorbedingungen: Lehrgebiet nicht liquidiert**Nachbedingungen:** ...**BEGIN**

Betrag := Lehrgebiet.Kontingente * Lehrgebiet.Äquivalent;

FOR alle Buchungen B mit B.KostenstellenID = Lehrgebiet.KostenstellenID **DO****IF** B ist nicht storniert **THEN**

Betrag := Betrag - B.Betrag

END; (*IF*)**END;** (*FOR*)**IF** Betrag <> 0 **THEN**

B := Buchung.create (Lehrgebiet.KostenstellenID, UNDEF, -Betrag, Datum, "Abschluss");

Buchungen.add (B)

END; (*IF*)

Lehrgebiet.LiquidiertAm := Datum;

automatische Druckaufbereitung (Lehrgebiet, Betrag)

END

PROZESS-SPEZIFIKATION *betroffene Kurse ermitteln (D2.4.1)*

Eingaben: Datum (*von Prozeß D2.1*)
Ausgaben: Kurs + Datum (*an Prozeß D2.4.2*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

FOR alle Kurse K **DO**

IF K.Ende = Datum **THEN**
Kurs sperren (K, Datum)

END (*IF*)

END

PROZESS-SPEZIFIKATION *Kurs sperren (D2.4.2)*

Eingaben: Kurs + Datum (*von Prozeß D2.6.1*)
Ausgaben: Druckauftrag (*an Prozeß D2.5*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

IF Kurs nicht gesperrt **THEN**
Kurs.GesperrtAm := Datum;
automatische Druckaufbereitung (Kurs)

END (*IF*)

END

PROZESS-SPEZIFIKATION *automatische Druckaufbereitung (D2.5)*

Eingaben: zu druckende Daten (*von den Prozessen D2.2 - D2.4 des
automatischen Betriebs*)
Ausgaben: zum Druck aufbereitete Ergebnisse des automatischen Betriebs
(* an den Drucker*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

gib alle Daten Zeilenweise in eine Druckdatei ERG.txt aus

END

PROZESS-SPEZIFIKATION *Kostenstellendaten inspizieren (D3.2.1*)*

Eingaben: Kostenstelle KS (*vom Personalsachbearbeiter*)
 Kostenstellenverwaltungsdaten KV (*vom Personalsachbearbeiter*)
 Buchungsdaten B (*vom Personalsachbearbeiter*)
 Erlaubnis zum Bedienerbetrieb (*vom Prozeß D1*)
 Datum (* vom Prozeß D3.1*)

Ausgaben: Kostenstellendaten (*an Prozeß D3.2.2*)
 Kostenstelle + Änderungen (*an Prozeß D3.2.3*)
 Kostenstelle + Datum (*an Prozeß D3.2.4*)
 (*je nach Anfrageart*)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

IF Erlaubnis zum Bedienerbetrieb **THEN**

IF KV.Befehl = NeueKostenstelleEinrichten **THEN**

 OK := TRUE;

FOR alle Kostenstellen K **DO**

IF (K.KostenstellenID = KV.KostenstellenID) **THEN**

 OK := FALSE

END (*DO*);

IF OK **THEN**

 KostenstelleEinrichten (KV.daten, KV.KostenstellenID) (*Prozeß D3.2.2*)

ELSE

 (*Kostenstelle existiert bereits*)

ELSIF KV.Befehl = "Buchung" **OR** KV.Befehl = "Aendern" **THEN**

IF (KS existiert und KS ist nicht liquidiert) **THEN**

 ÄnderungsartBestimmen (KS, KV, B) (* Prozeß D3.2.3.1*)

ELSE

 (* Kostenstelle existiert noch nicht oder ist liquidiert *)

ELSIF KV.Befehl = KostenstelleLiquidieren **THEN**

 OK := FALSE;

IF KS existiert **THEN**

 KostenstelleLiquidieren (KS, Datum) (* Prozeß D3.2.4 *)

ELSE

 (* Kostenstelle existiert nicht*)

END (*IF Erlaubnis*)

END

PROZESS-SPEZIFIKATION *Kostenstelle einrichten* (D3.2.2)

Eingaben: Kostenstellenverwaltungsdaten KD,
 KostenstellenID KID (* vom Prozeß D3.2.1*)
 Datum Heute

Ausgaben: (*keine, es wird nur auf Lagern gearbeitet*)

Vorbedingungen: ...**Nachbedingungen:** ...**BEGIN****IF** KD.Entitätstyp = Lehrgebiet **THEN**

 K := Lehrgebiet.Create;
 K.KostenstellenID := KID;
 K.EingerichtetAm := Heute;
 Initialisiere K mit dem Kostenstellendatensatz KD;
 Lehrgebiete.add (K)

ELSE

 (*KD.Entitätstyp = Kurs *)
 (* ... *)

PROZESS-SPEZIFIKATION *Änderungsart bestimmen*(D3.2.3.1)

Eingaben: Kostenstelle K (* vom Prozeß D3.2.1*)
 Kostenstellenverwaltungsdaten KV (* vom Prozeß D3.2.1*)
 Buchungsdaten BU (* vom Prozeß D3.2.1*)

Ausgaben: Kostenstelle + Buchungsdaten (* an Prozeß D3.2.3.2*)
 Lehrgebiet + spezielleÄnderung (*an Prozeß D3.2.3.3)
 Kurs + spezielleÄnderung (*an Prozeß D3.2.3.4*)

Vorbedingungen: K existiert**Nachbedingungen:** ...**BEGIN****IF** KV.befehl = "Buchung" **THEN**

 Buchen (K, BU) (* Prozeß D3.2.3.2*)

ELSE**IF** K.Entitätstyp = Lehrgebiet **THEN**

 LehrgebietÄndern (K, KV)

ELSE (*K.Entitätstyp = Kurs*) (* Prozeß D3.2.3.3*)

 KursÄndern (K, KV) (* Prozeß D3.2.3.4*)

END (*IF*)**END** (*IF*)**END**

PROZESS-SPEZIFIKATION *Buchen (D3.2.3.2)*

Eingaben: Kostenstelle K + Buchungsdaten BD (*von Prozeß D3.2.3.1*)
Ausgaben: (*keine; es werden Veränderungen auf Lagern veranlasst*)

Vorbedingungen: K existiert

Nachbedingungen: ...

BEGIN

B := Buchung.create
Rechnungsdaten von BD übernehmen
B.KostenstellenID := K.KostenstellenID;
Buchungen.add (B) (* Buchung in das Lager einfüegen*)

END**PROZESS-SPEZIFIKATION** *Lehrgebiet ändern (D3.2.3.3) [D3.2.3.4]*

Eingaben: Kostenstelle + Änderung (*von Prozeß D3.2.3.1*)
Ausgaben: (*keine; es werden Veränderungen auf Lagern veranlasst*)

Vorbedingungen: Kostenstelle ist Lehrgebiet

Nachbedingungen: ...

BEGIN

Änderung vornehmen durch Überschreiben der Attribute
(*nicht geändert werden dürfen die Attribute EingerichtetAm, LiquidiertAm, Name
und Jahr sowie die ID*)

END

(* Die Spezifikation des Prozesses Kurs ändern (D3.2.3.4) ist analog zu dieser; nicht geändert werden dürfen neben Liquidations- und Einrichtungsdatum die Attribute, die die Kostenstelle eindeutig identifizieren, z.B. die Kursnummer sowie natürlich die ID*)

PROZESS-SPEZIFIKATION *Kostenstelle liquidieren* (D3.2.4)

Eingaben: Kostenstelle K + Datum Heute (* vom Prozeß D3.2.1*)

Ausgaben: keine

Vorbedingungen: K IN Kurse**Nachbedingungen:****BEGIN****IF** K.Entitätstyp = Kurs **THEN****IF** K.LiquidiertAm = UNDEF **THEN**OK := (laufendeVertraege = {}) **AND** (offeneVertraege = {}) **AND**
(K.KlausurenHK <> UNDEF) **AND** (K.KlausurenNK <> UNDEF) **AND**
(K.Einsendungen <> UNDEF);**IF** OK **THEN**

K.LiquidiertAm := Heute

ELSE (*Liquidation nicht möglich*)**END** (*IF*)**ELSE** (*Kostenstelle ist bereits liquidiert, keine Aktion erforderlich*)**END**(*IF*);**END****PROZESS-SPEZIFIKATION** *Informationsretrieval*(D3.4)

Eingaben: Anfrage (* vom Sachbearbeiter bzw. Beauftragten*)

Erlaubnis zum Bedienerbetrieb (* vom Prozeß D1*)

Ausgaben: Anfrageergebnis (*an Beauftragten bzw. Sachbearbeiter*)

Anfrageergebnis (*an D3.8*)

Vorbedingungen: ...**Nachbedingungen:** ...**BEGIN****IF** Erlaubnis zum Bedienerbetrieb **THEN**

(*Vergleiche Produktskizze*)

Druckaufbereitung (Anfrageergebnis)

END (*IF*)**END**

PROZESS-SPEZIFIKATION *Tarifkondition geprüft einfügen (D3.3.3)*
 (* Weihnachtsgeldkondition geprüft einfügen (D3.3.1) und
 Vertragskondition geprüft einfügen (D3.3.2) werden
 analog spezifiziert *)

Eingaben: Tarifkonditionsdaten bzw. Weihnachtsgeldkonditionsdaten bzw.
 Vertragskonditionsdaten (*vom Personalsachbearbeiter*)

Ausgaben: (*keine; es werden nur Änderungen auf Lagern vorgenommen*)

Vorbedingungen: Tarifkonditionsdaten.GueltigVon > Heute

Nachbedingungen: Tarifkonditionen.GueltigeKondition (GueltigVon) = K =>
 K.GueltigBis = UNDEF **AND**
 K.Tarifkonditionsdaten = Tarifkonditionsdaten

BEGIN

TK := Tarifkondition.create (GueltigVon);
 TK die Tarifkonditionsdaten zuweisen;
 (* Alle späteren Konditionen loeschen *)

FOR alle K in Tarifkonditionen mit TK.GueltigVon <= K.GueltigVon **DO**
 Tarifkonditionen.delete (K);
 (* neues Ende der letzten Kondition setzen *)

K := letzte Tarifkondition;
 K.GueltigBis := TK.GueltigVon - 1 Tag;
 Tarifkonditionen.add (TK);

FOR alle laufenden oder offenen Verträge V mit V.GueltigBis >= TK.GueltigVon **DO**
 KS := Kostenstellen.get (V.KostenstellenID);
IF NOT KS.Gesperrt **AND NOT** KS.Liquidiert **THEN**
 storniere Buchung B mit B.VertragID = V.VertragID **AND** B.Art = "Gehalt";
 (* VertragBuchen *)
 (* Dazu gueltige Kondition(en) ermitteln und Buchungsbetrag berechnen *)
 Betrag := 0;
FOR alle K in Tarifkonditionen mit K.GueltigVon <= V.GueltigBis **AND**
 (V.GueltigVon <= K.GueltigBis **OR** K.GueltigBis = UNDEF) **DO**
 Anfang := MAX(K.GueltigVon, V.GueltigVon);
 Ende := MIN(K.GueltigBis, V.GueltigBis);
 Wochen := (Ende-Anfang).Wochen;
 Betrag += K.Stundensatz*V.Stundenzahl*Wochen;
 B := Buchung.createVertragsBuchung (Betrag, V);
 Buchungen.add (B);
IF KS ist überzogen **THEN**
 Vertrag und überzogene Kostenstelle melden

ELSE
 Vertrag und Gesperrte / Liquidierte Kostenstelle melden

END

PROZESS-SPEZIFIKATION *Vertragsdaten pruefen (D3.6.1.1)*

Eingaben: HilfskraftID HKID,
 KostenstellenID KID,
 Erlaubnis zum Bedienerbetrieb (* vom Prozeß D1*)
 Vertragsdaten VD (* vom Hilfskraft-Beauftragten *)
 Datum Heute (* von Prozeß D3.1 *)
 Anzahl E (*von Prozeß D3.6.1.3*)
 Anzahl K (*von Prozeß D3.6.1.4*)

Ausgaben: HKID,
 KID,
 Vertragsdaten VD (* an Prozeß D3.6.1.2 *)
 Klausurdaten VD (* an Prozeß D3.6.1.3 *)
 Einsendungsdaten VD (* an Prozeß D3.6.1.4 *)

Vorbedingungen: HK IN Hilfskraefte AND K IN Kostenstellen

Nachbedingungen: ...

BEGIN

```

IF Erlaubnis zum Bedienerbetrieb THEN
  K := Kostenstellen.get (KID);
IF VD.GueltigBis < Heute THEN
  (* Fehler: Vertrag schon abgelaufen *)
ELSIF VD.GueltigVon.Jahr > Heute.Jahr THEN
  (* Fehler: Vertrag beginnt nicht im laufenden Jahr *)
ELSIF K.GesperrtAm <> UNDEF OR K.LiquidiertAm <> UNDEF THEN
  (* Fehler: Kostenstelle gesperrt oder liquidiert *)
ELSE
  Summe := 0;
FOR alle B in Buchungen mit B.StorniertAm = UNDEF AND
  B.KostenstellenID = K.ID DO
  Summe += B.Betrag;
  (* Gueltige Kondition(en) ermitteln und Buchungsbetrag berechnen *)
  Betrag := 0;
FOR alle TK in Tarifkonditionen mit TK.GueltigVon <= VD.GueltigBis AND
  (VD.GueltigVon <= TK.GueltigBis OR TK.GueltigBis = UNDEF) DO
  Anfang := MAX(TK.GueltigVon, VD.GueltigVon);
  Ende := MIN(TK.GueltigBis, VD.GueltigBis);
  Wochen := (Ende-Anfang).Wochen;
  Betrag += TK.Stundensatz * VD.Stundenzahl * Wochen;
END (*FOR*);
  Summe + := Betrag;
  WGK := Weihnachtsgeldkonditionen.get(VD.GueltigVon);
  (*Etats der durch evtl. erhoehtes Weihnachtsgeld betroffenen
  Kostenstellen pruefen;
  das Weihnachtsgeld wird wie im Automatischen Betrieb berechnet, allerdings mit
  den aktuell bekannten Daten*)
IF WGK.Stichtag IN [VD.GueltigVon, VD.GueltigBis] THEN
  Summe += Anteil des geschaeetzten Weihnachtsgeldes;
  (* Saldo der Kostenstelle pruefen *)
CASE K.Entitaetstyp OF

```

```

"Lehrgebiet" :
  OK := K.Kontingente * K.Aequivalent >= Summe;
"Kurs" :
  E := 0;
  KE := 1;
  (* Letzten Kurs holen *)
  FOR alle K1 in Kostenstellen mit K1.Entitätstyp = "Kurs" AND
    K1.gesperrtAm <> UNDEF AND
    K.Nummer = K1.Nummer DO
    IF AK = UNDEF THEN
      AK := K1;
    ELSIF AK.Jahr < K1.Jahr THEN (* AK abgeschlossen *)
      AK := K1;
    IF AK = UNDEF THEN
      AK := K;
    WHILE (K.Einsendungen[KE] <> UNDEF) AND ( KE <= K.MaxKE) DO
      E+ := K.Einsendungen[KE];
      KE+ := 1
    E := EinsendungenHochrechnen (AK, K.Beleger, E, KE-1, K.maxKE);
    HauptK := KlausurenHochrechnen ('HK', K.Beleger, K.AnmelderHK, AK);
    NachK := KlausurenHochrechnen ('NK', K.Beleger, K.AnmelderNK, AK);
    Stundensatz := Tarifkonditionen.get (VD.GueltigVon)Stundensatz;
    OK = (Stundensatz * 4,348 * E * SWS * K.FaktorE * K.StundenE +
      (HauptK + NachK) * K.FaktorK * K.StundenK) >= Summe;
  END (*CASE*);
  IF NOT OK THEN
    (* Fehler: Kostenstellenetat ueberzogen *)
  ELSE
    (* Hilfskraft nach Vertragskonditionen pruefen *)
    Stunden := VD.Stundenzahl;
    Monate := Anzahl Monate, in denen die Hilfskraft einen Vertrag (der Art
      VD.Art) hatte (incl. dem neuen Vertrag);
    FOR alle V in Vertraege mit V.PersonenID = VD.PersonenID DO
      tempMaxStunden := VD.Stundenzahl;
      FOR alle VI in Vertraege mit VI.PersonenID = VD.PersonenID AND
        V.GueltigVon ∈ [VI.GueltigVon, VI.GueltigBis] DO
        tempMaxStunden := tempMaxStunden + VI.Stundenzahl
      END (*FOR*)
      Stunden := Max(Stunden, tempMaxStunden)
    END (*FOR*)
    OK := TRUE;
    FOR alle VK in Vertragskonditionen mit VK.GueltigVon <= V.GueltigBis AND
      (V.GueltigVon <= VK.GueltigBis OR VK.GueltigBis = UNDEF) DO
      OK := Stunden <= VK.Stundenobergrenze AND
        Monate <= VK.Beschaefigungshoehstdauer
    IF OK THEN
      Vertrag erzeugen und buchen (HKID, KID, VD); (* D3.6.1.2 *)
    ELSE
      (* Fehler: Vertragskondition verletzt *)
    END (* IF Erlaubnis zum Bedienerbetrieb *)
  END
END

```

PROZESS-SPEZIFIKATION *Vertrag erzeugen und buchen (D3.6.1.2)*

Eingaben: HilfskraftID HID, KostenstellenID KID,
Vertragsdaten VD (* von Prozeß D3.6.1.1 *)

Ausgaben:

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

V := Vertrag.create(HID,KID);

V.daten := VD;

Vertraege.add (V);

(* Gueltige Kondition(en) ermitteln und Buchungsbetrag berechnen *)

FOR alle TK in Tarifkonditionen mit TK.GueltigVon <= V.GueltigBis **AND**
(V.GueltigVon <= TK.GueltigBis **OR** TK.GueltigBis = UNDEF) **DO**

Anfang := MAX(TK.GueltigVon, V.GueltigVon);

Ende := MIN(TK.GueltigBis, V.GueltigBis);

Wochen := (Ende-Anfang).Wochen;

Betrag += TK.Stundensatz*V.Stundenzahl*Wochen;

B := Buchung.create (V.VertragsID, V.KostenstellenID, Betrag, „Gehalt“);

Buchungen.add (B);

END

PROZESS-SPEZIFIKATION *Stornierung prüfen (D3.6.2.1)*

Eingaben: VertragsID VID, (* vom Hilfskraft-Beauftragten*)
Datum Heute (* von Prozeß D3.1 *)

Ausgaben: Vertrag, Datum (*an Prozeß D3.6.2.3 *)

Vorbedingungen: V IN Vertraege

Nachbedingungen: ...

BEGIN

V := Verträge.get (VID)

K := Kostenstellen.get (V.KostenstellenID);

IF V.GueltigBis < Heute **THEN**

(* Meldung: Vertrag schon abgelaufen *)

ELSIF K.GesperrtAm <> UNDEF **OR** K.Liquidiert <> UNDEF **THEN**

(* Meldung: Kostenstelle gesperrt oder liquidiert *)

ELSE

Vertrag stornieren (V, Heute); (* Prozeß D3.6.2.3 *)

END

PROZESS-SPEZIFIKATION *EinsendungenHochrechnen (D3.6.1.3)*

Eingaben: Einsendungsdaten ED (* von Prozeß (D3.6.1.1) *)

Ausgaben: Anzahl Einsendungen E (*an Prozeß (D3.6.1.1) *)

BEGIN

E := 0;

K := ED.Kurs;

FOR KE := 1 TO ED.AktuelleKE **DO**

E += K.Einsendungen[KE];

IF ED.AktuelleKE = ED.LetzteKE **THEN**

Faktor := (ED.Beleger / K.Beleger)

ELSE

Faktor := (ED.Einsender / E); (* UNDEF, wenn E=0!*)

FOR KE := ED.AktuelleKE + 1 TO ED.LetzteKE **DO**

E += K.Einsendungen[KE]

RETURN E * Faktor**END****PROZESS-SPEZIFIKATION** *KlausurenHochrechnen (D3.6.1.4)*

Eingaben: Klausurdaten (* von Prozeß (D3.6.1.1) *)

Ausgaben: Anzahl Klausuren K (*an Prozeß (D3.6.1.1) *)

BEGIN

K := Klausurdaten.Kurs;

IF Klausurdaten.KlausurArt = 'HK' **THEN** **IF** Klausurdaten.AnmelderHK = UNDEF **THEN** **RETURN** K.KlausurenHK * (Klausurdaten.Beleger / K.Beleger) **ELSE** **RETURN** K.KlausurenHK * (Klausurdaten.AnmelderHK / K.AnmelderHK)**ELSIF** Klausurdaten.AnmelderNK = UNDEF **THEN** **RETURN** K.KlausurenNK * (Klausurdaten.Beleger / K.Beleger) **ELSE** **RETURN** K.KlausurenNK * (Klausurdaten.AnmelderNK / K.AnmelderNK)**END**

PROZESS-SPEZIFIKATION *Vertragsende pruefen (D3.6.2.2)*

Eingaben: VertragsID VID, Datum EndeNeu(* vom Hilfskraft-Beauftragten *)
 Datum Heute (* von Prozeß D3.1 *)
 Ausgaben: Vertrag (* an Prozeß D3.6.2.4 *)

Vorbedingungen: V IN Vertraege **AND** EndeNeu >= V.GueltigVon **AND**
 EndeNeu < V.GueltigBis

Nachbedingungen: ...

BEGIN

V := Verträge.get(VID);

K := Kostenstellen.get (V.KostenstellenID);

IF V.GueltigBis < Heute **THEN**

(Fehler: Vertrag schon abgelaufen *)*

ELSIF K.GesperrtAm <> UNDEF **OR**

K.LiquidiertAm <> UNDEF **THEN**

(Fehler: Kostenstelle gesperrt oder liquidiert *)*

ELSE

Vertrag stornieren (V, Heute); *(* Prozeß D3.6.2.3 *)*

IF V.GueltigVon > Heute **THEN**

(Vertrag noch nicht laufend *)*

V.GueltigBis := EndeNeu;

V.StorniertAm := UNDEF

VertragUngeprueftBuchen (V)

ELSIF V.GueltigVon < EndeNeu < V.GueltigBis **THEN**

(Vertrag laufend *)*

NeuVertrag := V.Duplicate;

NeuVertrag.StorniertAm := UNDEF;

NeuVertrag.GueltigBis := EndeNeu

VertragUngeprueftBuchen (NeuVertrag); *(* D3.6.2.4 *)*

Vertraege.add (NeuVertrag);

ELSE

(Fehler: Vertraege nicht verlaengerbar *)*

END

PROZESS-SPEZIFIKATION *Vertrag stornieren (D3.6.2.3)*

Eingaben: Vertrag V (* vom Prozeß D3.6.2.1 *)
Datum Heute (* von Prozeß D3.1 *)

Ausgaben: keine

Vorbedingungen: V IN Vertraege

Nachbedingungen: ...

BEGIN

(* Buchungen stornieren *)

FOR alle Buchungen B mit B.VertragID = V.VertragID AND B.Art = "Gehalt" **DO**

B.StorniertAm := Heute;

END; (*FOR*)

V.StorniertAm := Heute;

END

PROZESS-SPEZIFIKATION *Vertrag ungeprüft buchen (D3.6.2.4)*

Eingaben: Vertrag V (* von Prozeß D3.6.2.2 *)

Ausgaben: Vertrag V (*an Prozeß D3.6.1.3 *)

Vorbedingungen: ...

Nachbedingungen: ...

BEGIN

(* Gueltige Kondition(en) ermitteln und Buchungsbetrag berechnen *)

FOR alle TK in Tarifkonditionen mit TK.GueltigVon <= V.GueltigBis **AND**

(V.GueltigVon <= TK.GueltigBis **OR** TK.GueltigBis = UNDEF) **DO**

Anfang := MAX(TK.GueltigVon, V.GueltigVon);

Ende := MIN(TK.GueltigBis, V.GueltigBis);

Wochen := (Ende-Anfang).Wochen;

Betrag += TK.Stundensatz*V.Stundenzahl*Wochen;

B := Buchung.create (V.VertragsID, V.KostenstellenID, Betrag, "Gehalt");

Buchungen.add (B);

END

3.5 Das Datenlexikon

Zum Abschluß der Modellierung präsentieren wir nun das Datenlexikon. Daten, deren Typ klar ist (z.B. Name, Geburtsdatum) haben wir nicht extra aufgeführt. Ebenso sind Typen, die speziell zur Kommunikation zwischen Prozessen dienen (z.B. „Kostenstellenverwaltungsdaten“) und die nach einem bestimmten Muster aus einem Grundtyp (z.B. „Kostenstelle“) und einem Befehl (z.B. „Ändern“) zusammengesetzt werden, nicht aufgeführt.

*ID	= <i>cardinal</i> **Alle ID's
*Änderung	= **Angabe einzelner Datenfelder, die geändert werden sollen
*Daten	= ** wenn nicht anders angegeben die Datenfelder von * ohne dessen ID
*Verwaltungsdaten	= ** Befehl + zugehörige Informationen zum Aufnehmen, Ändern und Löschen eines Datensatzes
Abschluß	= <i>string</i> ** Auflistung aller Schul- und Hochschulabschlüsse
Adresse	= <i>Straße + Postleitzahl + Stadt + Land + Telefon + FAX + eMail</i>
Anfrage	= ** <i>Informationsanfrage</i>
Äquivalent	= <i>cardinal</i> ** Höhe eines Kontingents für den Lehrgebietsetat
AltesJahr	= <i>Datum</i>
NeuesJahr	= <i>Datum</i>
AktuelleKE	= <i>cardinal</i>
AnmelderHK	= <i>cardinal</i>
AnmelderNK	= <i>cardinal</i> ** Zahl der Anmeldungen zur Haupt- bzw. Nachklausur
Anspruch	= <i>cardinal</i> ** Urlaubsanspruch resultierend aus neuen Verträgen
Anzahl	= <i>cardinal</i>
Beginn	= <i>Datum</i>
Begründung	= <i>string</i>
Beleger	= <i>cardinal</i>
Bemerkungen	= <i>string</i>
Beschäftigungshöchstdauer	= <i>cardinal</i> ** Maximale Anzahl Vertragsmonate einer Hilfskraft
Buchung	= <i>@BuchungsID + KostenstellenID + VertragsID + Betrag + GebuchtAm + StorniertAm + BuchungsArt</i>
BuchungsArt	= <i>"Gehalt" "WG" "Abschluss" "Sonst"</i>
Dateidaten	= ** Alle Daten des aktuellen Systemzustandes
Datum	= <i>Tag + Monat + Jahr</i>
Dienstadresse	= <i>Adresse + Firma + Gebäudeteil + Zimmer</i>
Druckauftrag	= ** beliebige Daten, die ausgegeben werden sollen
Druckdokument	= ** Druckaufträge und Formatierungsanweisungen
Ende	= <i>Datum</i>
EingerichtetAm	= <i>Datum</i>
Einsender	= <i>cardinal</i>
Einsendungen	= ARRAY [1..MaxKE] OF <i>cardinal</i> ** (Gesamt-)Zahl der Einsendungen pro Kurseinheit, ** werden mit den Daten der letzten Kursdurchführung initialisiert
Einsendungsdaten	= <i>Kurs + Beleger + Einsender + AktuelleKE + LetzteKE</i>
Erstellungsdatum	= <i>Datum</i>
Eternity	= 31.12.9999 ** nicht gesetztes Datum
FaktorE	= <i>cardinal</i> ** Faktoren zur Berechnung der Gelder, die für die Korrektur der
FaktorK	= <i>cardinal</i> ** Einsendungen bzw. der Klausuren zur Verfügung stehen

GebuchtAm	= Datum
GesperrtAm	= Datum
GültigBis	= Datum
GültigVon	= Datum
Hilfskraft	= @PersonenID + Name + Vorname + Geburtsdatum + Geschlecht + Adresse + Dienstadresse + Geburtsort + Staatsangehoerigkeit + Familienstand + Kinder + BankinstitutOrt + Bankleitzahl + Kontonummer + Abschluss + Personalnummer + Bemerkungen + Präsenztage + Interessen + VeruegbarAb + VoraussichtlicheStundenzahl
Interessen	= string
KlausurArt	= "HK" "NK"
Klausurdaten	= Kurs + KlausurArt + Beleger + (AnmelderHK AnmelderNK)
KlausurenHK	= cardinal
KlausurenNK	= cardinal ** Zahl der Teilnehmer an der Haupt- bzw Nachklausur
Kondition	= @Vertragskondition Tarifkondition Weihnachtsgeldkondition + @GueltigVon + @GueltigBis
Kontingente	= cardinal ** Berechnungsfaktor für den Lehrgebietsetat
Kostenstelle	= @KostenstellenID + ** explizit vom Personalsachbearbeiter angegeben EingerichtetAm + GesperrtAm + LiquidiertAm
Kurs	= @Kostenstelle + Nummer + Jahr + Beginn + Ende + SWS + MaxKE + Beleger + Einsendungen + AnmelderHK + AnmelderNK + KlausurenHK + Klausuren NK + FaktorE + Faktor K + StundenE + StundenK
Lehrgebiet	= @Kostenstelle + Name + Jahr + Kontingente + Äquivalent
LetzteKE	= cardinal
LiquidiertAm	= Datum
MaxKE	= cardinal ** Obergrenze der Anzahl der Kurseinheiten
Präsenztage	= {Mo Di Mi Do Fr}(Sa) (So)}
PSInfos	= ** Rückgaben des Infoetrievals
StorniertAm	= Datum
StundenE	= cardinal ** durchschnittliche Zeit zur Korrektur einer Einsendung bzw.
StundenK	= cardinal ** Klausur zur Berechnung des zur Verfügung stehenden Kursetats
Stundenobergrenze	= cardinal ** Maximale Wochenstundenanzahl einer Hilfskraft
Stundensatz	= cardinal ** Bruttostundenlohn einer Hilfskraft in Pfennigen
SWS	= cardinal
Systemdaten	= ** Alle Daten der letzten Sicherung
Tätigkeit	= string
Tarifkondition	= Stundensatz
Verfügbar	= boolean
Vertrag	= @VertragsID + PersonenID + KostenstellenID + Stundenzahl + Tätigkeit + Begründung + GültigVon + GültigBis + Erstellungsdatum + StorniertAm
Vertragskondition	= Beschäftigungshöchstdauer + ** in Monaten Stundenobergrenze ** pro Woche
Weihnachtsgeldkondition	= Bemessungsmonat + Stichtag
Wochenarbeitstage	= cardinal

