

Berücksichtigung temporaler Aspekte in der Geschäftsprozessmodellierung am Beispiel öffentlicher Verwaltungen

Robert Braun, Werner Esswein, Andreas Gehlert

Technische Universität Dresden
Lehrstuhl für Wirtschaftsinformatik, insbes. Systementwicklung
MommSENstr. 13
01062 Dresden
{braun | esswein | gehlert}@wise.wiwi.tu-dresden.de

Abstract: Aus analytischen Überlegungen heraus zeigt der Beitrag am Beispiel öffentlicher Verwaltungen, welche temporale Aspekte bei der Modellierung von Geschäftsprozessen prinzipiell berücksichtigt werden müssen und betrachtet im Folgenden zwei davon näher. Zur Demonstration der Nutzenpotentiale der daraus gewonnenen Erkenntnisse wird ein UML-Profil vorgeschlagen und exemplarisch angewendet, das auf diese Aspekte zugeschnitten ist. Die Erkenntnisse sind in ihrem Kern von der Domäne der öffentlichen Verwaltungen und von der eingesetzten Modellierungssprache unabhängig und können damit sowohl auf eine andere Sprache als auch auf eine andere Domäne übertragen werden.

1 Einleitung

1.1 Motivation

Ende der 70er, Beginn der 80er Jahren des letzten Jahrhunderts wuchs aufgrund der zunehmenden Anwendungsbereiche der Informatik und Wirtschaftsinformatik auch die Relevanz, Zeit bei der Entwicklung verschiedenster automatisierter Systeme angemessen zu berücksichtigen (vgl. [RBL88]; [Sc94], S. 284). Die Forschungsschwerpunkte lagen bisher auf den Bereichen der (konzeptuellen) Modellierung temporaler Datenbanken und Real-Time-Systemen (vgl. [Ma91]; [TL91]; [TLW91]; [WI97]; [KJS01]; [KW02]; [KS⁺03]). Demzufolge hat die Berücksichtigung temporaler Aspekte, bezogen auf diese zwei Untersuchungsbereiche, zum gegenwärtigen Zeitpunkt bereits eine intensive theoretische Durchdringung und einen hohen Entwicklungsstand erreicht.

Für die Modellierung temporaler Datenbanken (vgl. [KS⁺03]) bedeutet dies, dass verschiedene um zeitliche Aspekte erweiterte Modellierungstechniken zur Abbildung von Systemstrukturen auf konzeptioneller Ebene entwickelt wurden. Dazu zählen beispielsweise das *Spatio-Temporal Entity-Relationship Model (STER)* (vgl. [KS⁺03], S. 82ff) oder die *Temporal Unified Modeling Language (TUML)* (vgl. [ST99]), die sich jedoch lediglich auf die Erweiterung des Klassendiagramms der *Unified Modeling Language (UML)* (vgl. [OM03a]) beschränkt. Weitere Diagrammarten der UML wurden auch hier bisher nicht beleuchtet.

Für die Modellierung von Real-Time-Systemen³ bedeutet dies, dass verschiedene formale Sprachen und bestehende Modellierungstechniken zur Abbildung der Strukturen und Funktionen dieser Systeme entwickelt bzw. erweitert wurden. Dazu zählen beispielsweise das *Statechart* (vgl. [HP85]; [Ha87]) und dessen spätere Derivate sowie das UML-Profil für *Schedulability, Performance and Time Specification* (vgl. [OM03b]). Weitere Beispiele sind bei DOUGLASS zu finden (vgl. [Do99], 721f).

Da inzwischen mindestens die Hälfte der softwareentwickelnden Industrie in die Entwicklung von Anwendungssystemen für Wirtschaft und Verwaltung und in die damit einhergehende Geschäftsprozessmodellierung involviert ist (vgl. [BJ02], S. 7), gewinnt die Notwendigkeit der Berücksichtigung temporaler Aspekte in der konzeptuellen Modellierung in diesem Untersuchungsbereich zunehmend an Bedeutung. Jedoch lassen sich die Erkenntnisse und die entsprechenden Modellierungstechniken aus den Bereichen der temporalen Datenbanken und Real-Time-Systemen aufgrund der unterschiedlichen Spezifika der Untersuchungsbereiche nur teilweise für die Geschäftsprozessmodellierung nutzen.

Besonders für das zentrale Teilgebiet der der Geschäftsprozessmodellierung per se inhärenten Modellierung dynamischer Aspekte sind die existierenden Konzepte aus dem Bereich der Real-Time-Systeme ohne Modifikation zur Anwendung ungeeignet. Eine verwendete Modellierungssprache muss zwar einerseits eine adäquate und vollständige Abbildung des Gegenstandsbereichs ermöglichen (Grundsatz der Sprachadäquanz), aber andererseits auch dem Modellnutzer eine gute Verständlichkeit der Modelle bieten (Grundsatz der Klarheit) (vgl. [Sc98b], S. 131ff). Da in der Geschäftsprozessmodellierung mit Fachexperten gearbeitet wird, d. h. mit Personen, die nur über geringe Modellierungskennntnisse verfügen, müssen die Modellierungssprachen dafür besonders intuitiv ausgestaltet werden. Der formale Charakter der Modellierungssprachen aus dem Bereich der Real-Time-Systemmodellierung wird dieser Anforderung nicht gerecht.

³ Der Begriff Real-Time-System wird hier im weiteren Sinne auch für reaktive bzw. ereignisgesteuerte (event-driven) und zeitgesteuerte (time-driven) Systeme verwendet (vgl. [Do99], S. 58).

Beispielsweise wird das Zustandsdiagramm und im Besonderen das Aktivitätsdiagramm (vgl. [OM03a], S. 2-170; [OW⁺03], S. 175ff) bereits seit längerem auch im Bereich der Geschäftsprozessmodellierung eingesetzt (vgl. [OM03a], S. 2-140). Die gute Verständlichkeit der so erstellten Geschäftsprozessmodelle wird dabei jedoch lediglich durch die Nicht- bzw. unzureichende Darstellung von temporalen Aspekten erreicht und führt daher zu Konflikten mit dem Grundsatz der Sprachadäquanz. Das Problem der Berücksichtigung von Zeitaspekten in der Geschäftsprozessmodellierung bleibt in diesem Umfeld deshalb weitgehend ungelöst.⁴

Auf dem Gebiet der Strukturierung und Darstellung von temporalen Aspekten bei der Geschäftsprozessmodellierung existieren zur Zeit nur wenige Ansätze. Hervorzuheben ist HOHEISEL, welcher temporale Aspekte mittels Geschäftsregeln modelliert, denen die ECAA-Notation (Event-Condition-Action-Alternative Action) zugrunde liegt (vgl. [Ho99]). Dem Ansatz inhärent ist seine Beschränkung auf die Abbildung von Zeitereignissen und deren Gültigkeit, die jedoch – wie gezeigt wird – nur einen der temporalen Aspekte von Geschäftsprozessen ausmachen. Ferner wird die Abbildung komplexer Geschäftsprozesse mittels der ECAA-Notation ohne geeignete Dekompositionsregeln schnell unübersichtlich (vgl. bspw. [Ho99], S. 17, Abbildung 19) und erstellte Modelle geraten unnötig in Konflikt mit dem Grundsatz der Klarheit von Modellen.

1.2 Zielstellung

In diesem Beitrag sollen zunächst in Abschnitt 2 die prinzipiell möglichen temporalen Aspekte in der Geschäftsprozessmodellierung analysiert werden. Dies erfolgt am Beispiel öffentlicher Verwaltungen. Dabei wird auch eine Einschränkung der Aspekte vorgenommen, die den Gegenstand des Beitrages bilden sollen und im weiteren Verlauf des Abschnittes dazu näher betrachtet werden.

Da die von der OBJECT MANAGEMENT GROUP (OMG) standardisierte UML sich während ihrer nunmehr 6-jährigen Existenz auch im Bereich der konzeptuellen Modellierung inzwischen als de facto Standard etabliert hat⁵ (vgl. [JHC02], S. V; [OW⁺03], S. 147) und deswegen auch verstärkt den Kontext für die neueren Arbeiten der konzeptuellen temporalen Modellierung bildet (vgl. [ST99]; [Do99]; [KW02]; [APR02]; [FM02]), werden die so gewonnenen Erkenntnisse in Abschnitt 3 exemplarisch in die UML implementiert, um deren Anwendbarkeit nachzuweisen. Dazu wird ein entsprechendes UML-Profil erstellt und ein zugehöriges Modellierungsbeispiel präsentiert.

Eine Zusammenfassung in Abschnitt 4 schließt den Beitrag.

⁴ So steht beispielsweise im Aktivitätsdiagramm der UML lediglich das Konzept `Signal` zur temporalen Modellierung zur Verfügung, mit dessen Hilfe sich jedoch nur Zeitereignisse abbilden lassen (vgl. [OW⁺03], S. 185).

⁵ „The amount of modeling projects that use UML, the amount of books written about UML and the number of software tools that support UML are large in relation with the analogous practical achievements of other modeling languages. This proves that UML in its current state is more practical than other modelling solutions, ...“ (vgl. [NW02], S. 3f).

2 Temporale Aspekte in der Geschäftsprozessmodellierung

2.1 Analyse

Viele Geschäftsprozesse und insbesondere solche der öffentlichen Verwaltungen sind an bestimmte zeitliche Perioden gekoppelt, in deren Rahmen sie stattfinden müssen bzw. nach denen sich diese Prozesse ständig wiederholen. Die zentrale Zeitperiode in der öffentlichen Verwaltung ist dabei das Haushaltsjahr. Diesem ordnen sich alle Prozesse unter, d. h. für jedes Haushaltsjahr repetieren sich die Prozesse entsprechend. Zudem existiert eine ganze Reihe von unterjährigen (Teil-) Prozessen, für die eine von einem Haushaltsjahr abweichende, aber dennoch fest bestimmte Zeitperiode als gesetzt gelten muss, innerhalb derer sie sich wiederholen. Diese Zeitperiode kann dabei aus gesetzlichen Vorschriften oder einfach aus der Charakteristik des Prozesses resultieren. Zum Beispiel erfolgt die Vereinnahmung und Verausgabung von Haushaltsmitteln täglich, die Aufstellung eines Haushaltsplanes jährlich und die Erstellung von Abschlüssen täglich, monatlich und jährlich.

Die Prozessmodelle sind daher stets durch eine von Modell zu Modell möglicherweise variierende Zeitperiode gekennzeichnet, nämlich die, in deren Rahmen der jeweils dargestellte Prozess ausgeführt wird. Ein zu berücksichtigender temporaler Aspekt in der Geschäftsprozessmodellierung ist demnach die *Periodizität von Geschäftsprozessmodellen*.

Weiterhin liegen Prozessen stets Informationsobjekte zugrunde, an denen und/oder mit deren Hilfe sie ausgeführt werden (vgl. [Sc98b], S. 100).⁶ Unter temporalen Gesichtspunkten ist dabei interessant, dass für (periodische) Prozesse Informationsobjekte unterschiedlicher Zeitperioden benötigt werden. Beispielsweise könnten in einen Prozessschritt zur Erstellung einer monatlichen Statistik sämtliche Wochenstatistiken des betroffenen Monats oder auch die Vormonatsstatistik einfließen. Ein weiterer zu berücksichtigender temporaler Aspekt in der Geschäftsprozessmodellierung ist demnach der *Zeitbezug der Informationsobjekte* zu den Prozessmodellen, in denen sie dargestellt sind.

Ferner zeichnen sich Informationsobjekte durch eine unterschiedliche zeitliche Gültigkeit ihrer Attribute und unterschiedliches Interesse an der Kenntnis der Historie über den Attributausprägungsverlauf aus. Beispielsweise ist das Geburtsdatum eines Mitarbeiters über den Zeitverlauf konstant, während seine Stelle, die er im Unternehmen begleitet oder sein Gehalt, variieren können. Wobei zudem eventuell der Karriereverlauf des Mitarbeiters noch von Interesse ist. Die Attribute eines Informationsobjekttypen lassen sich daher in zeitabhängige und zeitunabhängige Attribute unterteilen. Ein weiterer zu berücksichtigender temporaler Aspekt in der Geschäftsprozessmodellierung ist deshalb der *Zeitbezug der Attribute von Informationsobjekttypen*.

⁶ Abweichend von SCHÜTTE, der einem Prozess genau ein prägendes Informationsobjekt zugrunde legt (vgl. [Sc98b], S. 100, wird hier die Möglichkeit, dass einem Prozess mehrere Informationsobjekte zugrunde liegen können, nicht ausgeschlossen. Die Einschränkung von SCHÜTTE wird als Spezialfall angesehen, in dem die Granularitäten eines Informationsobjektes und eines Prozessschrittes entsprechend homogen sind.

Diese Erkenntnis entstammt der konzeptuellen Modellierung im Zusammenhang mit temporalen Datenbanken (vgl. [TLW91]) und ist für die Datenmodellierung der Informationsobjekttypen relevant und auch bereits angewendet worden (vgl. [KW02]). In diesem Beitrag soll sich daher explizit auf die Prozesssicht konzentriert werden, so dass dieser temporale Aspekt nicht Gegenstand des Beitrages ist.

Letztlich sind einzelne Prozessschritte eines Geschäftsprozesses zum Zwecke einer Geschäftsprozessplanung und -steuerung selbst noch zeitlich attribuiert. Beispielsweise können bestimmte Bearbeitungsdauern für Prozessschritte festgelegt sein oder Fristen, zu denen sie ausgeführt sein müssen und dies im Zusammenwirken mit vor- und nachgelagerten Prozessschritten. Daher ist ein weiterer zu berücksichtigender temporaler Aspekt in der Geschäftsprozessmodellierung die *zeitliche Ablaufplanung von Prozessschritten*.

SCHEER (vgl. [Sc98a], S. 31ff) verweist im Zusammenhang der Modellierung einer zeitlichen Ablaufplanung von Prozessschritten auf Methoden der Netzplantechnik (vgl. dazu [Jo74]; [Ru99], S. 180ff) und auch der bereits vorgestellte Ansatz von HOHEISEL lässt sich diesem temporalen Aspekt der Geschäftsprozessmodellierung zuordnen. Daher soll dieser letzte zeitliche Aspekt aufgrund der bereits bestehenden Lösungsansätze in der Literatur, seiner eigenen Komplexität und Unabhängigkeit von den anderen Aspekten ausdrücklich nicht Gegenstand dieses Beitrages sein. Vielmehr wird sich im Folgenden auf die Periodizität von Geschäftsprozessmodellen und den Zeitbezug von Informationsobjekten beschränkt.

2.2 Periodizität von Geschäftsprozessmodellen

Der temporale Aspekt der Periodizität von Geschäftsprozessmodellen lässt sich wie folgt formalisieren.

Es sei z_p definiert als eine nicht fest in der Realzeit verankerte **Zeitperiode**, d. h. sie ist nicht an ein konkretes Datum gebunden (beispielsweise den Dezember 2003 oder den 23.10.2003). Als einzig zulässige Ausprägungen der Zeitperiode z_p werden dabei unterschieden:

- $j \equiv \text{Jahr}$
- $m \equiv \text{Monat}$
- $w \equiv \text{Woche}$
- $t \equiv \text{Tag}$
- $s \equiv \text{stetig}$ ⁷

⁷ Eine Zeitperiode wird als **stetig** bezeichnet, wenn sie keiner der anderen Zeitperioden (j ; w ; m ; t) zuordenbar ist $s \notin (j; w; m; t)$.

Die höchste Ebene in der Prozessmodellierung wird aufgrund der bereits skizzierten Bedeutung des (Haushalts-) Jahres für die Geschäftsabläufe in der öffentlichen Verwaltung mit einer Zeitperiode von $zp=j$, also gleich einem Haushaltsjahr, fixiert. Nichtsdestotrotz können einige Prozessschritte der höchsten Ebene unterjährigen Wiederholungen unterliegen.⁸

Zwischen den Zeitperioden gilt die Beziehung $t \subset w \subset m \subset j$. Das bedeutet, dass ein Tag Teil einer Woche, eine Woche Teil eines Monats und ein Monat Teil eines Haushaltsjahres ist. Für $w \subset m$ muss zudem präzisiert werden, dass eine Woche, die sich auf zwei Monate aufteilt, stets dem ersten Monat zugerechnet wird.

Die Teile-Ganzes Beziehung zwischen den Zeitperioden ist insofern von Bedeutung, weil dadurch ausgesagt wird, dass nur solche zeitperiodendifferenzierenden Prozessmodellhierarchien durch Prozessfaltungen, d. h. Verfeinerung von Prozessschritten in separaten Modellen, aufgebaut werden können, in denen die Zeitperiode des untergeordneten Modells einer Zeitperiode zp_{mu} entsprechen muss, die Teil oder gleich der Zeitperiode zp_{mo} des übergeordneten Modells ist, in der also $zp_{mu} \subseteq zp_{mo}$ gilt. Damit wird ausgeschlossen, dass beispielsweise ein Modell, das einen monatlich ablaufenden Prozess visualisiert (für das also $zp=m$ gilt) einen Prozessschritt enthalten kann, der in einem weiteren Modell verfeinert wird und dieses Modell mit $zp=j$ deklariert wird, denn dann könnte der Prozess in dem übergeordneten Modell nicht innerhalb eines Monats ablaufen, da der verfeinerte Prozessschritt erst nach einem Jahr beendet ist.

Diese beschriebene Inklusionsbeziehung ist besonders beim Umgang mit der Zeitperiode $zp=s$ zu beachten und mit der nötigen Sorgfalt anzuwenden, da sich Prozessmodelle a posteriori einer dahingehenden Überprüfung entziehen können. Dies ist beispielsweise dann der Fall, wenn sich die Zeitperiode $zp=s$ für ein Prozessmodell über einen Zeitraum erstreckt, der zwischen einer Woche und einem Monat liegt. Dann kann ein übergeordnetes Prozessmodell keine Zeitperiodenzuordnung von $zp=t$ aufweisen.

Es wird ersichtlich, dass die Notwendigkeit besteht, Zeitperioden für eine Modellierung zu definieren, aber auch, dass diese je nach Kontext beliebig festlegbar sind. Dies ist besonders im Hinblick auf die Zeitperiode $zp=s$ relevant, da diese letztlich nur einen Container darstellt, der nicht weiter definierte Zeitperioden enthält. Durch zweckmäßige Zeitperiodendefinition kann diese Menge gering gehalten werden und ist im besten Fall leer.

⁸ Der vorgestellte Ansatz schließt auch eine überjährige Wiederholung von Prozessschritten oder eine komplette Änderung der Zeitperiode des Ausgangsmodells prinzipiell nicht aus, nur wird sich zur Komplexitätsreduktion des Beitrages wie dargelegt beschränkt.

2.3 Zeitbezüge von Informationsobjekten

Um temporale Aspekte von Informationsobjekten zu formalisieren, wird das Konzept Informationsobjekt dazu neben einem bestimmten (Bearbeitungs-) Status mit einem Zeitbezug zu dem modellierten Prozess attribuiert.

Der **Status** eines Informationsobjektes entspricht dabei einem bestimmten Zustand, in dem sich das Informationsobjekt befindet (bspw. Zustand [geprüft]) und manifestiert sich durch die Belegung seiner Attribute (vgl. [Sc98b], S. 102).

Der **Zeitbezug** zb eines Informationsobjektes setzt sich aus einer Zeitperiode (zp) und einer Zeitkonstante (zk) zusammen, also es gilt: $zb=zp+zk$, und ist über alle Prozessmodelle hinweg zum einen für dieses Informationsobjekt *unveränderlich* und zum anderen für alle Informationsobjekte einer Klasse *identisch*.

Für die Zeitperiode zp muss dabei ferner gelten, $zp \in \{j; w; m; t\}$, d. h. zp muss einer der als zulässig definierten Zeitperioden aus Abschnitt 2.2 entsprechen. Die Beziehung $t \subset w \subset m \subset j$ aus Abschnitt 2.2 impliziert darüber hinaus, dass ein Informationsobjekt, welches z. B. einer bestimmten monatlichen Zeitperiode zugeordnet wurde (z. B. $zb=m-1$), auch immer einem bestimmten Jahr zugeordnet ist, sofern eine entsprechende modellzeitdifferierende Prozessmodellhierarchie existiert. Der Umkehrschluss ist hingegen nicht zulässig.

Die Höhe der **Zeitkonstante** zk gibt eine relative Verschiebung zur Zeitperiode zp an. Sie richtet sich nach sachlogischen Überlegungen und ist damit auch nur im Kontext überprüfbar. Sie ist notwendig um auszudrücken, dass beispielsweise ein Informationsobjekt zu einem Vormonat gehört.

Der Zeitbezug ($zb=zp+zk$) eines Informationsobjektes drückt damit aus, welcher Zeitperiode es zuzuordnen ist (Haushaltsjahr, Monat, Woche) sowie auf welche (relative) Zeitperiode dabei referenziert wird (aktuelle Woche, vorheriger Monat, usw.).

In der Regel bezieht sich der Zeitbezug eines Informationsobjektes immer direkt auf die aktuelle Zeitperiode ($zk=0$). In Ausnahmefällen kann jedoch ein dargestelltes Informationsobjekt auch einer vorangegangenen oder zukünftigen Zeitperiode ($zk \neq 0$) zugeordnet sein.

Die Zuordnung zu einer vorangegangenen Zeitperiode muss vorgenommen werden, wenn z. B. in einem Modell mit $zp=j$, in welchem der Ablauf eines Jahresabschlusses modelliert wird, am Ende nicht nur ein Informationsobjekt Jahresabschluss für das aktuelle Jahr entsteht, sondern zu Beginn des Prozesses der Jahresabschluss des Vorjahres als Input benötigt wird. Der Jahresabschluss des Vorjahres ist dem Vorjahr ($zb=j-1$) zuzuordnen.

Analog verhält es sich mit Informationsobjekten, die Folgeperioden zuzuordnen sind. Beispielsweise kann in einem Modell mit $z_p = j$, welches den Prozess der Aufstellung eines Haushaltsplanes eines Bundeslandes darstellt, ein (initiales) Informationsobjekt Haushaltsplan erstellt werden, der aber für das darauffolgende Jahr erst gilt. Der Prozess selbst gehört zum aktuellen Jahr, das Informationsobjekt Haushaltsplan lässt sich sachlogisch dem Folgejahr (Folgeperiode; ($z_b = j - 1$)) zuordnen.

Bei allen Informationsobjekten, die nicht aus der aktuellen Zeitperiode z_p stammen, ist der Zeitbezug relativ zur aktuellen Periode zu belegen ($z_k \neq 0$). Dafür gilt, resultierend aus der Nestung von Modellen mit unterschiedlicher Periodizität, folgende allgemeine Konvention:⁹

$$z_b = z_p + z_k \text{ mit } z_p \in \{j; m; w; t; s\} \text{ und } z_k := \begin{cases} |z_k| \in \{1; \dots; \infty\}; & \text{für } z_p = j, \\ |z_k| \in \{1; \dots; 12\}; & \text{für } z_p = m, \\ |z_k| \in \{1; \dots; 5\}; & \text{für } z_p = w, \\ |z_k| \in \{1; \dots; 31\}; & \text{für } z_p = t, \\ |z_k| = n; & \text{für } z_p = s. \end{cases}$$

Wobei die Zeitkonstante z_k ($z_k \in \mathbb{Z}$) der Anzahl von Zeitperioden entspricht, die zwischen der aktuellen Periode und derjenigen, aus der das Informationsobjekt stammt, liegen. Für den Fall von $z_p = s$ wird statt einer konkreten Zeitkonstante z_k der Platzhalter n eingesetzt, da durch eine weitere Konkretisierung dem Modell keine weitere Bedeutung hinzugefügt werden würde. Das liegt daran, dass nicht ersichtlich ist, welche konkrete Periode z_p assoziiert (2 Wochen, 3 Tage, etc.) und somit eine relative Auszeichnung zu dieser nicht bekannten Periode ohne verwendbaren Informationsgehalt ist.

Im Folgenden werden die Konsistenzbedingungen spezifiziert, die sich aus der Verwendung dieses Modellierungsansatzes bezüglich der Informationsobjekte ergeben, die nicht der aktuellen Zeitperiode entstammen.

⁹ Durch diese Formalisierung wird allerdings auch die Grenze des Ansatzes deutlich, nämlich dass eine konsistente Modellierung von Informationsobjekten mit einem beliebigen Zeitbezug (bspw. $m - 20$) damit nicht möglich ist.

Konsistenzbedingung für Informationsobjekte aus Vorperioden

Bei Informationsobjekten, die aus einer vorangegangenen Zeitperiode stammen und erstmalig für Prozessschritte der aktuellen Periode benötigt werden, muss als Konsistenzbedingung allgemein gefordert werden, dass der Status eines Informationsobjektes mit dem Zeitbezug $z_b = z_p + z_k$ mit $z_p \in \{j; w; t; s\}$ und $z_k \in \mathbb{Z}: z_k < 0$, mit dem es für einen Prozessschritt benötigt wird, einem Status dieses Informationsobjektes mit dem Zeitbezug $z_b = z_p + z_k + k$ mit $k \in \mathbb{N}$ und $|z_k| \geq k \geq 1$ entspricht. Dabei wird offensichtlich, dass zugeordnete und aktuelle Zeitperiode zusammenfallen, wenn als aktuelle Zeitperiode des Modells die zugeordnete des Informationsobjektes angenommen wird, da in diesem Fall $z_p + z_k + k = 1$ bzw. $z_k = k$ gilt.

Die beschriebene Konsistenzbedingung ist erforderlich um sicherzustellen, dass Informationsobjekte, die in einen Prozessschritt in einem bestimmten Zustand eingehen, auch (zeitlich) zuvor in diesen Zustand versetzt wurden.

Konsistenzbedingung für Informationsobjekte aus Folgeperioden

Bei der Modellierung von Informationsobjekten, die einer zukünftigen Zeitperiode zugeordnet sind, ist zu beachten, dass damit gleichzeitig vorausgesetzt wird, dass diese Informationsobjekte in der aktuellen oder bereits in einer der vorangegangenen Perioden generiert wurden. Die dafür verantwortlichen Prozessschritte müssen aus Gründen der Klarheit mit in den Modellen erfasst werden.

Als Konsistenzbedingung muss allgemein gefordert werden, dass ein Status eines Informationsobjektes mit einer Folgeperiodenzuordnung $z_b = z_p + z_k$, wobei $z_p \in \{j; m; w; t; s\}$ und $z_k \in \mathbb{Z}: z_k > 0$ gilt, dem ersten Status dieses Informationsobjektes mit dem Zeitbezug $z_b = z_p + z_k - k$ mit $k \in \mathbb{N}$ und $z_k \geq k \geq 1$ entspricht, in dem dieses erstmalig wieder für einen Prozessschritt benötigt wird.

Diese Konsistenzbedingung ist erforderlich um sicherzustellen, dass Informationsobjekte, die in einem bestimmten Zustand für eine Folgeperiode erstellt wurden und in der aktuellen Periode benötigt werden, zuerst einen Zustand haben, in den sie zeitlich gesehen nun bereits in der Vergangenheit für die zukünftige (jetzt aktuelle) Periode versetzt wurden. Da sie bereits in einem oder mehreren Zuständen vorliegen, müssen sie auch in einem von diesen zunächst verwendet werden, unabhängig davon, welche Zustände dann folgen.

3 Exemplarische Implementierung in die UML

Zur Demonstration, wie die spezifizierten temporalen Aspekte der Geschäftsprozessmodellierung in eine konkrete Modellierungssprache eingebettet werden können, wird exemplarisch die UML herangezogen (vgl. [OM03a]). Da beide betrachteten temporalen Aspekte, wie in Abschnitt 1.1 ausgeführt, in der bisherigen Geschäftsprozessmodellierung – und damit auch in der UML – nicht berücksichtigt wurden, wird im Folgenden ein eigener Vorschlag unterbreitet. Abschnitt 3.1 beschäftigt sich mit der dazu notwendigen Erweiterung der UML und in Abschnitt 3.2 werden die Konzepte anhand eines Modellierungsbeispiels angewendet.

3.1 Das UML-Profil zur Geschäftsprozessmodellierung

Durch domänenspezifische Profile kann die UML an die konkreten Bedürfnisse innerhalb eines Projekts angepasst werden (vgl. [OM03a], S. 2-73ff). Im vorliegenden Fall ist die konsistenzhaltende Einbindung der Periodizität von Geschäftsprozessmodellen und der Zeitbezüge von Informationsobjekten zu berücksichtigen.

Für die Modellierung der Zeitbezüge der Prozesse selbst wird auf das von OESTERREICH vorgestellte Konzept der *multiplen Transitionen* aufgebaut (vgl. [Oe01], S. 189), das im Folgenden kurz skizziert wird. Der Grundgedanke ist, dass sich die Nebenläufigkeit von Prozessschritten nicht wie bei der „normalen“ Parallelisierung und Synchronisation stets die nebenläufige Ausführung von *verschiedenen* Prozessschritten bedeuten muss, sondern vielmehr sich auch auf die zu sich selbst nebenläufig verhaltenden Prozessschritte (im Hinblick auf deren Instanzen) beziehen kann.

Zur Modellierung von Parallelisierungen und Synchronisationen von (Teil-) Prozessen ist in der UML das Konzept `Pseudostate` vorgesehen (vgl. [OM03a], S. 2-145f). Eine multiple Transition repräsentiert sich durch einen speziellen Constraint (vgl. dazu [OM03a], S. 3-26ff), der diesem UML-Konzept zugeordnet wird.

Abbildung 1 veranschaulicht den von OESTERREICH adaptierten Vorschlag beispielhaft. Der obere horizontale Balken entspricht der grafischen Darstellung einer Parallelisierung – in der UML-Notation handelt es sich dabei um die Visualisierung eines `Pseudostate`, bei dem das Attribut `kind` mit dem Wert `fork` belegt ist – mit dem Merkmal, dass optisch nur eine Transition hinausführt, die aber mehrfach (multiple) geschaltet wird. Der angetragene Constraint `{pro Ressort}` besagt, dass die folgenden Prozessschritte für alle Ministerien (Ressorts) des Bundeslandes wiederholt werden. Der untere horizontale Balken entspricht der grafischen Darstellung einer Synchronisation – in der UML-Notation handelt es sich dabei um die Visualisierung eines `Pseudostate`, bei dem das Attribut `kind` mit dem Wert `join` belegt ist – mit dem Merkmal, dass optisch nur eine Transition hineinführt, die aber ebenfalls mehrfach geschaltet wird.

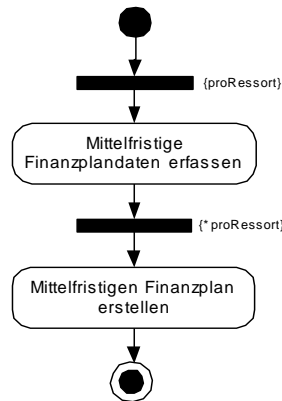


Abbildung 1: Verwendung einer multiplen Transition nach OESTERREICH (vgl. [Oe01], S. 189)

Die Idee von OESTERREICH nun zugrunde gelegt, lässt sich der Zeitbezug von Prozessmodellen analog in die UML einbinden. Notwendig ist die Bildung von zwei Stereotypen (vgl. dazu [OM03a], S. 2-74) im UML-Profil *Geschäftsprozessmodellierung*, die das UML-Konzept *Pseudostate* dahingehend spezialisieren, dass die in Abschnitt 2.3 definierten Zeitperioden über Constraints zugeordnet sind.

Dabei ist zu beachten, dass ein stetiger Zeitbezug ($zP=s$) ggf. weiter untersetzt werden kann. Um eine Separierung der Constraints in diesem Falle zu gewährleisten, muss die Zeitperiode s weiter spezifiziert werden (z. B. vierzehntägig, zweimonatlich etc.), wobei diese Zeitperiode formal immer mit s korrespondiert. Dies hat darüber hinaus den Vorteil, dass die in Abschnitt 2.2 beschriebenen Defizite, die mit der Verwendung der Zeitperiode s einhergehen, reduziert werden.

Die Auszeichnung von Informationsobjekten mit ihrem Zeitbezug ist ebenfalls über die Bildung eines Stereotypen realisierbar, indem das UML-Konzept *Object*, welches zur Darstellung von Informationsobjekten in Prozessmodellen dient (vgl. [OM03a], S. 2-101), dahingehend spezialisiert wird, dass diesem ein spezielles Attribut *zeitbezug* zugeordnet wird.

Abbildung 2 veranschaulicht die getroffenen Aussagen, in dem sie das entworfene UML-Profil mit den gebildeten Stereotypen darstellt.

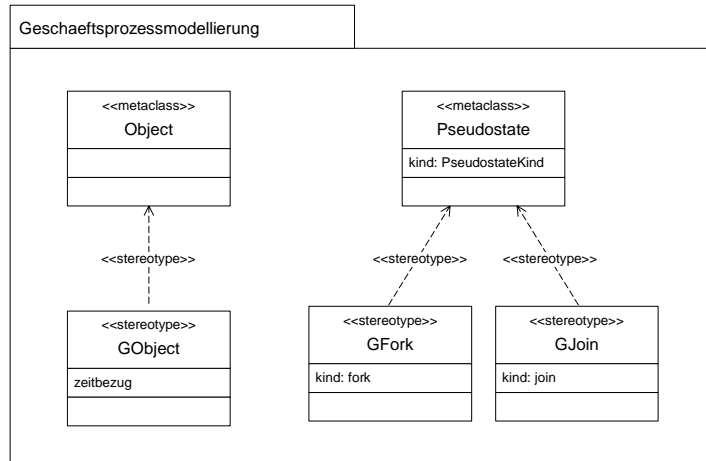


Abbildung 2: UML-Profil *Geschäftsprozessmodellierung*

Tabelle 1 spezifiziert die Stereotypen und ggf. deren Constraints zusätzlich in prosaischer Form (vgl. dazu [OM03a], S. 2-83f). Die Spalte *Stereotyp* beinhaltet den Namen des Stereotypen, die Spalte *Basisklasse* benennt dasjenige UML-Konzept, vom welchem der Stereotyp gebildet wurde, die Spalte *Beschreibung* erläutert die Differenzierung zur Basisklasse und die Spalte *Constraints* beinhaltet die für diesen Stereotyp geltenden Constraints.

Standard	Basisklasse	Beschreibung	Constraints
GObject	Object	GObject ist ein Object, welches ein zusätzliches Attribut zeitbezug besitzt.	Das Attribut zeitbezug darf die Ausprägung $zp+zk$ annehmen, mit $zp \in \{j; m; w; t; s\}$ und $zk \in Z$; wobei für zk die in Abschnitt 2.3 festgelegten Restriktionen gelten.
GFork	Pseudo-state	Ein GFork ist ein Pseudostate, dessen Attribut kind mit fork belegt ist, d. h. es gilt: kind = fork.	Ein GFork ist mit einem der folgenden Constraints zu versehen: <ul style="list-style-type: none"> • {pro Jahr} verweist auf einen Zeitbezug der nachfolgenden Prozessschritte von $zp=j$, also einem Haushaltsjahr. • {pro Monat} verweist auf einen Zeitbezug der nachfolgenden Prozessschritte von $zp=m$, also einem Monat. • {pro Woche} verweist auf einen Zeitbezug der nachfolgenden Prozessschritte von $zp=w$, also einer Woche. • {pro Tag} verweist auf einen Zeitbezug der nachfolgenden Prozessschritte von $zp=t$, also einem Tag. • {pro Quartal} oder {aller zwei Wochen} verweisen stets auf einen Zeitbezug der nachfolgenden Prozessschritte von $zp=s$, also auf einen stetigen.
GJoin	Pseudo-state	Ein GJoin ist ein Pseudostate, dessen Attribut kind mit join belegt ist, d. h. es gilt: kind = join	Ein GJoin ist mit einem der folgenden Constraints zu versehen: <ul style="list-style-type: none"> • {* pro Jahr} synchronisiert einen Zeitbezug von vorangegangenen Prozessschritten von $zp=j$, also einem Haushaltsjahr. • {* pro Monat} synchronisiert einen Zeitbezug von vorangegangenen Prozessschritten von $zp=m$, also einem Monat. • {* pro Woche} synchronisiert einen Zeitbezug von vorangegangenen Prozessschritten von $zp=w$, also einer Woche. • {* pro Tag} synchronisiert einen Zeitbezug von vorangegangenen Prozessschritten von $zp=t$, also einem Tag. • {* pro Quartal} oder {* aller zwei Wochen} synchronisieren stets einen Zeitbezug von vorangegangenen Prozessschritten von $zp=s$, also einen stetigen.

Tabelle 1: Spezifikation der Stereotypen

3.2 Modellierungsbeispiel

Abbildung 3 veranschaulicht an einem Modellierungsbeispiel die beschriebene Implementierung der in diesem Beitrag erörterten temporalen Aspekte bei einer Geschäftsprozessmodellierung in die UML.

Das Modell stellt in diesem Fall die höchste Modellierungsebene dar und besitzt danach implizit einen Zeitbezug von einem Haushaltsjahr ($z_p=j$, vgl. Abschnitt 2.2). Der Prozess startet mit einem Prozessschritt `Plandaten übernehmen` in den die buchhalterischen Plandaten aus dem Vorjahr einfließen. Dies wird durch das Informationsobjekt `Plandaten` vom (Stereo-) Typ `Gobject` ersichtlich, welches mit seinem Attribut `zeitbezug`, in der Ausprägung $j-1$, ausgezeichnet ist. Im Anschluss an diesen Prozessschritt folgen Prozessschritte, die sich im Rahmen des Haushaltsjahres monatlich wiederholen. Dies ist an dem horizontalen Balken als grafische Repräsentation einer Parallelisierung (`GFork`) erkennbar, welcher mit dem Constraint `{pro Monat}` versehen ist. Am Ende liegen zum Zeitpunkt der zeitlichen Synchronisation (erkennbar an dem horizontalen Balken als grafische Repräsentation einer Synchronisation (`GJoin`), welcher mit dem Constraint `{* pro Monat}` versehen ist) demnach 12 Informationsobjekte der Klasse `Monatsabschluss` vor, welche in den nachfolgenden Prozessschritt `Jahresabschluss erstellen` mit einfließen, der wiederum nur einmal pro Jahr stattfindet. Eine Verwendung einer Parallelisierung (`GFork`), die mit dem Constraint `{pro Jahr}` versehen ist, statt der Synchronisation mit dem Constraint `{* pro Monat}`, widerspricht der Semantik der zugehörigen Basisklasse der UML, dem Pseudostate mit der Attributbelegung `kind=fork` und ist demnach auch in diesem UML-Profil nicht zulässig.

Der Prozessschritt `Monatlichen Rechnungsabschluss erstellen` besitzt eine Verfeinerung. Dies ist beispielhaft durch die Auszeichnung eines entsprechenden Attributes (`verfeinert`) kenntlich gemacht. Dies bedeutet für das Prozessmodell, in dem dieser Prozessschritt dargestellt ist, dass es per se einen monatlichen Zeitbezug besitzt ($z_p=m$).

Am Ende resultiert aus dem dargestellten Beispielprozess ein `Jahresabschluss` im Zustand `[geprüft]`, der dem modellierten (aktuellen) Haushaltsjahr zugerechnet wird ($z_p=j$).

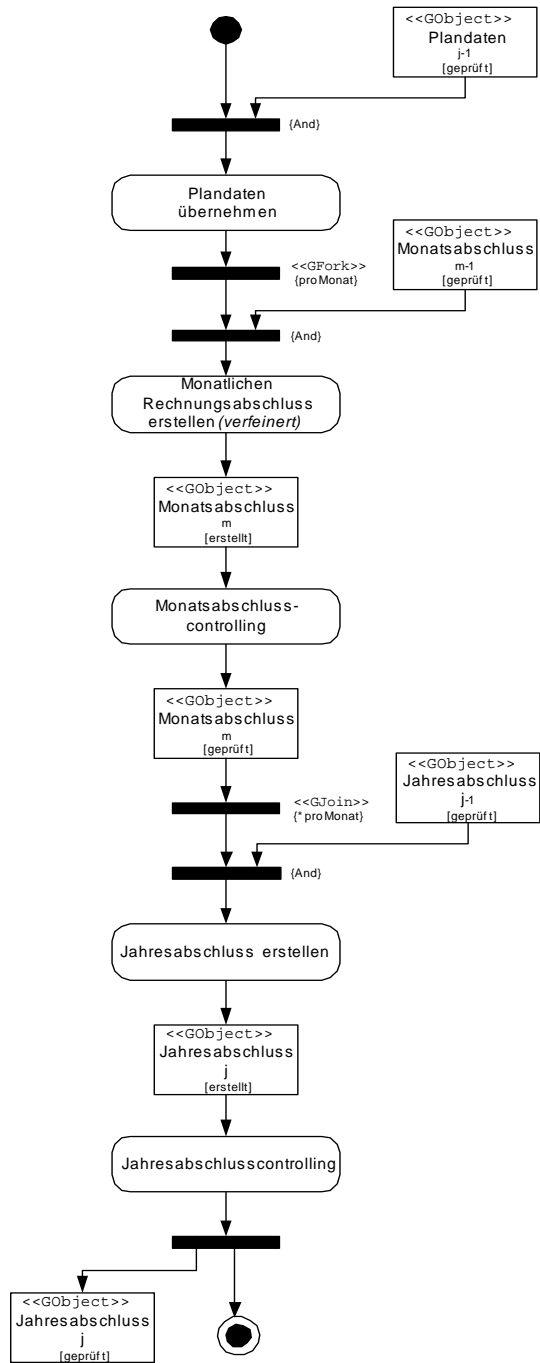


Abbildung 3: Modellierungsbeispiel

An dem einfachen Modellierungsbeispiel lässt sich ermes sen, dass die Berücksichtigung temporaler Aspekte notwendig ist, jedoch auch schnell sehr komplex werden kann. Bezogen auf die zeitbehafteten Informationsobjekte lässt sich die Verständniskomplexität jedoch durch die Darstellung der zugehörigen Klasse in einem Zustandsdiagramm reduzieren.

Die konsistente Verwendung der Parallelisierungen und Synchronisationen vom (Stereo-) Typ `GForK` und `GJoin` kann bei werkzeuggestützter Modellierung ebenfalls weitestgehend automatisiert sichergestellt werden. Beispielsweise kann bei Nutzung des Meta-CASE-Werkzeuges *Generischen Modelleditor (GME)* (vgl. [Gr03]) über die Definition und anschließende Ausführung von Skripten eine diesbezügliche automatisierte Konsistenzüberprüfung der erstellten Modelle vorgenommen werden.

4 Zusammenfassung

Aus analytischen Überlegungen heraus zeigte der Beitrag am Beispiel öffentlicher Verwaltungen, welche temporale Aspekte bei der Modellierung von Geschäftsprozessen prinzipiell berücksichtigt werden müssen, und betrachtete die Periodizität von Geschäftsprozessmodellen und den Zeitbezug von Informationsobjekten näher.

Die gewonnenen Erkenntnisse sind in ihrem Kern von der Domäne der öffentlichen Verwaltungen und von der eingesetzten Modellierungssprache unabhängig und können damit sowohl auf eine andere Sprache als auch auf eine andere Domäne übertragen werden. Bei der Konzeption wurden insbesondere die Grundsätze der Sprachadäquanz und der Klarheit berücksichtigt, indem die verwendeten Konstrukte eine semantische Verankerung erfahren haben.

Zur Demonstration der Nutzenpotentiale wurden die Ergebnisse in die UML in Form einer konkreten Erweiterung (UML-Profil) implementiert. Damit wurde auch deren Praxistauglichkeit aufgezeigt.

Das vorgeschlagene UML-Profil kann in einem nächsten Schritt in einem Modellierungswerkzeug umgesetzt werden. Durch den hohen Formalisierungsgrad der notwendigen Konsistenzbedingungen kann deren Einhaltung ebenfalls im Werkzeug verankert werden. Damit reduziert sich der Aufwand für qualitätssichernde Arbeiten erheblich (Grundsatz der Wirtschaftlichkeit).

Literaturverzeichnis

- [APR02] André, C.; Peraldi-Frati, M.-A.; Rigault, J.-P.: Integrating the Synchronous Paradigm into UML: Application to Control-Dominated Systems. In: [JHC02], S. 163-178.
- [BJ02] Bider, I.; Johannesson, P.: Modeling Dynamics of Business Processes: Key for Building Next Generation of Business Information Systems. In: (Spaccapietra, S.; March, S. T.; Kambayashi, Y.) Conceptual Modeling – 2002. Proceedings of the 21th International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002. Springer, 2002, (Lecture Notes in Computer Science; Vol. 2503), S. 7-9.
- [Do99] Douglass, B. P.: Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns. Addison-Wesley, 1999. (The Addison-Wesley Object Technology Series)
- [FM02] Flake, S.; Mueller, W.: A UML Profile for Real-Time Constraints with the OCL. In: [JHC02], S. 179-195.
- [Gr03] Greiffenberg, S.: Generischer Modelleditor 2001. <http://wise.wiwi.tu-dresden.de/gme>, Download: 19.10.2003, 2003.
- [Ha87] Harel, D.: Statecharts: A visual formalism for complex systems. In: Science of Computer Programming Nr. 8, 1987, S. 231-274. <http://www.wisdom.weizmann.ac.il/~dharel/SCANNED.PAPERS/Statecharts.pdf>, Download: 10.03.2003.
- [Ho99] Hoheisel, H.: Berücksichtigung temporaler Konstrukte bei der Geschäftsprozess-Modellierung durch Geschäftsregeln. <http://www.seda.sowi.uni-bamberg.de/fg510/rundbrief/Artikel/Hohe99.pdf>, Download: 20.10.2003, 1999.
- [HP85] Harel, D.; Pnueli, A.: On the Development of Reactive Systems. In: (Apt, K. R.) Logics and Models of Concurrent Systems. Springer Verlag, 1985, S. 477-498. (NATO-ASI Series; Vol. F13)
- [HV97] ter Hofstede, A. H. M.; Verhoef, T. F.: On the feasibility of Situational Method Engineering. In: Information Systems, Vol. 22, 1997, Nr. 6/7, S. 401-422.
- [JHC02] Jézéquel, J.M.; Hussmann, H.; Cook, S.: The Unified Modeling Language: Model Engineering, Concepts, and Tools. Proceedings of the 5th International Conference "UML" 2002, Dresden, Germany, September 30 – October 4, 2002. Springer, 2002. (Lecture Notes in Computer Science; Vol. 2460)
- [Jo74] Johnson, K. L.: Grundlagen der Netzplantechnik. VDI-Verlag, 1974. (VDI-Taschenbücher; T-53)
- [KW02] Kaiser, A.; Wurglitsch, R. W.: Modellierung temporaler Aspekte betrieblicher Informationssysteme mit UML: Temporale Aspekte und das objektrelationale Modell. In: (Sinz, E. J.; Plaha, M.) Modellierung betrieblicher Informationssysteme – MobIS 2002, Proceedings zur Tagung, 10. September 2002, Nürnberg, Gesellschaft für Informatik, 2002. (Lecture Notes in Informatics; Vol. P-18)
- [KJS01] Kunii, H. S.; Jajodia, S.; Sølvberg, A.: Conceptual Modeling – 2001. Proceedings of the 20th International Conference on Conceptual Modeling, Yokohama, Japan, November 27-30, 2001. Springer, 2001. (Lecture Notes in Computer Science; Vol. 2224)
- [KS⁺03] Koubarakis, M.; Sellis, T.; Frank, A. U.; Grumbach, S.; Güting, R. H.; Jensen, C. S.; Lorentzos, N.; Manolopoulos, Y.; Nardelli, E.; Pernici, B.; Schek, H.-J.; Scholl, M.; Theodoulidis, B.; Tryfona, N.: Spatio-Temporal Databases: The CHOROCHRONOS Approach. Springer, 2003. (Lecture Notes in Computer Science; Vol. 2520)
- [Ma91] Materne, S.: MTMM: Ein System zur Verwaltung von Zeitverhältnissen. Oldenburg-Verlag, 1991. (Berichte der Gesellschaft für Mathematik und Datenverarbeitung; Nr. 193)
- [NW02] Naumenko, A.; Wegmann, A.: A Metamodel for the Unified Modeling Language. In: [JHC02], S. 2-17.

- [Oe01] Oestereich, B.: Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language. Oldenburg-Verlag, 2003.
- [OM03a] OMG (Object Management Group): Unified Modeling Language Specification, Version 1.5. <http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.zip>, Download: 30.05.2003, 2003.
- [OM03b] OMG (Object Management Group): UML Profile for Schedulability, Performance, and Time, Version 1.0. <http://www.omg.org/cgi-bin/apps/doc?formal/03-09-01.pdf>, Download: 20.10.2003, 2003.
- [OW⁺03] Oestereich, B.; Weiss, C.; Schröder, C.; Weikiens, T.; Lenhard, A.: Objektorientierte Geschäftsprozessmodellierung mit der UML. dpunkt-Verlag, 2003.
- [RBL88] Rolland, C.; Bodart, F.; Leonard, M.: Temporal Aspects in Information Systems. Proceedings of the IFIP TC 8/WG 8.1 Working Conference on Temporal Aspects in Information Systems, Sophia-Antipolis, France, 13-15 May, 1987, IFIP, Elsevier Science B.V. (North-Holland), 1988.
- [Ru99] Runzheimer, B.: Operations research: Lineare Planungsrechnung, Netzplantechnik, Simulation und Warteschlangentheorie. Gabler, 1999.
- [Sc98a] Scheer, A.-W.: ARIS-Modellierungsmethoden, Metamodelle, Anwendungen. Springer, 1998.
- [Sc98b] Schütte, R.: Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Neue betriebswirtschaftliche Forschung 233, Gabler-Verlag, 1998.
- [Sc94] Schreiber, F. A.: Is Time a Real Time? An Overview of Time Ontology in Informatics. In (Halang, W. A.; Stoyenko, A. D.) Real Time Computing. Springer Verlag, 1994, S. 283-307. (NATO-ASI Series; Vol. F127)
- [ST99] Svinterikou, M.; Theodoulidis, B.: TUML: A Method for Modeling Temporal Information Systems. In: (Jarke, M.; Oberweis, A.) Proceedings of the 11th International Conference on Advanced Information Systems Engineering, CAiSE'99, Heidelberg, Germany, June 1999. Springer, 1999, S. 456-461. (Lecture Notes in Computer Science; Vol. 1626)
- [TL91] Theodoulidis, C. I.; Loucopoulos, P.: The Time Dimension in Conceptual Modelling. In: Information Systems Vol. 16 (Nr. 3), 1991, S. 273-300.
- [TLW91] Theodoulidis, C. I.; Loucopoulos, P.; Wangler, B.: A Conceptual Modelling Formalism for Temporal Database Applications. In: Information Systems Vol. 16 (Nr. 4), 1991, S. 401-416.
- [WI97] Mertens, P.; Hasenkamp, U.; et al. (Hrsg.): Das Jahr 2000-Problem und andere Aspekte zeitorientierter IS. Schwerpunktheft: Wirtschaftsinformatik Nr. 39, 1997 (1).