Objektorientierte Modellierung mit dem Objekt-Prozeß-Modell

Prof. Dr. I. Philippow, Dr. R. Burkhardt, Dipl.-Inf. M. Wolf
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung, Fachgebiet Prozeßinformatik
Postfach 0565, 98684 Ilmenau
eMail: [Philippow|RSB|MWolf]@TheoInf.TU-Ilmenau.de

Zusammenfassung

Die entwickelte Methode zur objektorientierten Prozeßmodellierung berücksichtigt in besonderem Maße die Abbildung dynamischer Aspekte in objektorientierten Modellen. Auf der Grundlage insbesondere der dynamischen Aspekte der Objekttechnologie werden im zugeordneten methodenspezifischen Modellierungsprozeß die Abläufe im modellierten Informationsbereich in den Vordergrund gerückt. Das Ergebnis der Modellierung, die am Beispiel ausgewählter Geschäftsprozesse einer Universitätsbibliothek erläutert wird, ist ein Objekt-Prozeß-Modell. Der Einsatz von Zeitanalysen und modellbezogenen (statischen) Verfahren, aber auch eine dynamische Modellüberprüfung zum Feststellen der Konformität zwischen Modell und der realen Welt bieten Ansatzpunkte für die Überprüfung des entstehenden Modells.

Schlüsselwörter

Objektorientierte Prozeßmodellierung, Objekt-Prozeß-Modell, Modellverifikation, Objekttechnologie, Bibliotheksmodellierung

Einleitung

Die Modellierung der Vorgänge eines Aufgabenbereichs (z.B. Unternehmen: Geschäftsprozesse) ist nicht nur geeignet zur Optimierung der Organisationsstrukturen, sondern ist auch Ausgangspunkt für das Auslösen von Teilen des Gesamtmodells (z.B. Unternehmensmodell) mit dem Ziel der Entwicklung von Software für einzelne Teilprozesse. Als Modellierungsmethode wird das Objekt-Prozeß-Modell (OPM, [Burkhardt94]) vorgestellt, das auf der Objekttechnologie basiert und die dynamischen Aspekte eines Systems in den Vordergrund stellt. Wichtige Eigenschaft der Modellierungsmethode ist die Überprüfbarkeit der Modelle. Die Überprüfung beim OPM erfolgt zum einen auf der Basis der Petri-Netz-Theorie, zum anderen durch Einsatz verschiedener KI-Methoden. Die Ergebnisse dieser Modellüberprüfung können Ausgangspunkt für Optimierungen sein. Im folgenden berichten wir vom Einsatz der Objektorientierten Prozeßmodellierung bei der Neustrukturierung eines Unternehmens (Universitätsbibliothek) und den dabei gesammelten Erfahrungen.

Modellbildung

Modellierung ist die Erstellung einer idealisierten, vereinfachten, in gewisser Hinsicht ähnlichen Darstellung eines Gegenstands, Systems oder sonstigen Weltausschnitts mit dem Ziel, bestimmte Eigenschaften des Vorbildes besser studieren zu können ([Hesse94]). Ein Modell spiegelt immer nur einen ausgewählten Teil des realen Systems wider. Diese Einschränkung ist notwendig, um sich auf die wesentlichen zu untersuchenden Aspekte konzentrieren zu können. Die Notwendigkeit der Erstellung von Modellen zur abstrakten Beschreibung technischer,

gesellschaftlicher, softwaretechnischer oder ökonomischer Abläufe oder Zustände kann sich aufgrund unterschiedlichster Anforderungen ergeben und verschiedene Ziele verfolgen.

Das Wesen dieser Herangehensweise besteht darin, die Vorstellungen über ein zu beschreibendes System als Modell mit den entsprechenden Beschreibungskomponenten anzugeben. Dabei ist im Rahmen des Modellierungsprozesses eine intensive Zusammenarbeit zwischen der Fachabteilung – in unserem Falle Vertreter des Direktorats der Bibliothek – und den Informationsverarbeitungsexperten unabdingbar.

Werden im Prozeß des Business Process Reengineering ([Pietsch94], [Jacobson94]) Modelle für Geschäftsabläufe erstellt, so können in Auswertung der Analyse der Modelle neue Organisationsstrukturen zutagetreten, die von den bestehenden Strukturen abweichen können, wenn sich diese als ineffektiv, umständlich oder überholt herausstellen. Die Verbesserungen wirken unmittelbar in den kritischen aktuellen Leistungsparametern wie Qualität, Zeit und Kosten.

Modellierungsprozeß

Die objektorientierte Prozeßmodellierung erfolgt nach folgendem Schema (Bild 1).

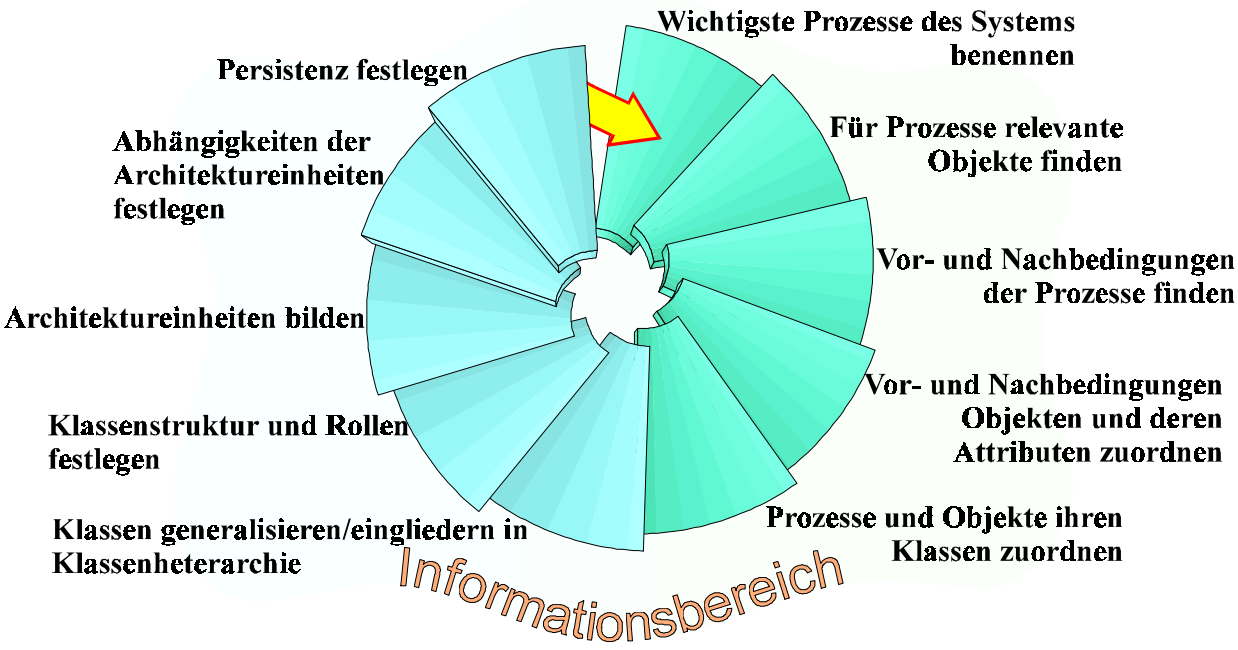

Bild 1: Methodenspezifischer Modellierungsprozeß für das Objekt-Prozeß-Modell

Die ersten fünf der gezeigten Stufen sind dabei der dynamischen Sicht, der zweite Teil eines Modellierungszyklus' der statischen Sicht zuzuordnen. Ausgangspunkt der Modellierung ist das Identifizieren der wesentlichen Geschäftsprozesse im Unternehmen bzw. im aktuell betrachteten Ausschnitt des Unternehmens. Über sie können im nächsten Schritt relevante Objekte gefunden werden. Nach der Angabe von Vor- und Nachbedingungen der Prozesse sind diese den entsprechenden Attributen der gefundenen Objekte zuzuordnen. Den Kontext der Prozesse bilden Objekte. Objektorientiert interpretiert sind Prozesse Instanzen von Methoden und Objekte Instanzen von Klassen. Ausgehend von der Zuordnung zu den Klassen ist beim erstmaligen Modellieren eine Klassenheterarchie zu entwerfen, in die bei weiteren Modellierungstätigkeiten hinzukommende Klassen integriert werden, so daß die Klassenstruktur des Unternehmens entsteht. Genauso wie bei der Angabe der Klassenstruktur und der Rollen zwischen den Klassen handelt es sich hierbei bereits um statische Modellierungsaspekte. Auf Grundlage der bis zu dieser Stufe gewonnenen Informationen kann der dynamische Prozeßablauf überprüft werden; soll weiterhin eine projektbezogene Softwarearchitektur definiert wer-

den, sind zunächst Architektureinheiten zu bilden, um dann deren Abhängigkeiten untereinander (Benutzung) festlegen zu können. Gegebenenfalls kann anschließend objekt- bzw. klassenbezogen die Persistenz angegeben werden.

Das Modell kann, angepaßt an den aktuellen Stand der Wissensakquisition des Entwerfers, kontinuierlich wachsen, indem Stück für Stück weitere Aspekte hinzugefügt oder aber Objekte und Prozesse näher beschrieben werden können. Dazu ist das in Bild 1 angegebene Vorgehen inkrementell zu durchlaufen, wobei die Anzahl der Durchläufe durch die Komplexität und den Grad der Verfeinerung des modellierten Informationsbereich bestimmt wird. Dabei können sowohl neue Aspekte hinzugefügt als auch bestehende überarbeitet und modifiziert werden. Das Ziel solcher Veränderungen ist die Verbesserung des Modells. Im Zuge einer ständigen Modellüberarbeitung und -erweiterung ist eine möglichst häufige Überprüfung der erstellten Modelle sinnvoll.

Das im folgenden beschriebene Objekt-Prozeß-Modell beschränkt sich auf die dynamischen Aspekte im beschriebenen Modellierungsprozeß.

Objekt-Prozeß-Modell

Ein Objekt-Prozeß-Modell wird in einer graphischen Notation dargestellt. Die Modellierung erfolgt anhand von Schemata, die Objekte, Prozesse und Kanten enthalten.

Objekte sind Konzepte, Abstraktionen oder Dinge, die einen Zustand, ein Verhalten und eine Identität besitzen ([Booch94], [Rumbaugh91]). Integriert in ein Softwaresystem ist ein Objekt aus der Sicht des OPM ein zur Laufzeit verfügbarer Datenraum, das einer Klasse zugeordnet ist. Das Objekt erhält kontrollflußrelevante Attribute, die in Anlehnung an Petri-Netze ([Jensen92], [Starke90]) Farben genannt werden, und ist die gefaltete Darstellung eines Objektschemas (Bild 2).

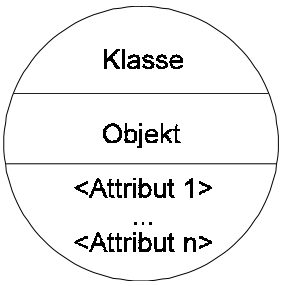

Bild 2: OPM-Objekt

Bild 3: OPM-Prozeß

Bild 3 zeigt die gefaltete Darstellung eines Prozesses. Identifizierend sind der Klassenname, der in der oberen Hälfte des Prozeßsymbols angegeben wird, und der Prozeßname (Methodenname), der in der unteren Hälfte des Prozeßsymbols einzutragen ist. Aus der Sicht der Objekttechnologie ist ein Prozeß eindeutig einer Klasse zugeordnet. Weitere im Kontext dieses Prozesses relevante Prozesse werden in der verfeinerten Darstellung des Prozesses einschließlich ihrer logischen Zusammenhänge angegeben. Für einen Prozeß können Fristen (Zeitraum, den ein Prozeß für die Abarbeitung benötigt), Termine (zeitliche Fälligkeit des Prozesses) und Prioritäten vorgegeben werden. Mit Hilfe von Prioritäten können Konflikte zwischen konkurrierenden Prozessen gelöst werden.

Kanten verbinden jeweils ein Objekt- mit einem Prozeßsymbol, die Kantenbeschriftung gibt eine Bedingung an. Gerichtete Kanten zwischen Objekt und Prozeß beschreiben attributbezogene Voraussetzungen des Objektes, die bei der Aktivierung des Prozesses gelten müssen. Eine Nachbedingung (Kante Prozeß – Objekt) gibt an, welche Werte bei den betreffenden Objekten nach dem Prozeß vorliegen müssen. Für die gültigen Farbtypen sind Operatoren definiert, die innerhalb der Kantenbedingungen verwendet werden können.

Modellüberprüfung

Die angelegten Modelle enthalten dynamische Aspekte des Informationsbereichs. Die Überprüfung dieser dynamischen Aspekte kann statisch erfolgen, d.h. zum Zeitpunkt der Modellüber-

prüfung muß lediglich das Modell existieren. Die dynamische Überprüfung der dynamischen Aspekte erfordert dahingegen neben dem Modell auch die Existenz des dem Modell folgenden Anwendungssystems.

Statische Modellüberprüfung

Die statische Modellüberprüfung nutzt u.a. die zugrundeliegende Petri-Netz-Theorie, wobei die Kriterien Konfliktfreiheit, Lebendigkeit, Erreichbarkeit und Sicherheit überprüfbar sind.

Die Untersuchung der Lebendigkeit kann beispielsweise in einem Unternehmensmodell stagnierende Ablaufsituationen oder blockierende Geschäftsprozesse aufdecken, die Anlaß zur Umgestaltung sein können. Die Überprüfungskomponenten sind in einem Entwurfswerkzeug eingebettet, das u.a. die Ergebnisse der Petri-Netz-Analysen in die Terminologie des Objekt-Prozeß-Modells transformiert.

Die statische Überprüfung der Schemata des Objekt-Prozeß-Modells kann anhand von Regeln geschehen, die gemäß vorgegebener Schablonen vom System generiert werden oder global vorgegeben sind. Die Regeln lassen sich in vier Kategorien einteilen:

- Objekttechnologie-Regeln
  z.B. Instanziierung, Klassifizierung, Komponentenbeziehung
- Schemaregeln
  syntaktische Festlegungen, die die Gültigkeit eines Schemas betreffen
- Komplexe Regeln
  Festlegungen, die schemabezogen sein können und u.a. die Petri-Netz-Kriterien Konfliktfreiheit, Lebendigkeit, Erreichbarkeit und Sicherheit betreffen
- Ästhetik-Regeln
  zur Erhöhung der Lesbarkeit und Verständlichkeit getroffene Forderungen, die über Fuzzy-Variablen eine qualitative Aussage zur Übersichtlichkeit ermöglichen

Dynamische Modellüberprüfung

Dynamische Korrektheit liegt dann vor, wenn die aktuellen Instanzen des OPM und der Anwendung für spezifizierte Anwendungsfälle konform sind. Eine Kopplung zwischen dem Modell und der Anwendungssoftware ermöglicht die Kontrolle der Einhaltung des im Modell spezifizierten Verhaltens während der Ausführung des Anwendungssystems. Ein möglicher Ansatz zur Realisierung einer solchen dynamischen Konsistenzprüfung besteht aus der Anwendung folgender vier Schritte:

- Generieren
  Einfügen eines methodenspezifischen Überprüfungscodes in die Anwendung, Kontrolle struktureller Eigenschaften (Klassen, Methoden)
- Compilieren
  Erzeugung eines ablauffähigen Anwendungscodes mit integrierten Überprüfungsmechanismen. Erkennen von syntaktische Abweichungen, Fehlern, die die Regeln der Objekttechnologie betreffen; inkorrekte Verwendung von Operatoren und Parametern der Farbklassen oder unzulässige Benutzung von Objekten
- Ausführen
  Überwachung der Einhaltung der geforderten Bedingungen (Vor- und Nachbedingungen, Zeitrestriktionen, Invariante, schemaübergreifende Regeln)
- Abgleich
  Feststellung von Inkonsistenzen; Veränderung von Modell oder Anwendungssystem zum Herstellen der Konsistenz mit der Realität

Bibliotheksmodellierung

Dieser Abschnitt soll das Objekt-Prozeß-Modell anhand des Beispieles der Modellierung wichtiger Teilprozesse der Universitätsbibliothek Ilmenau erläutern und Vorgehen sowie Ziel der Modellierung aufzeigen. Untersucht wurden wichtige Geschäftsprozesse in der Bibliothek. Dabei wurde das Gebiet der Beschaffung und Eingliederung von Literatur vertieft, da es Ausgangspunkt jeglicher Bibliotheksarbeit ist.

Die Herangehensweise der primären Betrachtungen von Prozessen ermöglicht die aufgabenbezogene Neuordnung bestehender Organisationsstrukturen. Ausgehend von den für die Prozesse relevanten Objekten, die Personen, Arbeitsgruppenteile oder Organisationseinheiten beschreiben, können aufgrund erkannter Prozeßzuständigkeiten Geschäftsprozesse grundsätzlich überarbeitet werden, so daß durch eine geschickte Umverteilung und die Nutzung vorhandener Ressourcen eine höhere Effizienz der Aufgabenerfüllung erreicht wird. Dabei spielt die Rationalisierung von Arbeitsplätzen nicht die dominierende Rolle. Vielmehr soll der Mensch in seiner Arbeit durch die DV-Systeme unterstützt werden, die aus dem Modell entwickelt werden können.

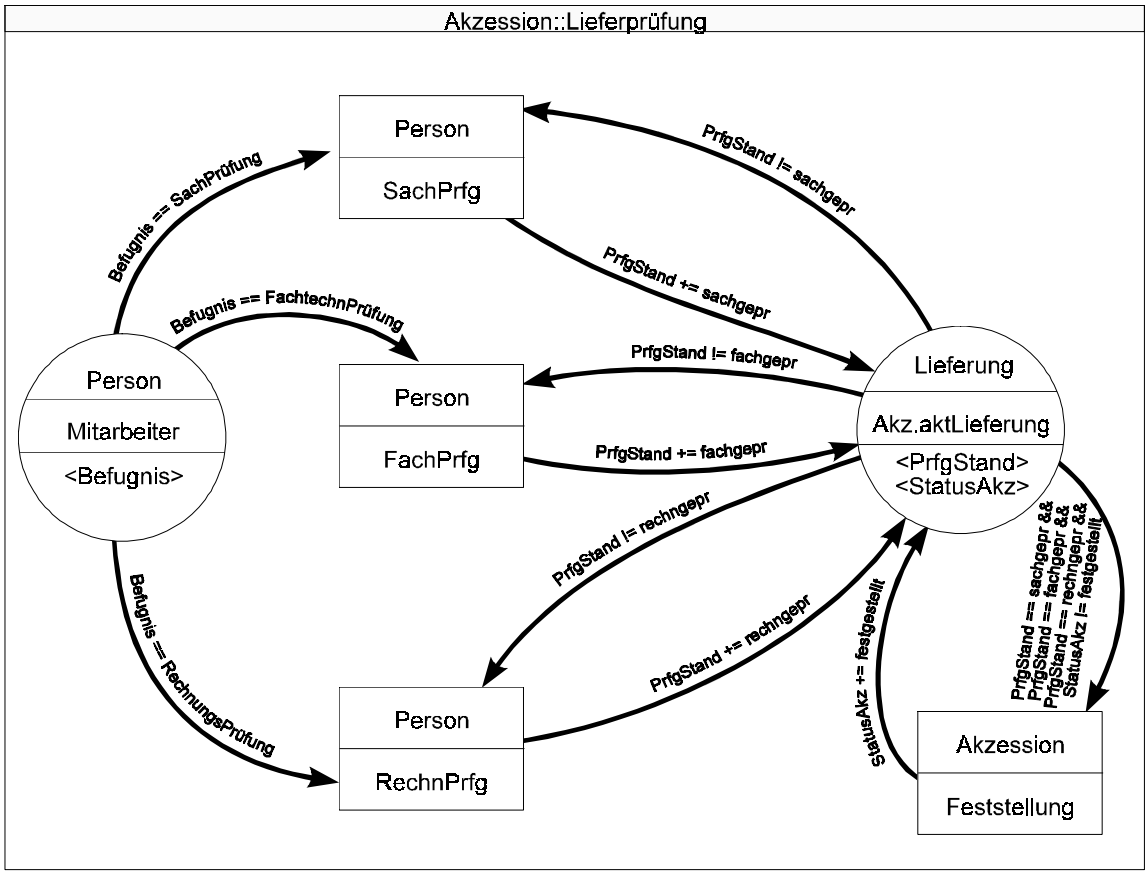

Bild 4: OPM-Schema Lieferprüfung, vereinigt alle notwendigen Prüfprozesse

Die Aufgabe der Bibliothek besteht darin, Forschung und Lehre der Technischen Universität Ilmenau mit aktueller Literatur in ausreichendem Maße zu unterstützen, sowie dem außeruniversitären Umfeld ein breites Angebot von wissenschaftlichen Veröffentlichungen und Nachschlagewerken anzubieten. Die Literaturbeschaffung besteht im wesentlichen aus den Prozessen Bedarfsermittlung/-bearbeitung, Bestellung von Literatur und Erschließung. Eingebettet in den auf die Anlieferung folgenden Bearbeitungsablauf ist die Durchführung der in Bild 4 angegebenen Prozesse zur Lieferprüfung.

Unter Anwendung des Modellierungsprozesses aus Bild 1 soll nun der im Schritt 1 gefundene Prozeß der fachtechnischen Überprüfung (*FachPrfg*) modelliert werden. Im Schritt 2 erfolgt die Identifizierung der relevanten Objekte *Mitarbeiter* und aktuelle Lieferung der Akzession (*Akz.aktLieferung*). Für den Prozeß gilt die Nachbedingung, daß die fachtechnische Korrektheit geprüft ist (*fachgepr*). Als Vorbedingungen wird erkannt, daß die Prüfung nur Befugten erlaubt ist und die Fachprüfung noch nicht erfolgt ist (Schritt 3). Im anschließenden Modellierungsschritt wird das den Bearbeitungszustand der Fachprüfung beschreibende Attribut *PrfgStand* genannt und dem Objekt *Akz.aktLieferung* zugeordnet, die *FachtechnPrüfung* wird als Ausprägung des Attributs *Befugnis* beim Objekt *Mitarbeiter* festgelegt. Die Vorbedingung „*Befugnis == FachtechnPrüfung*“ beschreibt, daß im Attribut *Befugnis* die Belegung *FachtechnPrüfung* gesetzt sein muß, während durch die Bedingung „*PrfgStand == PrfgStand*“ seitens der aktuellen Lieferung die Nichtexistenz des Elements *fachgepr* im Attribut *PrfgStand* gefordert wird. In der Nachbedingung wird spezifiziert, daß im Laufe der Abarbeitung des Prozesses das Element *fachgepr* dem Attribut *PrfgStand* hinzugefügt wurde, dies wird durch den Operator „+=” symbolisiert. Im Schritt 5, dem letzten der dynamischen Seite des Modellierungsprozesses wird die fachtechnische Prüfung (*FachPrfg*) der Klasse *Person* zugeordnet, womit festgelegt wird, daß dieser Prozeß im Kontext eines Objekts dieser Klasse abläuft (Zuordnung *Mitarbeiter* zur Klasse *Person*). Die Klasse von *Akz.aktLieferung* heißt *Lieferung*. Auf die Erläuterung der weiteren Modellierungsschritte (Stufe 6-10) wird hier verzichtet, da die dynamischen Aspekte im Vordergrund der Betrachtung stehen.Die Vervollständigung des Prozeßschemas *Akzession::Lieferprüfung* erfolgt analog, gegebenenfalls durch weitere Modellierungszyklen. Die Klasse *Akzession* wurde bereits auf einer darüberliegenden Verfeine-

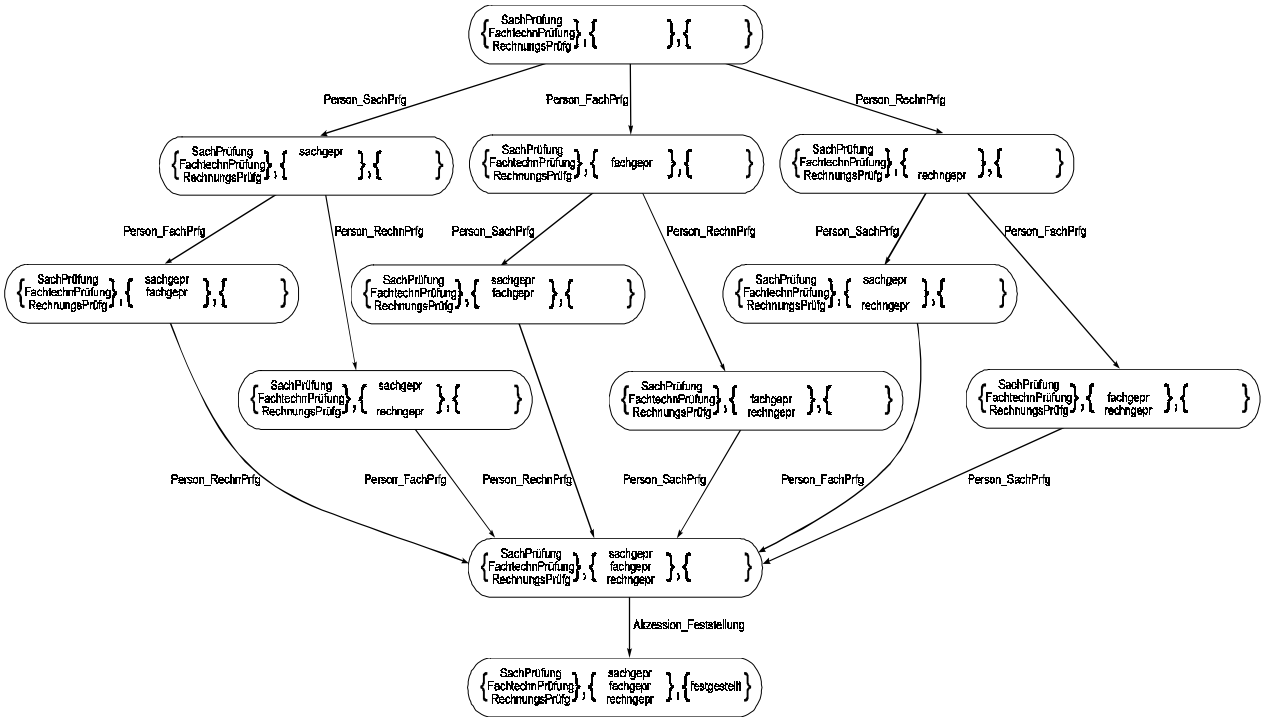

Bild 5: Erreichbarkeitsgraph zum OPM-Schema Lieferprüfung

rungsebene erkannt und die Vor- und Nachbedingungen an dem den Kontext bildenden Prozeß *Akzession::Lieferprüfung* angetragen. Dabei sei darauf hingewiesen, daß der positive bzw. negative Ausgang der Teilprüfungen keine charakteristische Vor- bzw. Nachbedingung des jeweiligen Prozesses (*FachPrfg*, *SachPrfg*, *RechnPrfg*, *Feststellung*) ist. Sind vom Ausgang der Teilprüfungen unterschiedliche Folgeprozesse betroffen, so sind diese in den jeweiligen Prozeßverfeinerungen zu spezifizieren. Ein besonderer Aspekt ist an den Bedingungen des Prozesses *Feststellung* zu erkennen, der erst ablaufen darf, wenn die drei Teilprüfungen durch-

geführt wurden; durch die drei Prozesse *FachPrfg*, *SachPrfg* und *RechnPrfg* werden die Ausprägungen *sachgepr*, *fachgepr* und *rechngepr* im Attribut *PrfgStand* gesetzt, die notwendige Vorbedingungen für die *Feststellung* sind. Es erfolgt implizit eine Sequentialisierung, obwohl die Reihenfolge der Ausführung der Prüfprozesse nicht explizit angegeben ist. Wie in der Realität wird also zunächst von einer parallelen Bearbeitung ausgegangen, bis diese über Vor- und Nachbedingungen durch Synchronisation eingeschränkt wird.

Die Überprüfung der einzelnen Modelle ist auf Schema- oder Systemebene möglich und setzt die oben beschriebenen Kriterien um.

Anwendung von komplexen Regeln

Zur Beurteilung der Petri-Netz-Kriterien Konfliktfreiheit, Lebendigkeit, Erreichbarkeit und Sicherheit kann das OPM-Schema in ein gefärbtes Petri-Netz entfaltet ([Keichel95]) und mit verfügbaren Simulatoren (z.B. [Peneca95]) oder Analysatoren untersucht werden. Zum anderen können auf OPM-Ebene Simulationen und Analysen durchgeführt werden ([Wolf95]). Ausgehend von dem in Bild 4 angegebenen Schema wurde der in Bild 5 angegebene Erreichbarkeitsgraph (in Anlehnung an [Starke90], [Philippow91]) beginnend bei einer Initialmarkierung berechnet, der die Analyse von Erreichbarkeit und Lebendigkeit zuläßt.

An dem angegebenen Graphen zum OPM-Schema wird beispielsweise sichtbar, daß eine vollständige *Lieferprüfung* nur dann möglich ist, wenn die *Befugnis* der Mitarbeiter für alle Teilprüfungen vorliegt. In Auswertung des Modells konnten Vorschläge unterbreitet werden, die umgesetzt in der Bibliothek Prozeßabläufe verbesserten.

Anwendung von Ästhetik-Regeln

Das Ziel der Betrachtung ästhetischer Aspekte ist die Verbesserung der Lesbarkeit und Verständlichkeit, welche die Grundlage für eine gute Dokumentation und einfache Handhabbarkeit sind ([Burkhardt94], [Wolf95]). Wichtigstes Kriterium ist dabei die Übersichtlichkeit der Gestaltung der einzelnen Schemata. Zum Grad der Übersichtlichkeit ist im allgemeinen keine scharfe Aussage möglich, sondern die Bewertung erfolgt über die Festlegung ungefährer Maße, die das Ergebnis einer Fuzzy-Inferenz über unscharfe Aussagen bezüglich des Schemas sind. Das Bewertungskriterium wird z.B. in einer Wissensbasis abgelegt, woraufhin eine Online-Modellierungsunterstützung erfolgen kann. Die Ermittlung der Übersicht erfolgt anhand der Anzahl der Objekte (*AnzahlObjekte*), Prozesse (*AnzahlProzesse*), Kante pro Prozeß (*KantenMass*), maximale und mittlere Anzahl der Bedingungen je Kante (*MaxBeding*, *MittelBeding*). Für alle diese über das Repository zu erfassenden Werte wurden linguistische Variablen definiert. In der Regelmenge (Produktionsregeln), die eine unscharfe qualitative Aussage in der Ausgangsvariable *Uebersicht* mit den Termen *trivial*, *gut* und *ueberladen* erzeugt, befinden sich exemplarisch folgende Regeln:

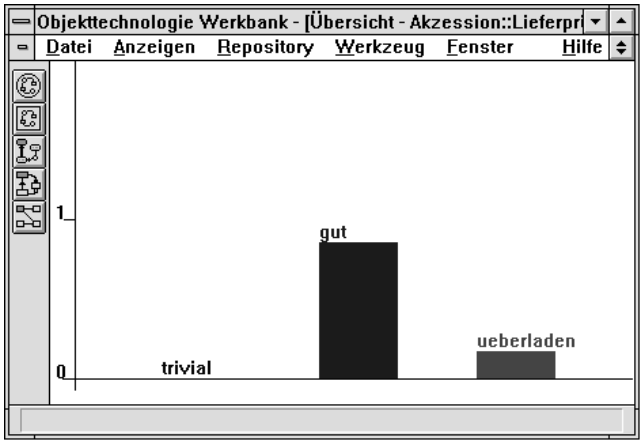

Bild 6: unscharfe Bewertung der Übersicht

WENN AnzahlObjekte == wenig	UND	AnzahlProzesse == wenig	DANN Uebersicht = trivial
WENN AnzahlObjekte == wenig	UND	KantenMass == wenig	DANN Uebersicht = trivial
WENN MittelBeding == wenig	ODER	MaxBeding == wenig	DANN Uebersicht = trivial
WENN AnzahlObjekte == mittel	UND	AnzahlProzesse == mittel	DANN Uebersicht = gut
WENN AnzahlObjekte == wenig	UND	AnzahlProzesse == viel	DANN Uebersicht = gut
WENN MittelBeding == mittel	UND	AnzahlProzesse == mittel	DANN Uebersicht = gut

WENN AnzahlObjekte == viel	UND	AnzahlProzesse == viel	DANN Uebersicht = überladen
WENN AnzahlObjekte == mittel	UND	AnzahlProzesse == viel	DANN Uebersicht = überladen
WENN MaxBeding == viel	UND	Objekte == viel	DANN Uebersicht = überladen

Die Einschätzung des in Bild 4 angegebene Schemas erfüllte die Kriterien einer trivialen Übersichtlichkeit zu 0%, einer guten zu 86% und einer überladenen zu 17%, was als zufriedenstellende Einschätzung zu werten ist. Das Inferenzergebnis ist in Bild 6 angegeben.

Dynamische Modellüberprüfung

Im Gegensatz zu den bisher aufgezeigten Ansätzen zur Modellüberprüfung erfordert die dynamische Überprüfung neben der Existenz des Modells auch die des dem Modell folgenden Anwendungssystems. Die Feststellung der Einhaltung bzw. Verletzung dieser Konsistenz erfolgt während der Ausführung von Testfällen der Anwendung. Zu diesem Zweck wird ein methodenspezifischer Überprüfungscode in den Quelltext der Anwendung generiert, der basierend auf entwickelten Überprüfungsklassen die Einhaltung der im Modell geforderten Restriktionen überwacht. Zum einen obliegt diesem die prozeßbezogene Überprüfung von Vor- und Nachbedingungen, zum anderen kann die Einhaltung von Zeitbedingungen für Prozesse verifiziert werden. Einen Ausschnitt aus einer Quellcodedatei nach der Generierung enthält Bild 7.

```
int Akzession::Feststellung()
{
    //OPMCC: Beginn der Einfügung Farben
    OPMSetColor( PrfgStand, PrfgStandEnum );
                        // legt Farbe _PrfgStand_ vom Typ SetColor
                        // zum Attribut PrfgStand an
    OPMSetColor( StatusAkz, StatusAkzEnum );
                        // legt Farbe _ StatusAkz_ vom Typ SetColor
                        // zum Attribut StatusAkz an
    ColorTransaction t("int Akzession::Feststellung()");
                        // Variable vom Typ ColorTransaction
    t + aktLieferung._pPrfgStand_;
                        // Einfügen der lokal verwendeten Farben
    t + aktLieferung._pStatusAkz_;
                        // Einfügen der lokal verwendeten Farben
    t.Start();

    OPMPreCond(aktLieferung._PrfgStand_ == Lieferung::sachgepr);
                        //Vorbedingung PrfgStand == sachgepr
    OPMPreCond(aktLieferung._PrfgStand_ == Lieferung::fachgepr);
                        //Vorbedingung PrfgStand == fachgepr
    OPMPreCond(aktLieferung._PrfgStand_ == Lieferung::rechngepr);
                        //Vorbedingung PrfgStand == rechngepr
    OPMPreCond(aktLieferung._StatusAkz_ != Lieferung::festgestellt);
                        //Vorbedingung StatusAkz != festgestellt

    OPMPostCond(aktLieferung._StatusAkz_ += Lieferung::festgestellt);
                        //Nachbedingung StatusAkz += festgestellt
    //OPMCC: Ende der Einfügung Farben
    ...
    // regulärer Code der Methode
    };
```

Bild 7: C++ - Quelldatei mit Überprüfungscode

Beim Eintritt in die Methode (Prozeß) ist zu überprüfen, ob die geforderten Vorbedingungen erfüllt sind, vor dem Verlassen desselben ist zu kontrollieren, ob die angegebenen Veränderungen in den Attributen der beteiligten Objekte durchgeführt worden sind. Ebenfalls zum Abschluß des Prozesses kann die Einhaltung von Zeitrestriktionen geprüft werden. Betrachtet man den Prozeß *Person::FachPrfg* (Bild 4), so ist am Prozeßbeginn festzustellen, ob das Attribut *Befugnis* des Objektes *Mitarbeiter* die Belegung *FachtechnPrüfung* besitzt bzw. enthält und im Attribut *PrfgStand* des Objektes *Akz.aktLieferung* die Belegung *fachgepr* nicht vorliegt. Zum Abschluß des Prozesses wird überprüft, ob dem Attribut *PrfgStand* im Laufe des Prozesses die Belegung *fachgepr* hinzugefügt worden ist.

Die Verletzung einer geforderten Bedingung wirft eine Fehlermeldung auf, die die Inkonsistenz aufzeigt.

Zusammenfassung

Voraussetzung für eine flüssige Handhabung im Modellierungsprozeß ist die Implementierung innerhalb eines Werkzeugs. Ein solches befindet sich unter Leitung der TU Ilmenau in Zusammenarbeit mit der OWiS Software GmbH in der Entwicklung, Petri-Netze werden darin für den Nutzer transparent bleiben.

Eine Erfahrung der Modellierungstätigkeit ist, daß die exakte Beschreibung des Geschäftsprozesses in mehreren inkrementellen Zyklen zusammen mit den Fachexperten reift und der detaillierten und formalisierten Notationsform zu verdanken ist, daß oftmals noch bestehende Mißverständnisse ausgeräumt werden konnten.

Ein Ausgangspunkt zur Integration von Software bietet sich auf der Grundlage des OPM, indem zum einen Teile des Unternehmensmodells herausgelöst werden können, die als Softwareimplementierung einzelne Arbeitsprozesse unterstützen, zum andern können sich Möglichkeiten eröffnen, bestehende SW-Komponenten unter Berücksichtigung vorhandener Schnittstellen einzubinden.

Die dynamische Modellierung mit dem Objekt-Prozeß-Modell kann als Ansatz betrachtet werden, die komplexen Abläufe in kooperativen Arbeitsgruppen objektorientiert zu beschreiben. Einen vielversprechenden Forschungsansatz läßt der Einsatz als Beschreibungsmittel innerhalb eines Workflow-Management-Systems erkennen.

Literatur

[Booch94]] Grady Booch, Object Oriented Analysis and Design with Applications, 2nd Edition, Benjamin/Cummings: Redwood City, California, 1994

[Burkhardt92] Rainer Burkhardt, Bernd Breutmann, Objektorientierte Systeme, Grundlagen – Werkzeuge – Einsatz, Hanser: München, Wien, 1992

[Burkhardt94] Rainer Burkhardt, Modellierung dynamischer Aspekte mit dem Objekt-Prozeß-Modell, Technische Universität Ilmenau, Dissertation, 1994

[Cremers94] Arnim B. Cremers, Oleg Balownew, Thomas Bode, Jürgen Kalinski, Heinz Rottmann, Ein Ablösesystem für den Bibliotheksverbund Nordrhein-Westfalen – Machbarkeitsstudie, 1994

[HBZ94] Hochschulbibliothekszentrum NRW, Hermann Kronenberg, Ute Schäfer, Ingrid Töteberg, Florian Seiffert, Anforderungskatalog Ablösesystem, Köln, 1994

[Hesse94] W. Hesse, G. Barkow, H. von Braun, H.-B. Kittlaus, G. Scheschonk, Technologie der Softwaretechnik, Ein Begriffssystem für die Analyse und Modellierung von Anwendungssystemen, Teil 1: Begriffssystematik und Grundbegriffe, Informatik-Spektrum (1994)17: 39-47

[Jacobson94] Ivar Jacobson, M. Ericsson, A. Jacobson, The Object Advantage, Business Process Reengineering With Object Technology, Addison-Wesley: 1994

[Jensen92] Kurt Jensen, Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical Use, Springer: Berlin, 1992

[Keichel95] Stefan Keichel, Algorithmen für Concurrent Object-Oriented Petri Nets, Technische Universität Ilmenau, Institut für Theoretische und Technische Informatik, 1995

[Peneca95] PENECA Chromos 2.1 (für Windows), Colored Petri Net Construction Tool, Handbuch, OWiS Software GmbH, Martinroda, 1995

[Philippow91] Wolfgang Fengler, Ilka Philippow, Entwurf industrieller Mikrocomputersysteme, Hanser: München, Wien 1991

[Pietsch94] Wolfram Pietsch, Dieter Steinbauer, Business Process Reengineering, WIRTSCHAFTSINFORMATIK, 36(1994)5: 502-505

[Rumbaugh91] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, Object-Oriented Modeling and Design, Prentice-Hall: Englewood Cliffs, New Jersey, 1991

[Starke90] Peter H. Starke, Analyse von Peri-Netz-Modellen, B. G. Teubner: Stuttgart, 1990

[Wolf94] Martin Wolf, Bibliotheksmodellierung mit dem Objekt-Prozeß-Modell, Technische Universität Ilmenau, Institut für Theoretische und Technische Informatik, Studienarbeit, 1994

[Wolf95] Martin Wolf, Validierung und Optimierung dynamischer Modelle auf der Basis des Objekt-Prozeß-Modells, Technische Universität Ilmenau, Institut für Theoretische und Technische Informatik, Diplomarbeit, 1995