

Flexible Modellierung mit Algebraischen Higher Order Netzen

Kathrin Hoffmann

Institut für Softwaretechnik und
theoretische Informatik
Technische Universität Berlin
Franklinstr. 28/29
D-10587 Berlin
hoffmann@cs.tu-berlin.de

Zusammenfassung

In diesem Papier wird die Flexibilisierung innerhalb der Prozeßmodellierung durch Algebraische Higher Order Netze vorgestellt. Durch die Erweiterung um Higher Order Konzepte wird die konkrete Ausführung von Aktivitäten vom Prozeßmodell getrennt. Damit wird ein dynamischer Austausch von Informationen mit der Umgebung unterstützt, so daß eine Modifikation der Ausführung von Aktivitäten zur Laufzeit ermöglicht wird. Die Vorteile, die sich durch den Einsatz von Algebraischen Higher Order Netze ergeben, werden anhand des Geschäftsprozesses Vertriebslogistik untersucht.

1 Einleitung

Petrinetze sind eine anerkannte Modellierungstechnik, die verteilte und parallele Abläufe komplexer Prozesse erfassen. Durch die graphische Oberfläche, der formalen Semantik und nicht zuletzt der Tool-Unterstützung werden Petrinetze erfolgreich im Bereich der Geschäftsprozesse eingesetzt [vdA98, ADM98, DG94]. Weiterhin wird die flexible Prozeßmodellierung durch unterschiedliche Konzepte unterstützt. Oft können Prozesse nicht komplett während der Modellerstellung angegeben werden, da eine Vielzahl von Alternativen zur Ausführung vorhanden sind. Manchmal können zwar die einzelnen Aktivitäten festgelegt werden, aber nicht die Reihenfolge, in der sie ausgeführt werden. In anderen Situationen steht die Art und Weise, in der eine Aktivität ausgeführt wird, erst zur Laufzeit des Prozesses zur Verfügung.

In [Val98] wird das Netz-in-Netz Paradigma unterstützt, indem eine Schnittstelle für den dynamischen Austausch von Prozeßfragmenten in das Modell integriert wird. Die Prozeßfragmente werden wiederum als Petrinetze angegeben. Durch die hierarchische Strukturierung wurde eine Wiederverwendung sowohl der statischen Modelle mit veränderten Prozeßfragmenten als auch der einzelnen Abläufe in einem anderen Kontext erreicht. Demgegenüber werden in dem Ansatz der Higher Order Object Netze (siehe [CHW97, HHSW96]) sämtliche Ressourcen vom Prozeßmodell getrennt. Ressourcen werden dabei in einem weiten

Sinne interpretiert. Neben Softwarekomponenten wird auch der Austausch von verfeinerten Prozeßmodellen zur Laufzeit unterstützt, so daß eine dynamische Modifikation der Prozesse auf unterschiedlichen Ebenen realisiert werden kann. In dem Ansatz der Higher Order Netze [LH94, Han97] wird die flexible Anpassung der Modelle ebenfalls durch die Integration zusätzlicher Informationen in den Formalismus erreicht. Dabei konzentrieren sich die Informationen, die dynamisch ausgetauscht werden, auf die Art und Weise der Ausführung von Aktivitäten. Die Idee besteht in einer Erweiterung der algebraischen Datenstruktur um Funktionsräume, in der die Alternativen der Ausführung als komplexe Algorithmen festgehalten werden. Diese werden aber erst während der Ausführung einer Aktivität konkret ausgewählt und angewendet. Hier liegt nur eine erste theoretische Fundierung vor und die Anwendung auf Geschäftsprozesse wurde bisher nicht untersucht.

Algebraische Higher Order Netze greifen den Ansatz der Higher Order Netze auf und werden in Hinblick auf die Flexibilisierung von Prozeßmodellen im Bereich der Geschäftsprozesse untersucht. Durch die Integration von Higher Order Konzepten in der Datenstruktur läßt sich die Art und Weise der Ausführung von Aktivitäten dynamisch zur Laufzeit modifizieren und an veränderte Rahmenbedingungen anpassen. Die verschiedenen Algorithmen, die zulässige Varianten der Ausführung innerhalb eines Prozeßschrittes wiedergeben, werden in der Design-Phase des Prozeßmodelles in der Datenstruktur spezifiziert. Erst zur Laufzeit werden sie als Token an die entsprechende Aktivität gebunden. Dadurch ist das Hinzufügen, Löschen und Ändern von einzelnen Algorithmen unter Beibehaltung der statischen Modelle gewährleistet.

In nächsten Abschnitt werden Algebraische Higher Order Netze eingeführt. Die Vorteile, die sich durch die Verwendung von Algebraischen Higher Order Netzen in Bezug auf die Flexibilisierung von Prozeßmodellen ergeben, werden anhand allgemeiner Abläufe erläutert und im dritten Abschnitt in der Modellierung des Geschäftsprozesses Vertriebslogistik untersucht. Nach einer Zusammenfassung wird ein kurzer Ausblick auf die geplanten Forschungsarbeiten gegeben.

2 Flexible Geschäftsprozeßmodellierung

Flexibilisierung im Bereich der Geschäftsprozesse wird auf unterschiedlichen Ebenen angegangen. Neben der flexiblen Rollenverteilung, die Verantwortlichkeiten innerhalb eines Prozesses festhält, und der flexiblen Bereitstellung von Informationen und Wissen, ist die flexible Prozeßmodellierung ein zentralen Aspekt innerhalb der Forschung [DGL00]. Flexibilisierung von Prozeßmodellen wird in Hinblick auf zusätzliche Information verstanden, die nicht vollständig während der Modellerstellung angegeben werden kann oder ständigen Veränderungen unterliegt. Hier setzen Algebraische Higher Order Netze an, indem sie Algebraischen High Level Netze [PER95], die klassische Petrinetze und algebraische Spezifikation [EM85] kombinieren, um Higher Order Konzepte [Möl87, Mei92] erweitern. Funktionen werden wie einfache Daten behandelt und sind wiederum

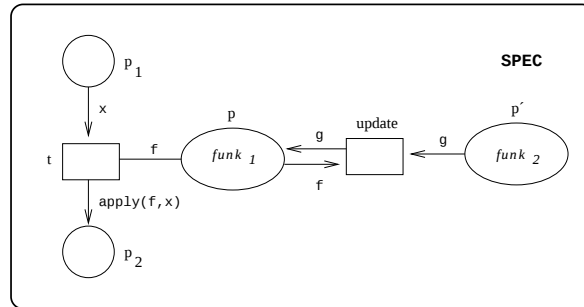


Abb. 2: Update

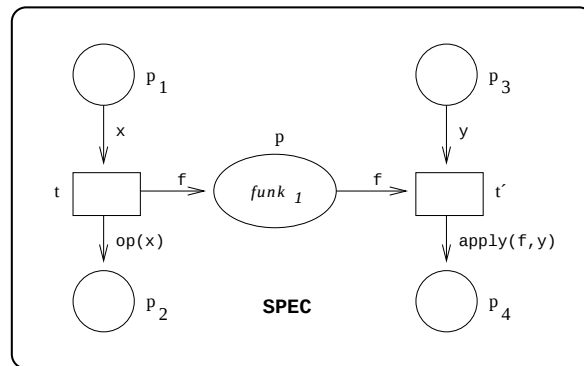


Abb. 3: Festlegung zur Laufzeit

parallelen Prozeß als eine Variante der Ausführung zur Verfügung gestellt. Dieser Aspekt der Modellierung von Abhängigkeiten zwischen einzelnen Aktivitäten ist für die spätere Reorganisation eines Geschäftsprozesses von Bedeutung, da z.B. für eine spätere Zusammenfassung einzelner Aktivitäten weiterführende Information zur Verfügung stehen. Die angegebenen Design-Patterns geben einen ersten Einblick in die flexible Modifikation, die sich durch die Trennung von Algorithmen und Prozeßmodell ergibt. Es sind weitere Design-Patterns denkbar, z.B. kann die Reihenfolge, in der die Algorithmen ausgeführt werden, variieren. Der praktische Einsatz der vorgestellten Design-Patterns wird im nächsten Abschnitt am Prozeßmodell der Vertriebslogistik untersucht.

In Abschnitt 3 wird ein wesentliches theoretisches Resultat praktisch umgesetzt, das in [Hof00] mit der Folding und Unfolding Konstruktion zwischen Algebraischen High Level Netzen und einer speziellen Klasse von Algebraischen Higher Order Netzen erzielt wurde. Der Grundgedanke ist die Faltung eines Algebraischen High Level Netzes an bestimmten Patterns bestehend aus parallelen Prozeßschritten. Durch die Patterns wird die unterschiedliche Verarbeitung von Daten mit typgleichen Operationen modelliert. Sie werden zu einer funktionalen

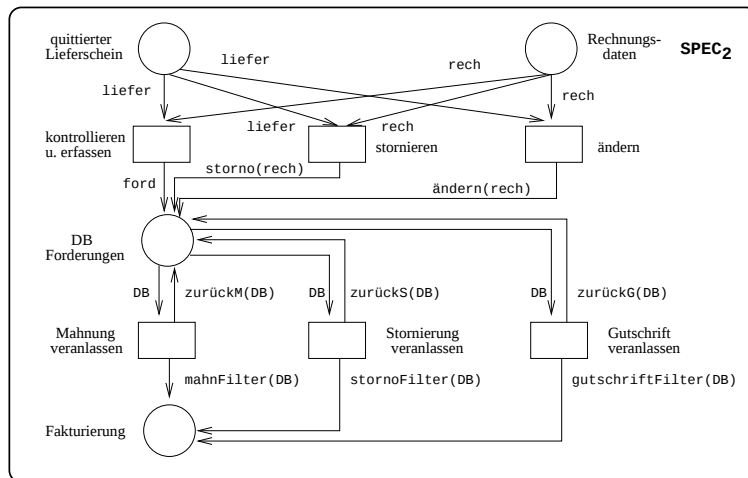


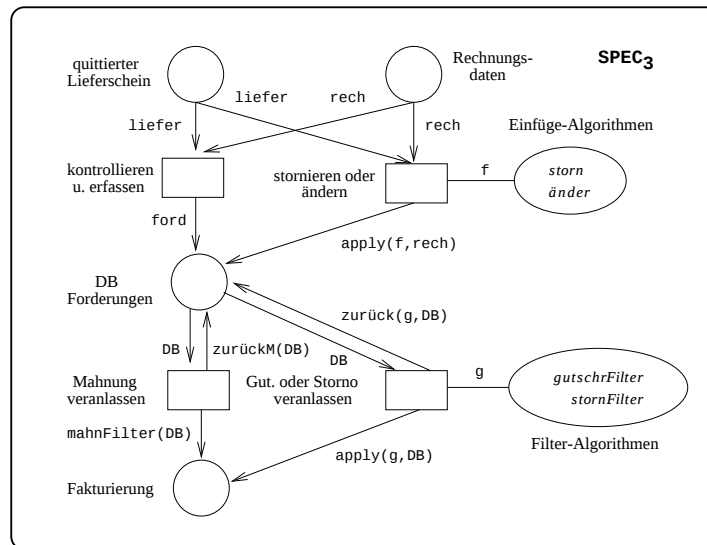
Abb. 4: Behandlung von Abweichungen

wicklung und Debitorenbuchhaltung sind in der folgenden Tabelle zusammengestellt:

Lieferung	Auftragsabwicklung	Debitorenbuchhaltung
Waren angenommen	Rechnungsdaten übergeben	ggf. Mahnung veranlassen
Waren storniert	Rechnungsdaten stornieren	Stornierung veranlassen
Waren beanstandet	Rechnungsdaten ändern	Gutschrift veranlassen
Waren abgelehnt	Rechnungsdaten stornieren	Gutschrift veranlassen

Aktivitäten der Organisationseinheiten

Die Abweichungen sind in Abbildung 4 modelliert und führen zu einem Aufblähen des Netzes. Auffällig am verfeinerten Modell ist die Übereinstimmung der Parameter, welche innerhalb paralleler Prozessschritte angewendet werden. Es lassen sich zwei verschiedene Patterns ausmachen. Zum einen werden mit Hilfe der quittierten Lieferscheine und der Rechnungsdaten modifizierte Rechnungsdaten (storno(rech) bzw. ändern(rech)) in die Datenbank der De-



bitorenbuchhaltung übergeben. Zum anderen werden anhand dieser Datenbank die Forderungen auf verschiedene Vermerke hin gefiltert (`stornoFilter(DB)` bzw. `gutschriftFilter(DB)`) und die modifizierte Liste der Forderungen zurückgelegt. Damit lassen sich diese parallelen Verarbeitungsschritte jeweils über das Konzept des Foldings (siehe 2. Abschnitt) zusammenfassen, indem von der konkreten Ausführung abstrahiert wird. Das Resultat des Foldings ist das semantisch äquivalente Algebraische Higher Order Netz in Abbildung 5. Die Algorithmen sind jetzt nicht mehr als Operationen in der statischen Netzstruktur fixiert, sondern werden als Token der Stellen Einfüge-Algorithmen bzw. Filter-Algorithmen angegeben. Sollten zu einem späteren Zeitpunkt (z.B. durch die Differenzierung zwischen der vollständigen und der teilweisen Ablehnung der Waren) weitere Abweichungen auftreten, so spiegelt sich diese Veränderung im Modell durch das Hinzufügen entsprechender Token unter Beibehaltung der Netzstruktur wider.

Durch die Gegenüberstellung der Abweichungen (siehe Tabelle) läßt sich ein direkter Zusammenhang zwischen den Aktivitäten innerhalb der Prozeßschritte stornieren oder ändern und Gutschrift oder Storno veranlassen erkennen. Die Art der Lieferung legt die nachfolgenden Aktivitäten in der Auftragsabwicklung und Debitorenbuchhaltung fest. Diese Informationen werden im weiteren für die Reorganisation des Geschäftsprozesses explizit im Modell angegeben. Die Reorganisation umfaßt die teil-automatisierte Weiterleitung der entsprechenden Kundendaten an die Fakturierung. Das Eintragen einer Stornierung, Beanstandung oder Ablehnung in die Datenbank der Forderungen soll direkt eine Filterung der Kundendaten anstoßen. Der reorganisierte Ablauf ist in Abbildung 6 dargestellt. Der Zusammenhang zwischen den Ausführungen der beiden Prozeßschritte ist über eine zusätzliche Transition Abweichungen behandeln

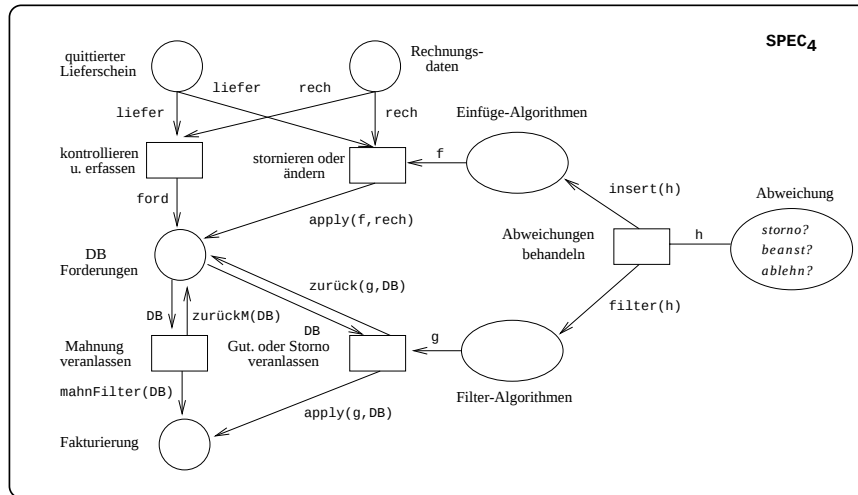


Abb. 6: Einfüge- bzw. Filter-Algorithmus zur Laufzeit

modelliert. Die einzelnen Varianten der Abweichungen finden sich als Token der Stelle Abweichung wieder. Der konkrete Algorithmus, der jeweils in den beiden Prozessschritten ausgeführt wird, wird erst zur Laufzeit durch die Auswahl einer Abweichung festgelegt. Dabei wird über die Higher Order Funktionen *insert* bzw. *filter*, die als Eingabeparameter die Art der Abweichung (*storno?*, *beanst?* bzw. *ablehn?*) erhalten, die entsprechenden Einfüge- bzw. Filter-Algorithmus zur Verfügung gestellt. Wird z.B. eine Lieferung abgelehnt, so wird einerseits der entsprechende Algorithmus zum Stornieren der Rechnungsdaten und andererseits der entsprechende Filteralgorithmus für Gutschriften generiert.

Damit wurden einige Vorteile vorgestellt, die sich durch den Einsatz von Algebraischen Higher Order Netzen und der Verwendung der Design-Patterns aus Abschnitt 2 in der Geschäftsprozeßmodellierung ergeben. Einerseits wurde die Auswahl zwischen unterschiedlichen Algorithmen in Abbildung 5 durch die Zusammenfassung der Abweichungen erreicht. Andererseits wurde in Abbildung 6 die Art und Weise, in der die Daten innerhalb der Aktivitäten verarbeitet werden, erst durch eine weitere Aktivität zur Laufzeit festgelegt und damit gleichzeitig ein Zusammenhang zwischen den beiden Prozessschritten hergestellt. Die Anwendung von weiteren Design-Patterns im Geschäftsprozeß Vertriebslogistik wird zu einem späteren Zeitpunkt im Rahmen einer Fallstudie untersucht.

4 Zusammenfassung und Ausblick

In diesem Papier wurden die Vorteile im Rahmen der Flexibilisierung durch den Einsatz von Algebraischen Higher Order Netzen vorgestellt und am Geschäftsprozeß Vertriebslogistik praktisch nachvollzogen. Dabei wurde auf der Grundlage von [Sch94] ein Prozeßmodell entwickelt, das den verteilten Ablauf der Aktivitäten in den Organisationseinheiten Auftragsannahme, Auftragsabwicklung und Debitorenbuchhaltung wiedergibt. Eine weitere Organisationseinheit, die An-

fragen von Kunden entgegennimmt und entsprechende Angebote erstellt, ist zu einem späteren Zeitpunkt in das Prozeßmodell zu integrieren. Diese Prozeßerweiterung soll durch formal fundierte Strukturierungstechniken unterstützt werden, um eine konsistente Modellentwicklung sicherzustellen. Dazu werden die für Algebraische High Level Netze existierenden Strukturierungstechniken [PGE98], die sowohl die Komposition als auch die Modelltransformation basierend auf Graphtransformation umfassen, auf Algebraische Higher Order Netze übertragen.

Literatur

- [ADM98] A. Agostini and G. De Michaelis. Simple Workflow Models. Workflow Management: Net-based Concepts, Models, Techniques and Tools. In *Workshop within the 19th International Conference on Applications and Theory of Petri Nets*, Lisbon, 1998.
- [CHW97] I. Claßen, Y. Han, and H. Weber. Towards Evolutionary and Adaptive Workflow Systems - Infrastructure Support Based on Higher Order Object Nets. In *Proc. of EDOC 97*, Australia, 1997.
- [DG94] W. Deiters and V. Gruhn. The FUNSOFT Net Approach to Software Process Management. *International Journal on Software Engineering and Knowledge Engineering*, 4(2):229–256, 1994.
- [DGL00] W. Deiters, T. Goesmann, and T. Löffeler. Flexibility in workflow management dimensions and solutions. *International Journal of Computer Science and Engineering*, 15(5):303–314, 2000.
- [EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer, Berlin, 1985.
- [Han97] Yanbo Han. *Software Infrastructure for Configurable Workflow System - A Model-Driven Approach Based on Higher-Order Nets and CORBA*. PhD thesis, Technical University of Berlin, 1997.
- [HHSW96] Y. Han, J. Himmighöfer, T. Schaaf, and D. Wikarski. Management of Workflow Ressources to Support Adaptability and System Evolution. In *Workshop on Adaptive Workflow, Proc. of the 1st. Int. Conf. on Practical Aspects of Knowledge Management*, volume 1, 1996.
- [Hof00] K. Hoffmann. Runtime Modifikation between Algebraic High Level Nets and Algebraic Higher Order Nets using Folding and Unfolding Construction. In G. Hommel, editor, *Workshop Communication Based Systems within the European Joint Conference on Theory and Practice of Software (ETAPS 2000)*, pages 131–146. Kluwer, 2000.
- [LH94] M. Löwe and J. Han. Higher Order Nets. Technical Report 94-34, Technical University Berlin, 1994.
- [Mei92] K. Meinke. Universal algebra in higher types. *Theoretical Computer Science*, (100):385–417, 1992.
- [Möl87] B. Möller. *Higher-Order algebraic specifications*. Fakultät für Mathematik und Informatik, Technische Universität München, 1987.

- [PER95] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science*, 5:217–256, 1995.
- [PGE98] J. Padberg, M. Gajewsky, and C. Ermel. Rule-Based Refinement of High-Level Nets Preserving Safety Properties. In E. Astesiano, editor, *Fundamental Approaches to Software Engineering*, pages 221–238. Springer Verlag, Lecture Notes in Computer Science 1382, 1998.
- [Sch94] A.-W. Scheer. *Business Process Engineering, Reference Models for Industrial Enterprises*. Springer-Verlag, Berlin, 1994.
- [Val98] R. Valk. Petri Nets as Token Objects: An Introduction to Elementary Object Nets. In J. Desel and M. Silva, editors, *Application and Theory of Petri Nets*, pages 1–25. Springer-Verlag, 1998.
- [vdA98] W.M.P. van der Aalst. The application of Petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

6 Anhang

