In Proceedings des 13. Workshops “Algorithmen und Werkzeuge für Petrinetze” (AWPN'06). 25.-26. September 2006, Hamburg. Universität Hamburg, 2006.

Ein Framework zur Modellierung und Analyse von XML-Netzen

Stefanie Betz, Thomas Karle, Stefan Klink, Agnes Koschmider, Yu Li, Marco Mevius,
Andreas Oberweis, Daniel Ried, Ralf Trunko, Markus Zaich

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
Universität Karlsruhe (TH), 76187 Karlsruhe
{betz, karle, klink, koschmider, li, mevius, oberweis, ried, trunko, zaich}@aifb.uni-karlsruhe.de

1 Einleitung

Um den Herausforderungen der globalisierten Märkte angemessen begegnen zu können, ist bei den Unternehmen eine Neugestaltung der Geschäftsprozesse notwendig, insbesondere da diese Geschäftsprozesse zunehmend überbetrieblichen Charakter besitzen. Bei der Ausführung überbetrieblicher Geschäftsprozesse werden Prozessobjekte elektronisch zwischen den kooperierenden Unternehmen ausgetauscht. Dies erfordert eine präzise, integrierte Beschreibung der Geschäftsprozesse und der damit verbundenen relevanten Prozessobjekte. In diesem Zusammenhang sind Prozessobjekte beispielsweise Dokumente (etwa Bestellungen, Rechnungen, Verträge, Protokolle), die zwischen den kooperierenden Unternehmen ausgetauscht oder bei der Prozessdurchführung gelesen bzw. bearbeitet werden [1]. Diese auszutauschenden und zu bearbeitenden Dokumente werden zunehmend im XML-Format erstellt, da dieses Format plattformunabhängig und in beliebiger Präsentationsform darstellbar ist. Hinzu kommt, dass XML-Dokumente als reine Textdateien schnell und einfach übertragbar und maschinenverständlich sind.

Für die effiziente Geschäftsprozessausführung ist die Integration von elektronischem Dokumentenaustausch mit überbetrieblichem Prozessmanagement erforderlich. Petri-Netze haben sich in der Vergangenheit zur einfachen und anschaulichen, zugleich aber auch formalen und ausdrucksstarken Modellierung von Geschäftsprozessen als geeignet erwiesen. In [1] werden im Zusammenhang mit der Anwendungsdomäne E-Business die so genannten XML-Netze, eine Variante höherer Petri-Netze, eingeführt. XML-Netze, welche aus der Idee der SGML-Netze [2] entstanden sind, unterstützen die Modellierung von überbetrieblichen Geschäftsprozessen basierend auf XML-Dokumenten.

Zur Modellierung, Simulation und Analyse von Petri-Netzen wurden viele graphische Werkzeuge[1] und Werkzeuge ohne graphische Benutzerunterstützung[2] vorgeschlagen. Im Bereich der Petri-Netz-basierten Workflow-Modellierung kann beispielsweise Woflan[3] verwendet werden. Es gibt auch einige Petri-Netz-Werkzeuge, die XML-basierte Dateiformate verwenden[4]. Eine werkzeugunabhängige Repräsentation von Petri-Netzen ist mit der Petri Net Markup Language (PNML) [3] möglich. PNML ist ein XML-basiertes Austauschformat für Petri-Netze. Allerdings ist zurzeit noch kein Framework verfügbar, das die Verwendung von XML-Netzen unterstützt. Der vorliegende Beitrag stellt INCOME2010[5] vor – ein Framework zur Modellierung und Analyse von XML-Netzen.

---

1 *Design/CPU*: http://www.daimi.au.dk/designCPN/ oder *PEP*: http://theoretica.informatik.uni-oldenburg.de/~pep/

2 *INA*: http://www.informatik.hu-berlin.de/lehrstuehle/automaten/ina/ oder
LoLA: http://www.informatik.hu-berlin.de/~kschmidt/lola.html

3 http://is.tm.tue.nl/research/woflan.htm

4 *Renew*: http://www.renew.de/ oder [4]

5 INCOME2010 ist auf Basis von XML-Netzen eine Weiterentwicklung des Werkzeuges INCOME [5], welches in seinen Grundkonzepten seit etwa 20 Jahren für die Modellierung und Analyse von Geschäftsprozessen verwendet wird.

Der Beitrag ist wie folgt aufgebaut: Abschnitt 2 erläutert kurz die Grundlagen von XML-Netzen. In Abschnitt 3 werden Anforderungen an ein Framework, das XML-Netze unterstützt, definiert, während in Abschnitt 4 die Architektur eines solchen Frameworks beschrieben wird. In Abschnitt 5 werden Implementierungsaspekte des Frameworks erläutert. Den Abschluss bildet Abschnitt 6 mit einer Zusammenfassung sowie einem Ausblick auf weiteren Forschungsbedarf.

2 Grundlagen

Petri-Netze [6] sind eine formale Beschreibungssprache mit einer grafischen Repräsentation und einer mathematischen Fundierung. Varianten von Petri-Netzen lassen sich in die Klassen der elementaren bzw. höheren Petri-Netze einordnen. XML-Netze stellen eine Variante der höheren Petri-Netze (und hier insbesondere der Prädikate-/Transitionen-Netze [7]) dar.

In einem XML-Netz wird jede Stelle durch ein XML-Schema-Diagramm beschriftet, wobei das XML-Schema den sog. *Stellentyp* repräsentiert. Stellen werden daher als Container für XML-Dokumente, welche dem jeweiligen Stellentyp genügen, interpretiert. XML-Dokumente stellen hier die prozessrelevanten Datenobjekte dar. Die Ausführung der Aktivitäten, d.h. das Schalten der Transitionen, wird durch den Fluss der XML-Dokumente definiert. Transitionen können mit einem prädikatenlogischen Ausdruck beschriftet werden, dessen freie Variablen in den Inschriften der adjazenten Kanten, den sog. *Filterschema-Diagrammen*, enthalten sein müssen. Filterschema-Diagramme repräsentieren Lese- oder Manipulationsfilter für die jeweiligen Zugriffsarten Lesen bzw. Ändern, Erzeugen oder Einfügen und Löschen eines oder mehrere Datenobjekte. Beim Schalten einer Transition werden für eine existierende Variablenbelegung XML-Dokumente aus den Markierungen der Vorbereichsstellen gelesen oder gelöscht und in die Markierungen der Nachbereichsstellen eingefügt bzw. bereits existierende XML-Dokumente werden geändert.

3 Anforderungen

Anforderungen an die Funktionalität

Ein Framework für die Modellierung und Analyse von XML-Netzen muss einerseits Funktionalität für die Modellierung von Abläufen und andererseits Funktionalität für die Modellierung von Prozessobjekten wie beispielsweise Bestellungen oder Rechnungen bereitstellen. Für die Beschreibung von Geschäftsprozessen muss das Framework darüber hinaus die Möglichkeit bieten, die Abläufe mit den Geschäftsobjekten zu verknüpfen. Für die Modellierung sollen grafische Editoren bereitgestellt werden, um eine komfortable Bedienung zu ermöglichen. Die einzelnen Elemente sollten per Drag-and-Drop-Funktion angeordnet werden können. Detailinformationen zu den Elementen der Abläufe, d.h. der Transitionen, Stellen und Kanten sollen per Mausklick in einem separaten Fenster editierbar angezeigt werden. Die Struktur der Geschäftsobjekte soll mit XML-Schema definiert werden können. Die Definition der Filterschemata soll ebenfalls über XML-Schema möglich sein.

Für die Modellierung auf Basis von XML-Netzen sind für die Operationen funktionale Anforderungen, welche über einfache Stellen-/Transitions-Netze hinausgehen, zu berücksichtigen. Folgende Funktionalität wird durch die XML-Anfragesprache XQuery [8] realisiert.

(i) Funktionalität bzgl. der Filterschemata an den Kanten:

- Löschen oder Erzeugen eines vollständigen XML-Dokuments,
- Hinzufügen von Elementen oder Attributen,
- Löschen von Elementen oder Attributen,
- Strukturelle Veränderung der XML-Dokumente,

- Umbenennen von Elementen oder Attributen,
- Verändern der Element-Belegung,
- Kombination mehrerer Dokumente mit unterschiedlichen XML-Schemata,
- Einfügen von Elementen mit absoluter Stellenangabe (z.B. am Anfang, am Ende oder an einer festgelegten Stelle) oder relativer Stellenangabe (z.B. hinter einem Element mit bestimmtem Inhalt),
- Selektion von XML-Dokumenten mit bestimmter Struktur (Anordnung der Elemente, Elementtypen),
- Selektion von XML-Dokumenten mit bestimmtem Inhalt (Element- bzw. Attribut-Belegung).

(ii) Funktionalität bzgl. der Transitionsinschriften:

- Schachtelung von Anfragen,
- Sortierung,
- Anwendung von Funktionen,
- Benutzung logischer Ausdrücke mit Operatoren,
- Textanfragen über Elementgrenzen hinaus.

Technische Anforderungen

Das Framework sollte komponentenbasiert realisiert werden, um eine einfache Erweiterung zu ermöglichen. Für die persistente Speicherung der Prozessmodelle sollte eine generische Schnittstelle für verschiedene Datenbank- bzw. Dateisysteme bereitgestellt werden.

4 Architektur

Das Framework INCOME2010 ist mit dem Ziel der Wiederverwendbarkeit und Erweiterbarkeit komponentenorientiert aufgebaut und besteht aus einem *Modeler*, einem *ObjectSchemaEditor* und einem *Analysator* (vgl. Abb. 1). Der Modeler dient zur graphischen Erstellung von Geschäftsprozess- und Organisationsmodellen. Der ObjectSchemaEditor ist eine XML-Netz-spezifische Komponente zum Erstellen und Editieren von XML-Netz-Objekten, wie XML-Schemata, Filterschemata und Transitionsinschriften. Der Analysator besteht aus einem *Verhaltensanalysator* und einem *Strukturanalysator*. Während der Verhaltensanalysator Prozessmodelle visuell simuliert und validiert, überprüft der Strukturanalysator strukturelle Eigenschaften der Petri-Netz-Modelle. Durch einheitliche Schnittstellen und Erweiterung der hier dargestellten Basiskomponenten können weitere Komponenten in das Framework eingebaut bzw. externe Werkzeuge oder Module integriert werden.

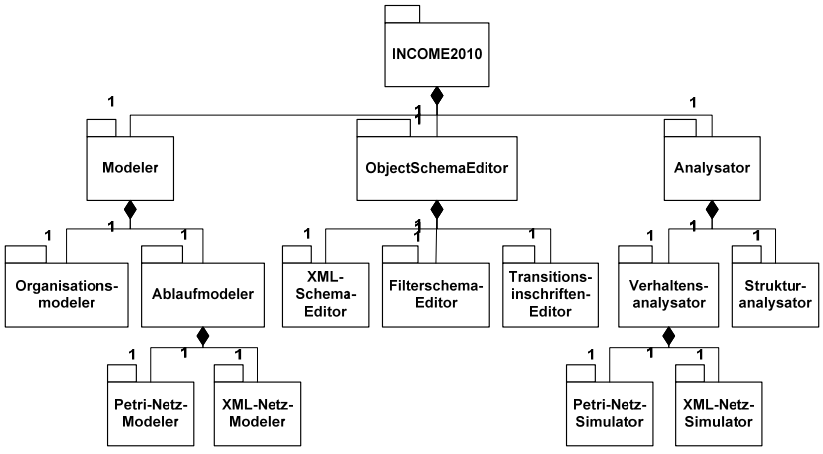

Abb. 1. UML-basiertes Komponentendiagramm für die Architektur von INCOME2010

Aus objektorientierter Sicht basiert das Framework auf dem Entwurfsmuster *Model-View-Controller* (MVC) [9], einem Architekturmuster für interaktive Softwaresysteme, wobei die Systeme streng in Datenmodel (Model), Präsentation (View) und Systemsteuerung (Controller) aufgeteilt sind (vgl. Abb. 2). Die View präsentiert dem Benutzer eine graphische Editoroberfläche (*EditorGui*) und eine Sicht auf die Objektdaten der Petri-Netz-, XML-Netz-, XML-Schema- und Filterschema-Element-Modelle. Die vom Benutzer aktivierten Ereignisse werden durch die Schnittstelle *Actions* abgefangen, welche daraufhin die entsprechenden Handler des Controllers aufrufen. Je nach Benutzerereignis modifiziert der Controller entweder die EditorGui direkt, oder er greift auf die Schnittstelle *ModelFactories & -Containers* des Models zu, um neue Modelle zu erstellen bzw. um existierende Modelle zu modifizieren. Nach der Erstellung bzw. Modifizierung der Modelle werden die entsprechenden Element-Views über die Schnittstelle *ViewFactories* erzeugt bzw. aktualisiert.

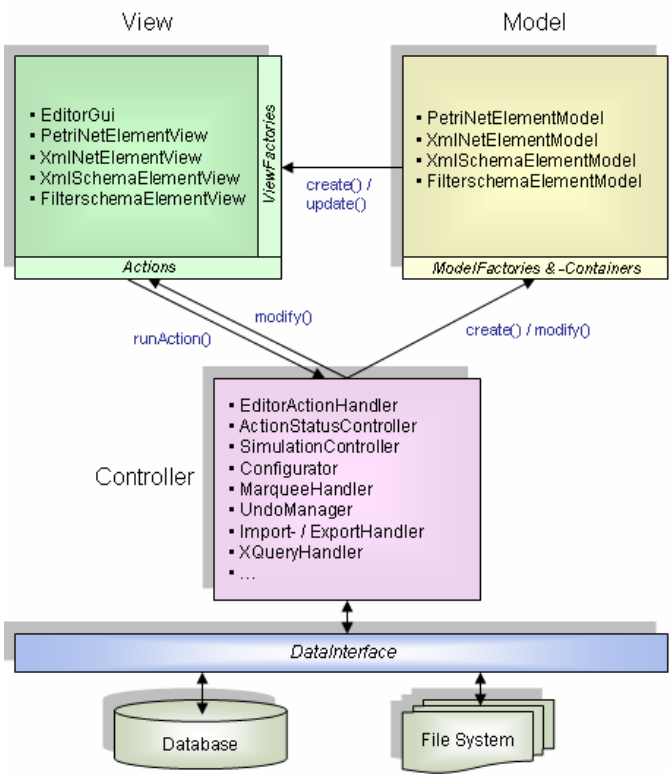

Abb. 2. Architektur von INCOME2010 nach dem MVC-Entwurfsmuster

Die Datenzugriffe erfolgen ausschließlich durch den Controller über das *DataInterface*, welches das Framework von konkreten Datenbank- bzw. Dateisystemen entkoppelt. Hierfür ist eine einheitliche und von der Datenquelle unabhängige Anfragesprache zu entwickeln, die gemäß konkreten und eingesetzten Datenbank- oder Dateisystemen durch das DataInterface geparsed wird. Aufgrund dessen ist das Framework unabhängig von den darunter liegenden Datenquellen und kann flexibel auf verschiedenen Datenhaltungssystemen aufgebaut werden.

5 Implementierung und Features

Das Framework wird zwecks Plattformunabhängigkeit komplett in Java entwickelt. Die in Abb. 3 dargestellte GUI basiert auf den SWT- und JFace-Toolkits[6] von Eclipse und der Graphen-

[6] SWT (Standard Widgets Toolkit) und JFace sind Java-Bibliotheken zur Erstellung graphischer Benutzeroberfläche (GUI). http://www.eclipse.org/swt.

bibliothek JGraph[7]. Anfragen auf XML-Schemata werden durch den *XQueryHandler* als XQuery-Anfrage geparsed und daraufhin mit Hilfe von Saxon[8] ausgeführt. Zum Speichern und zum Austausch von XML-Netz-Modellen wurde ein XML-basiertes Austauschformat *XNML* als Erweiterung von PNML entwickelt, in dem neben den graphischen Informationen der Netzelemente (Position, Größe, Farbe etc.) auch XML-Netz-spezifische Informationen wie beispielsweise Transitionsinschriften und Pfade der Schemata gespeichert werden.

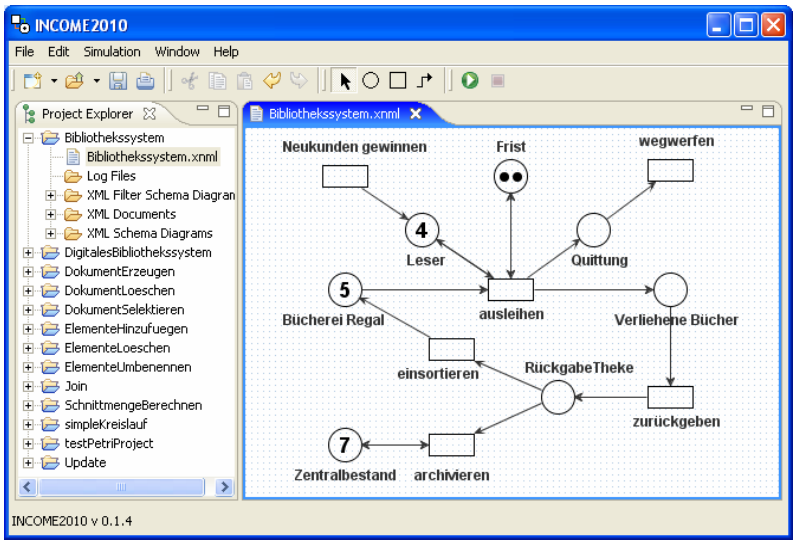

Abb. 3. Graphische Oberfläche von INCOME2010

Damit das Framework einfach und effizient erweitert werden kann, wurde eine generische Klassenstruktur für elementare Petri-Netze definiert, aus der die Strukturen weiterer Petri-Netz-Varianten (z.B. XML-Netze) abgeleitet werden können (vgl. Abb. 4). Die Datenmodelle der Stellen, Transitionen, Kanten und Marken implementieren jeweils die Schnittstellen *IPetriPlace*, *IPetriTransition*, *IPetriArc* und *IPetriToken*, die wiederum Erweiterungen der Schnittstelle *IPetriObject* darstellen. Die gemeinsamen Eigenschaften von Stellen und Transitionen werden zu *PetriNetModelElement* als Implementierung von IPetriObject aggregiert. Die Petri-Netz-Element-Modelle werden durch Element-Modelle von XML-Netzen als Subklassen erweitert, können aber auch durch Element-Modelle anderer Petri-Netz-Varianten erweitert werden.

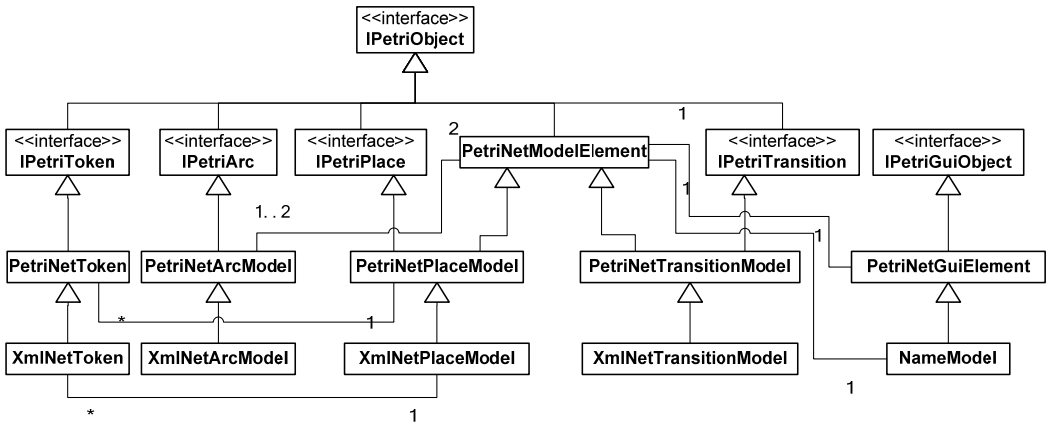

Abb. 4. Klassenstruktur von Petri-Netzen und XML-Netzen

[7] JGraph ist eine Java-Bibliothek zur Visualisierung von Graphen. http://www.jgraph.com/.

[8] Saxon ist ein XSLT- und XQuery-Prozessor. http://saxon.sourceforge.net/.

Um den Erweiterungsaufwand möglichst gering zu halten, werden die graphischen Daten von PetriNetModelElement getrennt in *PetriNetGuiElement* gespeichert, welches die Schnittstelle *IPetriGuiObject* implementiert und durch *NameModel* erweitert wird. PetriNetGuiElement und PetriNetModelElement registrieren sich gegenseitig als private Felder, damit sie ohne Probleme auf einander zugreifen können. PetriNetGuiElement liefert den Element-Views benötigte und GUI-relevante Informationen und bildet an dieser Stelle eine Schnittstelle zwischen den Datenmodellen und ihren Views. Die Klasse PetriNetGuiElement lässt sich bei Bedarf erweitern, um weitere Anforderungen bzgl. der Darstellung spezifischer Petri-Netz-Varianten zu erfüllen.

Die Entwicklung des Frameworks befindet sich noch im Anfangsstadium. Folgende Features wurden jedoch bereits implementiert:

- Graphische Modellierung von elementaren Petri-Netzen und XML-Netzen.
- Einfache Simulation von Petri-Netz- und XML-Netz-Modellen mit Ergebnisbericht.
- Import/Export von Netz-Modellen im PNML- bzw. XNML-Format.
- Graphischer Export von Netz-Modellen in die Formate BMP-, GIF-, JPEG- oder PNG.
- Erstellung von Projekten mit dem Projektexplorer.
- Unterstützung mehrer natürlicher Sprachen.

Im Rahmen der weiterführenden Projektarbeit ist geplant, folgende Features bzw. Module zu implementieren:

- ObjectSchemaEditor
- Organisationsmodeler
- Generischer Strukturanalysator
- DataInterface

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Framework zur Modellierung und Analyse von XML-Netzen vorgestellt. Die Kombination der formal fundierten Petri-Netze mit dem plattformunabhängigen XML-Format zu XML-Netzen bietet eine Unterstützung der Modellierung überbetrieblicher Geschäftsprozesse basierend auf XML-Dokumenten.

Das hier vorgestellte Framework zur Modellierung und Analyse von XML-Netzen bietet die Möglichkeit, komplexe Abläufe und Prozessobjekte zu modellieren.

Die auf dem Entwurfsmuster Model-View-Controller basierende Architektur des Frameworks ist komponentenorientiert aufgebaut und besteht aus dem Modeler, dem ObjektSchemaEditor und dem Analysator. Die Aufteilung in Komponenten ermöglicht eine einfache Erweiterung des Frameworks. Außerdem erfolgt durch den Einsatz des DataInterface als generische Schnittstelle eine Entkoppelung des konkreten Datenbank- und Dateisystem-Zugriffs. Dadurch kann das Framework flexibel auf den jeweiligen (Datenhaltungs-)Systemen aufgesetzt werden. Zusätzlich wurde eine generische Klassenstruktur für elementare Petri-Netze definiert. Die grafische Modellierung der Petri-Netz-Elemente wird mit Hilfe eines Editors ermöglicht, welcher einfach zu bedienen ist und über intuitive Drag-and-Drop-Funktionalitäten verfügt. Die Simulation von elementaren Petri-Netz- und XML-Netz-Modellen inklusive eines Ergebnisberichtes ist ebenso schon implementiert, wie der Import und Export der Netz-Modelle in das PNML- bzw. XNML-Format.

Für die Zukunft ist es geplant, die noch nicht umgesetzten Features bzw. Module zu implementieren, wobei sich insbesondere interessante Fragestellungen im Bereich der Spezifizierung der Transitionsinschriften ergeben, beispielsweise die der Umsetzung der Schachtelung von Anfragen. Des Weiteren sollen verschiedene andere Werkzeuge, welche in den unterschiedlichen Projektgruppen des Instituts AIFB im Zusammenhang mit Petri-Netzen entwickelt wurden, in das Framework eingebunden werden, z.B. das Werkzeug zur semantischen Annotation von Petri-Netzen [10].

Literatur

1. K. Lenz: Modellierung und Ausführung von E-Business Prozessen mit XML-Netzen, Frankfurt, 2003.
2. W. Weitz: Integrierte Dokumenten- und Ablaufmodellierung im Electronic Commerce, Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Karlsruhe, Shaker Verlag, Aachen, 2000.
3. M. Jüngel, E. Kindler, M. Weber: Towards a Generic Interchange Format for Petri Nets. In: R. Bastide, J. Billington, E. Kindler, F. Kordon, and K. H. Mortensen (eds.): Meeting on XML/SGML based Interchange Formats for Petri Nets. pp. 1-5, Århus, Denmark, 21st ICATPN. June 2000.
4. R. B. Lyngsø, T. Mailund: Textual Interchange Format for High-Level Petri Nets. In: Jensen, K. (Hrsg.): Workshop on Practical Use of Coloured Petri Nets and Design/CPN, Århus University, Denmark, Juni 1998 (Daimi PB 532), S. 47-64 (S. 29).
5. A. Oberweis, G. Scherrer, W. Stucky: INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems, Information Systems, Vol. 19, No. 8, S. 643-660, 1994.
6. W. Reisig: Petrinetze - Eine Einführung. Studienreihe Informatik. Springer. 2. Auflage, 1986.
7. H. J. Genrich, K. Lautenbach: “System modelling with high level petri nets”. Theoretical Computer Science, (13):109-136, 1981.
8. H. Katz, D. Chamberlin, D. Draper, M. Fernandez, M. Kay, J. Robie, M. Rys, J. Simeon, J. Tivy, P. Wadler: XQuery from the Experts: A Guide to the W3C XML Query Language. Addison-Wesley Professional, September, 2003.
9. E. Glenn, S. Krasner, T. Pope: A Cookbook for using the Model-View-Controller Userinterface Paradigm in Smalltalk-80, ParcPlace Systems, 1988.
10. A. Koschmider, A. Oberweis: Ontology based Business Process Description**.** In J. Castro; E. Teniente, Proceedings of the CAiSE´05 WORKSHOPS, no. 2, pp. 321-333. Porto/Portugal, June 2005.