

Modellierung und Simulation biologischer Prozesse mit diskreten Modellierungssprachen: ein MDE-Ansatz

Claudia Täubner, Silke Eckstein

tauebner@ifis.cs.tu-bs.de, eckstein@ifis.cs.tu-bs.de

Abstract: Signaltransduktionswege beschreiben, wie Zellen auf extrazelluläre Signale reagieren, die von Rezeptoren in der Zellmembran empfangen und in den Zellkern weitergeleitet werden. In diesem Beitrag stellen wir das System Pathway Modeler vor, das es ermöglicht, Modelle für Signaltransduktionswege in verschiedenen Modellierungssprachen zu generieren und mit den zugehörigen Simulationstools zu simulieren. Dadurch ist es möglich, Signaltransduktionswege aus unterschiedlichen Perspektiven zu visualisieren und qualitativ zu analysieren. Zur Erzeugung der Modelle wird ein MDE-Ansatz verfolgt.

1 Einleitung

Signaltransduktionswege (signal transduction pathways, Pathways) beschreiben, wie Signale, die an der Zellmembran von Rezeptoren empfangen werden, im Inneren der Zelle mittels biochemischer Reaktionen weitergeleitet werden. Typischerweise werden diese Signale in den Zellkern weitergeleitet, wo sie für eine Änderung des gerade ablaufenden genetischen Programms und somit für eine Reaktion der Zelle auf das empfangene Signal sorgen. Diesen Informationsfluss innerhalb der Zelle zu verstehen ist notwendig, um ein tieferes Verständnis für das Funktionieren der Zelle als Ganzes zu erlangen. Hier kann die Informatik die Systembiologie mit ihren Erfahrungen, Methoden und Werkzeugen zur Modellierung und Simulation nebenläufiger reaktiver Systeme unterstützen: Der Signalfuss in komplexen Netzwerken kann analysiert und Hypothesen zunächst durch Simulation eines Modells überprüft werden, bevor sie im Labor mit Experimenten validiert werden. Auch die Überprüfung der zu einem bestimmten Pathway zusammengetragenen Daten hinsichtlich ihrer Konsistenz wird durch einen solchen Ansatz unterstützt.

In der Biologie ist man seit langem damit beschäftigt, das Zusammenspiel einzelner Moleküle innerhalb der Zelle zu untersuchen. Seit einiger Zeit hat man damit begonnen, dieses Wissen zu größeren Einheiten zusammenzusetzen und in Datenbanken abzulegen. Unter anderem wird dabei der Informationsfluss innerhalb der Zelle durch Signaltransduktionswege beschrieben, die letztlich eine Folge von biochemischen Reaktionen darstellen. Erst langsam ist die Datenbasis im Bereich der Signalweiterleitung innerhalb von Zellen groß genug, um die Daten über einzelne biochemische Reaktionen zu Pathways zusammenzusetzen, diese mittels diskreter Modellierungssprachen zu repräsentieren und mit Hilfe von Simulationen das Systemverständnis weiter zu vergrößern. Entsprechend ist auch die auto-

matische Generierung von Modellen aus den entsprechenden Datenbanken noch nicht sehr weit fortgeschritten. Diese Ausgangssituation beschreiben wir genauer in den Abschnitten 2 und 6. In diesem Beitrag präsentieren wir einen systematischen Ansatz, der es erlaubt, Signaltransduktionswege, die in Pathway-Datenbanken abgelegt sind, automatisch in Modelle diskreter Modellierungssprachen zu transformieren. Dadurch ist es möglich, Signaltransduktionswege aus unterschiedlichen Perspektiven zu visualisieren und mit Hilfe von Simulationswerkzeugen qualitativ zu analysieren:

- Um die Abbildung von Signaltransduktionswegen auf diskrete Modellierungssprachen verständlich zu machen, erklären wir in Abschnitt 2, wie man sich den Informationsfluss innerhalb der Zelle vorzustellen hat.
- Wir haben ein erweiterbares System entwickelt, um für Pathways, die in einer Pathway-Datenbank abgelegt sind, Modelle in verschiedenen Modellierungssprachen zu generieren und diese zu simulieren. Dieses System, den Pathway Modeler, stellen wir in Abschnitt 3 vor.
- Wir zeigen, dass sich gefärbte Petri-Netze (Colored Petri Nets, CPNs) eignen, um Pathways zu modellieren und zu simulieren und skizzieren die Abbildung von Pathways auf CPNs (vgl. Abschn. 4.1 u. 4.2).
- Wir formalisieren die Abbildung unter Verwendung eines MDE-Ansatzes in Abschnitt 5.
- Erste Analyseergebnisse präsentieren wir in Abschnitt 5.4.

Der Beitrag schließt mit einer Einordnung unseres Ansatzes in die Literatur (Abschn. 6), einer Zusammenfassung und einem Ausblick auf zukünftige Arbeiten (Abschn. 7).

2 Ausgangssituation

Proteine haben in Organismen vielfältige Aufgaben und sie reagieren miteinander auf ebenso vielfältige Art und Weise. Diese Interaktionen von Proteinen werden mit Hilfe von Pathways oder Netzwerken beschrieben, die häufig auch graphisch in semi-formaler Weise repräsentiert werden. Man unterscheidet zwischen metabolischen und regulatorischen Pathways, wobei die metabolischen Pathways Stoffwechselforgänge beschreiben, also die Umsetzung von Stoffklassen in andere. Regulatorische Pathways hingegen regeln die Antworten von Zellen auf externe Stimuli, indem sie zum Beispiel die Synthese oder den Abbau anderer Moleküle bewirken. Hier steht nicht die Stoffumwandlung im Vordergrund sondern die Weiterleitung von Signalen. Man spricht daher auch von Signaltransduktionswegen.

Nimmt die Zelle ein extrazelluläres Signal wahr, kommt es zu einer „Zellantwort“, das heißt, die Signale in der Zelle aktivieren spezifische Transkriptionsfaktoren, die mit der entsprechenden DNA interagieren und als Folge eine Genexpression auslösen. Das Ergebnis der Genexpression sind Zielproteine oder Enzyme, die Einfluss auf eine Veränderung

Organismen. Diese einzelnen Ergebnisse werden von anderen Experten zu umfassenden Bildern, den Pathways, zusammengesetzt und in Pathway-Datenbanken zur Verfügung gestellt. Beispiele für solche Datenbanken sind TRANSPATH[®] [KPV⁺06], Reactome [JTG⁺05] und KEGG [KGH⁺06]. Sie stellen normalerweise die Daten über ein Web-Interface zur interaktiven Suche sowie als Text-Dateien im Flatfile- oder XML-Format zur lokalen Speicherung zur Verfügung. Einige dieser Datenbanken verwenden bereits halbwegs standardisierte Austauschformate auf XML-Basis wie zum Beispiel SBML (Systems Biology Markup Language, [HFB⁺04]).

Um solche Pathways im Detail verstehen zu können, ist es hilfreich, sie in Modelle umzusetzen, die sich mit den entsprechenden Werkzeugen simulieren lassen. Allein der Prozess der Umsetzung in eine formale Darstellung wie z.B. ein Petri-Netz hilft, Inkonsistenzen in den Daten zu erkennen. Simulationen können dazu beitragen, die Anzahl der im Labor durchzuführenden Experimente zu verringern.

Durch die Anwendung verschiedener Modellierungssprachen ist es möglich, Signaltransduktionswege aus unterschiedlichen Perspektiven zu analysieren. Zum Ersten eignen sich Live Sequence Charts (LSCs) in Verbindung mit der Software-Applikation Play-Engine zur Beobachtung von Signalflüssen durch Aktivierung und Inaktivierung von Molekülen. Zum Zweiten erlaubt die Modellierung von stöchiometrischen Gleichungen mithilfe von gefärbten Petri-Netzen Reaktionseffekte mittels ihrer spezifischen Strukturen in Modellen zu erkennen, Ablaufmuster zu identifizieren und Pathways zu analysieren. Zum Dritten werden durch UML-Statecharts Reaktionseffekte durch Modellierung des Intraobjektverhaltens von Molekülen präzisiert und das Verständnis für Reaktionen und Signaltransduktionswege vertieft.

3 Der Pathway Modeler

Wir haben ein erweiterbares System, den Pathway Modeler, entwickelt, das in der Lage ist, aus den Daten, die in Pathway-Datenbanken abgelegt sind, automatisch Modelle in verschiedenen Modellierungssprachen zu generieren, sodass sie mit den zugehörigen Simulationswerkzeugen simuliert werden können. Die generelle Architektur des Pathway Modelers ist in Abbildung 2 zu sehen. Dort wird als Datenquelle die TRANSPATH[®]-Datenbank dargestellt, mit der wir zunächst begonnen haben zu arbeiten. Die Anbindung weiterer Datenbanken ist in Vorbereitung.

Zunächst muss der Benutzer den Pathway auswählen, den er simulieren möchte. Dabei überprüft das System, ob die zugehörige Pathway-Instanz bereits im Instance Repository vorhanden ist (1). Falls nicht, extrahiert es alle zu diesem Pathway gehörenden Daten aus den XML-Dateien der TRANSPATH[®]-Datenbank und erstellt eine Pathway-Instanz (1.1 - 1.3). Anschließend wählt der Benutzer das Simulationswerkzeug aus, für das er ein Pathway-Modell generieren möchte (2). Zur Zeit stehen hier die Werkzeuge „CPN Tools“ und „Play Engine“ zur Verfügung. In Abhängigkeit von der Auswahl werden dann ein gefärbtes Petri-Netz bzw. eine Live Sequence Chart-Repräsentation für den selektierten Pathway generiert (3a/3b). Das System ist erweiterbar gestaltet, so dass sich andere Simu-

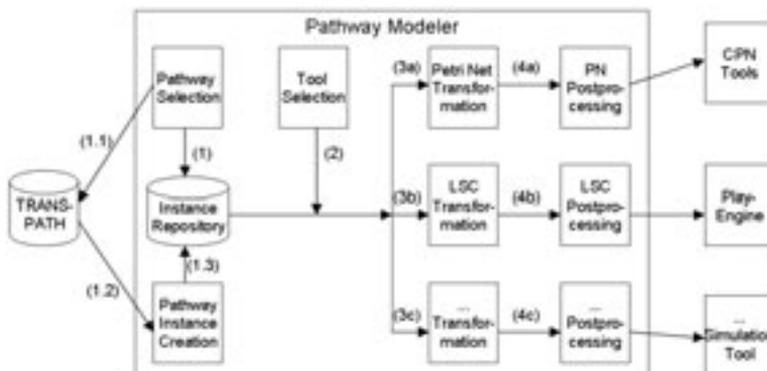


Abbildung 2: Architektur und Kontrollfluss des Pathway Modelers

lationstools integrieren lassen (3c). Nachdem die Transformation durchgeführt, also ein Pathway-Modell in der gewählten Sprache generiert wurde, müssen noch einige Nachverarbeitungsschritte durchgeführt werden. Das Ergebnis ist dann eine Eingabedatei für das selektierte Simulationstool.

Die Datenbank, die wir zunächst betrachtet haben, ist die TRANSPATH[®]-Datenbank, welche die in Abschnitt 2 erwähnten Zugriffsmethoden unterstützt, aber ihr eigenes XML-Format verwendet. Im Januar 2007 enthielt die TRANSPATH[®] etwa 60.000 Moleküleinträge, 100.000 chemische Reaktionen, 20.000 Gene und 57 Pathways. Die enthaltenen Informationen basierten auf 30.000 Veröffentlichungen. Das Web-Interface gewährt Zugriff auf alle Datenbankeinträge und stellt außerdem interaktive Graphiken zur Verfügung, die eine Übersicht über die Pathways geben (vgl. Abb. 1). Die XML-Version der Datenbank besteht aus 6 Dateien, die die Daten über Moleküle, Gene, Reaktionen, Pathways, Annotationen und Referenzen enthalten, sowie einer DTD, die die Struktur beschreibt. Die TRANSPATH[®] ist Teil des TRANSFAC[®]-Systems [Win04], einer Reihe von Datenbanken, die Informationen über Genexpression enthalten, wie zum Beispiel über Transkriptionsfaktoren, DNA-Bindungsstellen etc.

Zur Erstellung der Pathway-Modelle in den jeweiligen Modellierungssprachen wurde ein MDE-Ansatz verfolgt. Es wurde ein Metamodell für Pathways sowie Metamodelle für die Modellierungssprachen (bzw. die zugehörigen Simulationstools) erstellt und die Abbildungen zwischen diesen Metamodellen mit Hilfe von MOF-QVT formalisiert (vgl. Abschn. 5). Das Ziel dieser Formalisierung ist es, zwischen den Signaltransduktionswegen und den Sprachen Abbildungsregeln zu finden, die die Struktur und den kaskadenartigen Signalfloss der Pathways in den Modellierungssprachen widerspiegeln.

In dem vorliegenden Beitrag konzentrieren wir uns auf die Petri-Netz-Aspekte (Abschn. 4) sowie den MDE-Ansatz (Abschn. 5). Die Verwendung von Live Sequence Charts zur Modellierung von Signaltransduktionswegen sowie der Play Engine zur Simulation derselben stellen wir in [TE07] vor.

4 Modellierung und Simulation von Signaltransduktionswegen mit gefärbten Petri-Netzen

In diesem Abschnitt stellen wir ein Klassendiagramm vor, das Signaltransduktionswege beschreibt, geben einen Überblick über die Modellierung von Pathways mit gefärbten Petri-Netzen und präsentieren unser CPN-Metamodell.

4.1 Ein Pathway-Metamodell

Signaltransduktionswege werden in unserer Arbeit auf einem hohen Abstraktionslevel modelliert. Das basiert zum einen darauf, dass in der Systembiologie die Modellierung biochemischer Prozesse ausschließlich in der Betrachtung von Reaktionen - also der Interaktionen zwischen Molekülen - liegt. Diese Reaktionen lassen sich dann zu Signaltransduktionswegen zusammensetzen. Zum anderen umfasst unser Pathway-Metamodell mögliche Pathway-Modelle anderer Datenbanken und zwar auf dem Abstraktionslevel, wie wir ihn für die Abbildung auf diskrete Modellierungssprachen benötigen. Eine detaillierte Beschreibung ist in [Täu08] zu finden.

Den prinzipiellen Aufbau von Signaltransduktionswegen haben wir mit Hilfe eines Klassendiagramms formalisiert, das gleichzeitig als Metamodell in unserem MDE-Ansatz dient (vgl. Abschn. 5). Es ist in Abbildung 3 dargestellt. Jeder Pathway, jede Chain, jede Reaktion und jedes Molekül haben eine eindeutige Identifikationsnummer (*ac*) und einen Typ. Moleküle sind einer Klasse zugeordnet und besitzen wie Reaktionen einen Namen. Reaktionen besitzen einen Effekt, der entweder durch Aktivierung, Inhibierung oder, wenn keiner der beiden Effekte bisher nachgewiesen wurde, durch die Bindung spezifiziert wird. Ebenso kann hier eine andere Abstraktionsebene vorliegen, die den biochemischen Effekt durch z.B. Ubiquitinierung, Phosphorylierung etc. explizit angibt.

Ein Pathway besteht aus einer oder mehreren Chains, die sich aus einer Aneinanderreihung von Reaktionen zusammensetzen. Moleküle sind die Teilnehmer von Reaktionen und können folgende Rollen übernehmen: Als *inhibitors* hemmen sie eine Reaktion, Enzyme katalysieren sie, Moleküle sind durch *reactans* als Edukte spezifiziert und können schließlich Produkte einer Reaktion sein. Aus Sicht von *Reaction* spezifizieren *rkins* und *rkouts* Reaktionen, in denen das Molekül ein Edukt oder ein Produkt ist. *catalyses* spezifizieren Reaktionen, die von diesem Molekül katalysiert, *inhibits* solche, die von diesem Molekül gehemmt werden.

Moleküle liegen in unterschiedlichen Zuständen vor. *states* repräsentiert den elementaren Grundzustand eines Moleküls, *stateofs* seine modifizierte Form, z. B die phosphorylierte Form des Moleküls. Ist der Typ eines Moleküls mit *family* spezifiziert, zeigt *members* die dazugehörigen Moleküle auf. Jedes Molekül ist einer Proteinklasse zugeordnet. Ist diese noch nicht bestimmt, bekommt es das Attribut *other* zugewiesen.

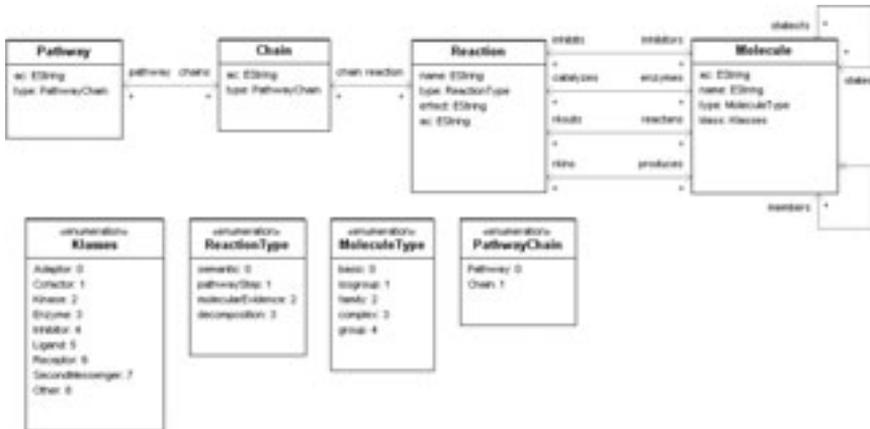


Abbildung 3: Pathway-Metamodell

4.2 Modellierung von Signaltransduktionswegen mit gefärbten Petri-Netzen

Im Gegensatz zu einfachen Petri-Netzen sind in gefärbten Petri-Netzen die Marken unterscheidbar [Jen97]. Jede Marke hat dabei einen Wert eines bestimmten Datentyps. Diese Werte werden Farben (Colors) genannt und die Datentypen Color-Sets.

Werden biologische Systeme mit einfachen Petri-Netzen modelliert, so definiert man für jedes Molekül eine eigene Stelle. Dadurch kann das Modell gerade von biochemischen Prozessen groß und unübersichtlich werden. CPNs begegnen diesem Problem mit der Einführung der Farben. Die Größe des Netzes wird dadurch reduziert, dass nicht jede Stelle ein Molekül repräsentiert, sondern Color-Sets für Klassen von Molekülen definiert sind. Jeder Stelle wird ein Datentyp, ein Color-Set, zugeordnet; die auf der Stelle liegenden Marken müssen dem Typ der Stelle entsprechen. Dadurch wird sichergestellt, dass die Prädikate auf den Schaltbedingungen durch die Stellen erfüllt werden können. Erst dann kann die Transition schalten.

Die Marken repräsentieren also Moleküle und Molekülkomplexe, und jede Stelle kann daher bestimmte Moleküle und Molekülkomplexe aufnehmen. Die Transitionen stehen für die chemischen Reaktionen. Die hereinkommenden und herausführenden Kanten einer Transition sowie die zugehörigen Kantengewichte beschreiben, wieviele Moleküle welcher Art als Edukte, Enzyme, Inhibitoren oder Produkte an der jeweiligen Reaktion beteiligt sind.

Abbildung 4 zeigt exemplarisch die Transformation einer chemischen Reaktion – die Bindung dreier Moleküle zu einem Molekülkomplex – in ein gefärbtes Petri-Netz. Die Objekte (Moleküle) im oberen Teil der Abbildung sind dabei Instanzen des Pathway-Klassendiagramms aus Abbildung 3. Abbildung 5 enthält ein Subnetz des TLR4-Pathways.

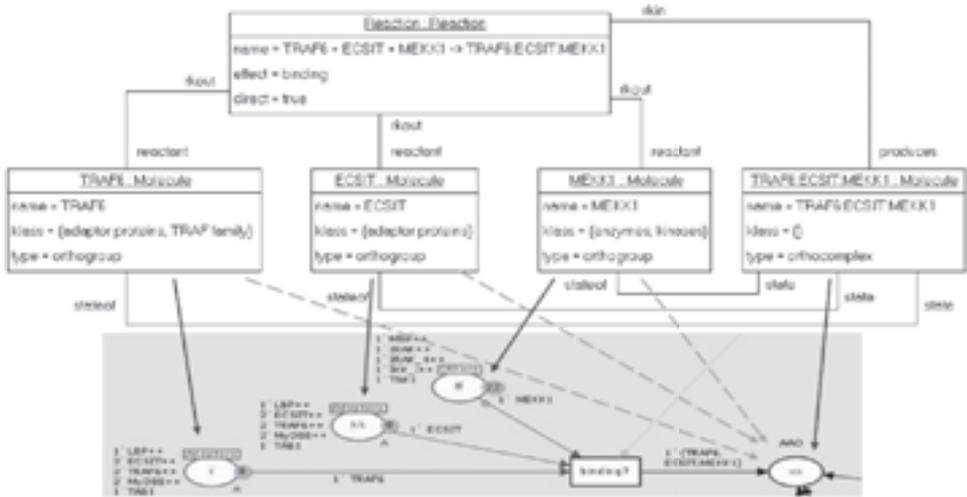


Abbildung 4: Exemplarische Transformation einer Reaktion in ein CPN

4.3 Ein CPN-Metamodell

Das CPN-Metamodell basiert neben der allgemeinen Definition für CPNs auch auf der zu der Software CPN Tools gehörenden DTD, die beschreibt, wie die einzelnen Netze intern abgelegt werden. In unserem Transformationsprozess erzeugen wir aus den Pathway-Instanzen, die ebenfalls im XML-Format vorliegen, Instanzen dieser DTD, die dann als Eingabe für das Simulationswerkzeug CPN Tools dienen. In Abbildung 6 sind die relevanten Petri-Netz-Konstrukte zusammengestellt. Eine ausführliche Beschreibung der gesamten CPN Tools DTD ist in [Zie07] zu finden.

Ein gefärbtes Petri-Netz besteht aus Stellen (Place) und Transitionen (Transition), die über Kanten (Arcs) miteinander verbunden sind. Jede Stelle ist genau einem Datentyp (ColorSet) zugeordnet und kann auch nur Elemente dieses Datentyps vereinen. Token repräsentieren die Anzahl der Elemente auf dieser Stelle. Initalmarking repräsentiert die initiale Belegung einer Stelle / des Netzes, Marking stellt die aktuelle Belegung der Stelle / des Netzes nach dem Schalten der Transitionen dar. Transitionen feuern, wenn die Belegung der Eingangsstellen die Bedingungen (Condition)/ Kantengewichtungen erfüllen.

5 Formalisierung der Transformation

In diesem Abschnitt formalisieren wir die im vorangegangenen Abschnitt skizzierte Abbildung von Signaltransduktionswegen auf gefärbte Petri-Netze. Wir geben dazu einen Überblick über die Sprache MOF-QVT, zeigen wie die Abbildung vom Pathway-Meta-



Abbildung 5: Ein Subnetz des TLR4-CPN

modell in das CPN-Metamodell formalisiert wird und diskutieren die Automatisierung von Modelltransformationen. Anschließend präsentieren wir erste Simulationsergebnisse.

5.1 MOF-QVT

Im MDA-Leitfaden [OMG05] werden Transformationen zwischen Modellen erwähnt. Diese werden in der QVT Adopted Specification in Form von Modell-zu-Modell-Transformationen formalisiert. QVT besteht aus drei Sprachen: QVT Relations und QVT Operational Mappings, die beide auf der dritten Sprache QVT Core aufsetzen. QVT Relations und QVT Core sind deklarative Sprachen, QVT Operation Mappings ist dagegen eine imperative Sprache. Die hybride Sprachkultur wurde eingeführt, um unterschiedlichen Benutzertypen gerecht zu werden.

Mit QVT Relations können Relationen zwischen Teilen des Quell- und Zielmetamodells in Form von Objektmustern und OCL-Ausdrücken spezifiziert werden. Die Sprache basiert auf drei Konzepten. Mit den Anfragen (**Queries**) wählt man aus einem Metamodell eine Menge von Modellelementen aus, die dann zur Weiterverarbeitung bereitgestellt werden. Sichten (**Views**) definieren komplexe Anfragen, die ganze Abschnitte aus einem Modell selektieren. Eine Sicht stellt dann ein Teilmodell dar, welches aus einem Ursprungsmodell abgeleitet wurde. Änderungen am Modell wirken sich auch immer auf das Teilmodell aus und umgekehrt. Transformationen (**T**ransformation) werden in dieser Arbeit verwendet,

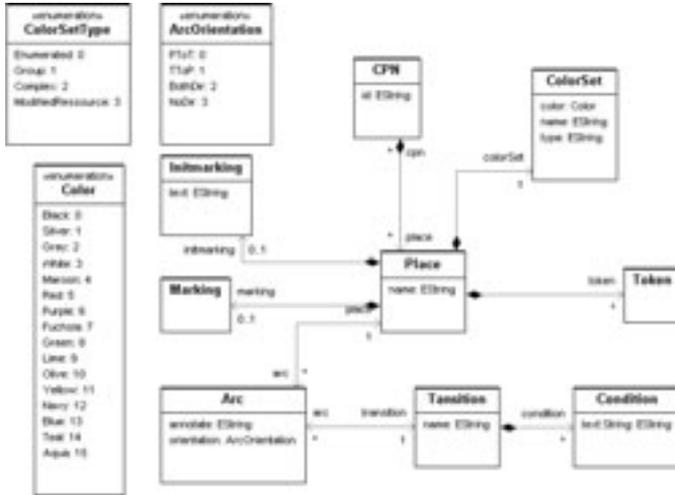


Abbildung 6: CPN-Metamodell

um Beziehungen zwischen Modellen herzustellen. Sie werden dazu benutzt, ein Quellmodell in ein Zielmodell zu überführen. Das Hauptanwendungsgebiet von MOF-QVT besteht gerade aus diesen Transformationen. Anfragen und Sichten sind Teile einer Transformation.

Eine Transformation besteht aus einer oder mehreren Domänen. Eine Domäne ist ein fachlicher Geltungsbereich, welcher sich in einer Menge von Modellelementen eines spezifischen Modells widerspiegelt. Die Modellelemente sind die Basis für die Transformationen und im Metamodell spezifiziert. Transformationen bestehen neben den Domänen aus einem Teil, welcher die Relationen definiert und einem optionalen Bereich für die Abbildungen. Relationen beschreiben und definieren Beziehungen zwischen Modellen und Modellelementen bzw. Beziehungen zwischen Domänen. Abbildungen beschreiben die Modellüberführung von einem Quellmodell in ein Zielmodell.

5.2 Transformation von Pathways in gefärbte Petri-Netze

In diesem Abschnitt wird die Transformation der Pathwayelemente auf die Colored Petri-Netz-Konstrukte beispielhaft vorgestellt. Jeder Pathway und jede Chain wird auf ein CPN abgebildet. Stellen repräsentieren Speicherstellen für die Moleküle, Marken die Moleküle selbst, Transitionen stellen die Reaktionen dar. Color-Sets repräsentieren die Typen der chemischen Elemente, deren Einteilung aus dem Elementtyp `Klassen` (vgl. Abb. 3) übernommen wird. Die gerichteten Kanten inkl. der Schaltbedingung entstammen aus der entsprechenden chemischen Reaktion. Dabei stellen die stöchiometrischen Koeffizienten die Kantengewichte dar. Die Zuordnungen vom Pathway-Metamodell zum CPN-

Metamodells sind in Tabelle 1 dargestellt. Die QVT-Transformation `PathwayToCPN`

Pathway-Metamodell	CPN-Metamodell
Pathway / Chain	CPN
Reaction	Transition
Molecule	Place
Inhibitors	Eingangsstellen
Enzymes	Eingangs-/Ausgangsstellen
Reactans	Eingangsstellen
Products	Ausgangsstellen
klass	Enumerated-Color-Sets
members	Gruppen-Color-Sets
stateofs	Komplexe-Color-Sets
states	Modified Ressources Color-Sets

Tabelle 1: Informelle Abbildung zwischen Pathway- und CPN-Metamodell

definiert eine Abbildung zwischen einer Pathway- und einer CPN-Instanz:

```
transformation PathwayToCPN (pathway : PathwayMM, cpn : CPNMM) {
  top relation PathwayToCPN {...}
  relation MoleculegroupToColorSet{...}
  relation ReactionToTransition {...}
  relation MoleculeToPlace {...} }
```

Die Transformation `PathwayToCPN` setzt sich also aus den Relationen `PathwayToCPN`, `MoleculegroupToColorSet`, `ReactionToTransition` und `MoleculeToPlace` zusammen [Täu08], von denen wir hier nur die erste vorstellen können:

```
top relation PathwayToCPN {
  pathwayAC, pathwayName, name: String;

  checkonly domain pathway p:Pathway {
    type = PathwayTypeToCPN(name),
    ac = pathwayAC}
  enforce domain cpn c:CPN {
    cpnType = PathwayTypeToCPN(name),
    id = pathwayAC};
  where {MoleculegroupToColorSet (pathwayAC)};
  function PathwayTypeToCPN (pathway : String) : String{
    if type = 'pathway'
      then name = pathwayName
      else name = pathwayName + 'chain'};

  context cpn
  inv: name = pathwayName xor name = pathwayName + 'chain' }
```

Diese Relation bildet eine Pathway-Instanz auf eine CPN-Instanz ab. Der Name des Pathways wird zum Namen des CPN deklariert. Da Pathway und Chain denselben Elementtyp innerhalb der TRANSPATH[®]-XML-Datei belegen, wird durch eine Funktion geprüft,

welche Ausprägung vorliegt. Ist `chain` als Elementtyp definiert, wird der Pathwayname um den String `Chain` erweitert. Zur Absicherung, dass innerhalb der CPN-Domäne einer der beiden Namen gilt, wird mit OCL eine Invariante definiert. Innerhalb der `cpn`-Domäne wird mit Hilfe des Schlüsselwortes `context` spezifiziert, welche Modellentität bzw. Klasse betroffen ist. Die anschließend spezifizierte Invariante definiert, mit welcher Variable ein Musterabgleich durchgeführt werden soll. Anschließend werden die Moleküle unter Verwendung der Elementtypen `groups`, `molecule`, `state` und `stateofs` auf die Color-Sets durch die Relation `MoleculegroupToColorSets (pathwayAC)` abgebildet.

Die für die Berechnung des Pathway-Namens definierte Funktion trägt nicht zum Transformationsprozess bei. Sie definiert eine Erweiterung des Transformationsprozesses. Invarianten stellen Regeln für die assoziierten Objekte auf und unterstützen die Definition des Pathway- bzw. der CPN-Namensgebung.

5.3 Automatische Modelltransformationen

Die Modelltransformation ist ein wesentlicher Bestandteil von modellbasierten Softwareentwicklungstechniken, die sich zunehmend als eine Schlüsseltechnologie für die Erstellung komplexer Software erweisen. Eine Übersicht über die zur Zeit vorhandenen MDE-Werkzeuge ist auf den Seiten von modelbased.net zu finden. Es sind Modelltransformationssprachen, wie z.B. ATL (Atlas Transformation Language) oder die vorgestellte MOF-QVT Relations definiert und standardisiert worden. Diese Sprachen werden zurzeit durch wenige Werkzeuge unterstützt, sodass Transformationen schwer automatisiert werden können.

Für MOF-QVT gibt es erste Ansätze in dem IBM Model Transformation Framework (MTF), welches ein modellbasiertes Transformationsframework auf Basis der deklarativen Sprache von MOF-QVT anbietet. Ein weiterer Ansatz ist das von uns eingesetzte **MOdel-Management**-Werkzeug **MOMENT** [BCR06], das seit 2004 an der *Universidad Politécnica of Valencia* entwickelt wird und mit eine der ersten Implementierungen von MOF-QVT Relations darstellt. Ebenso wie ADT stellt MOMENT ein Plug-In unter dem Eclipse Modelling Framework (EMF) dar. Durch diese Integration erlaubt MOMENT die Verarbeitung fast aller Modelle und Metamodelle, die auch durch EMF verarbeitet werden können. MOMENT implementiert verschiedene Operatoren zur Verarbeitung, Vereinigung und Transformation von Modellen und Metamodellen. Für die in dieser Arbeit automatisch zu generierenden Ausgabeinstanzen wird der Model-Generation-Operator verwendet, der es ermöglicht, eine Eingabeinstanz des Quellmetamodells in eine Ausgabeinstanz des Zielmetamodells zu transformieren. Die Transformationsregeln werden, wie in diesem Abschnitt gezeigt, auf Basis der Metamodelle formuliert und durch die MOF-QVT Relations spezifiziert. Eine ausführliche Beschreibung der Installation, der Anwendung und der Systemarchitektur ist in [Wit07] zu finden.

Im Rahmen einer gemeinsamen Fallstudie mit den Entwicklern von MOMENT wurde versucht, aus unseren Eingabeinstanzen für das Pathway-Quellmetamodell mit Hilfe der

Transformationsregeln Ausgabeinstanzen zu erzeugen. Unsere recht komplexen Metamodelle und Transformationsregeln, die hier nur angedeutet werden konnten, halfen, eine Reihe von Fehlern zu erkennen, die zur Zeit korrigiert werden. Informationen zum aktuellen Entwicklungsstand von MOMENT sind unter <http://moment.dsic.upv.es/> zu finden.

Parallel zu dieser Fallstudie haben wir Java-Programme entwickelt [Zie07], die die mit QVT spezifizierten Transformationsregeln implementieren und Instanzen erzeugen, die zu den Zielmetamodellen konform sind. Dadurch sind wir in der Lage, mit dem Pathway Modeler bereits Petri-Netze und Live Sequence Charts zu erzeugen, auch wenn die MOF-QVT-Transformationswerkzeuge noch einige Zeit bis zur Behebung der letzten Fehler benötigen. Nichtsdestotrotz sind wir der Ansicht, dass die automatisierte Modelltransformation für unseren Ansatz vorteilhaft ist, insbesondere, wenn es an die Integration anderer Pathway-Datenbanken und weiterer Modellierungssprachen und Simulationstools geht. Der in Abschnitt 3 vorgestellte Pathway Modeler wurde so entwickelt, dass er jederzeit von der Verwendung der manuell entwickelten Java-Programme auf die automatische Modelltransformation mit MOMENT umgestellt werden kann.

5.4 Ergebnisse

Mit den in Abschnitt 5.3 angesprochenen Java-Programmen als integrativem Bestandteil des Pathway Modelers haben wir Transformationen durchgeführt, die mithilfe einer Log-Datei ausgewertet werden konnten. Parallel dazu wurden die transformierten Pathwaymodelle mithilfe der zugehörigen Simulationstools analysiert. Bei der Analyse wurde untersucht, ob die aufgetretenen Fehler in den Transformationsmodulen, in der Selektion der Daten durch den Pathway Modeler oder in der TRANSPATH[®]-Datenbank liegen. Die Analyse wurde insgesamt mit acht Pathways durchgeführt, wobei die Unstimmigkeiten im Modell oder Fehler in der Log-Datei mit der TRANSPATH[®]-Datenbank abgeglichen wurden. Die am häufigsten aufgetretenen Fehler sind inkonsistente Referenzen in den XML-Dateien. Diese Referenzen zeigen auf Moleküle, deren Accession-Nummer sich geändert hat und die deshalb nicht (oder mit einer anderen AC) in der Signaltransduktions-Instanz enthalten sind. Ein Ausschnitt aus einer Log-Datei, die genau diese Inkonsistenz abfängt, ist im folgenden Quelltext zu finden:

```
1 DEBUG Create links for reaction LPS + LBP <=> LPS:LBP :
2 DEBUG Added reactant : LPS (1)
3 ERROR The molecule MO000008394 in the reaction XN000023302 indicated as
  reactants is missing in the xml-file.
4 DEBUG Added reactant : LBP (1)
5 DEBUG Added produces : LPS:LBP (1)
```

Aus der Log-Datei geht hervor, dass für die Reaktion $LPS + LBP \Leftrightarrow LPS:LBP$ Links zu den beiden Edukten LPS und LBP sowie zu dem Produkt LPS:LBP erstellt wurden. Desweiteren wurde eine weitere Referenz auf das Molekül MO000008394 (Zeile 3) gefunden, obwohl nach Überprüfung der Reaktion in der Datenbank kein weiteres Molekül an der Reaktion beteiligt ist und aus diesem Grund diese Referenz nicht vorhanden sein darf. Die Überprüfung ergab, dass das Molekül LBP die ID als Secondary Accession Number

(SAN) hat. Diese Nummern werden erstellt, wenn zwei Einträge in der TRANSPATH[®] zusammengelegt werden oder sich die ID aus einem anderen Grund ändert. Die SANs enthalten die ursprünglichen IDs. Der Fehler wurde bereits beseitigt.

6 Andere Ansätze

Um biologische Prozesse zu beschreiben, wurden bereits verschiedenste Arten von Petri-Netzen eingesetzt [HR04, PRA05]: Die Bandbreite reicht von gefärbten Petri-Netzen zur qualitativen Modellierung metabolischer Pathways [HK04] über stochastische Petri-Netze, die eine quantitative Modellierung erlauben, bis hin zu hybriden Petri-Netzen zur quantitativen Repräsentation von metabolischen Pathways [NDMM05]. Die Pathway Logic [TEK⁺04] ist ein Ansatz, in dem Rewriting-Logik zur Modellierung von Signaltransduktionswegen verwendet wird. Ebenso wurde das π -Kalkül [Mil99] bereits zur Modellierung von Signaltransduktionswegen eingesetzt sowie zur Beschreibung von genregulatorischen Netzwerken [RS04]. In [PCC06] wird das stochastische π -Kalkül zur Modellierung biologischer Prozesse verwendet. Live Sequence Charts wurden in [KHK⁺04] eingesetzt, um bestimmte Entwicklungsphasen des Wurms *C.elegans* zu beschreiben und Statecharts zur Modellierung der T-Zellen-Aktivierung [EHC07].

Gemeinsam ist den meisten dieser Ansätze, dass die Modelle manuell erstellt und nicht aus Datenbanken generiert werden. Und in den Ansätzen, in denen die Modelle bereits aus einer oder mehreren Datenbanken automatisch generiert werden (wie z.B. [NDMM05]), wird nur eine Art von Modellen erzeugt, d.h. nur Modelle einer Sprache. Im Gegensatz dazu ist unser Ansatz darauf ausgelegt, für ein und denselben Pathway Repräsentationen in verschiedenen Sprachen zu generieren, sodass für unterschiedliche Fragestellungen unterschiedliche Modellierungssprachen und Simulationstools verwendet werden können.

7 Zusammenfassung und Ausblick

In diesem Beitrag haben wir unseren Ansatz zur Modellierung und Simulation von Signaltransduktionswegen mit diskreten Modellierungssprachen vorgestellt. Zur Zeit verwenden wir die TRANSPATH[®]-Datenbank als Datenquelle und die Simulationswerkzeuge CPN Tools und Play-Engine zur Simulation von gefärbten Petri-Netzen bzw. Live Sequence Charts.

Zukünftig wollen wir weitere Modellierungssprachen und Simulationswerkzeuge unterstützen, wie zum Beispiel Statecharts und das Rhapsody-Tool, für das erste Vorbereitungen bereits begonnen haben. Des Weiteren werden wir unser System dahingehend erweitern, dass es CSML (Cell System Markup Language, [NJD⁺06]) als Eingabeformat für Signaltransduktionswege akzeptiert. Die Entwickler von CSML stellen Werkzeuge zur Verfügung, um Pathways, die in anderen XML-Formaten vorliegen, wie z.B. CellML [LHN04] and SBML [HFB⁺04], in CSML zu übersetzen. Da die meisten der großen Pathway-Datenbanken eines dieser Formate unterstützen, werden wir dann in der

Lage sein, nicht mehr nur Signaltransduktionswege aus der TRANSPATH[®] zu verarbeiten sondern ebenso aus den anderen großen Datenbanken wie beispielsweise KEGG [KGGH⁺06] und Reactome [JTGV⁺05].

Danksagung

Wir möchten uns ganz herzlich bedanken bei C. Pommerenke, E. Shelest, B. Störmann und C. Rio Bartulos für wichtige Informationen über Signaltransduktionswege, bei unseren Kollegen aus dem Intergenomics-Projekt (insbesondere bei Richard Münch) für interessante Diskussionen, bei den Entwicklern von MOMENT (insbesondere bei A. Boronat, A. Gomez und I. Ramos) für die gute Zusammenarbeit sowie bei den anonymen Gutachtern für wertvolle Hinweise, die zur Verbesserung des Manuskripts beigetragen haben.

Literatur

- [BCR06] Artur Boronat, José A. Carsí und Isidro Ramos. Algebraic Specification of a Model Transformation. In *FASE, LNCS 3922*, Seiten 262–277, 2006.
- [DK06] S. M. Dauphinee und A. Karsan. Lipopolysaccharide signaling in endothelial cells. *Lab Invest*, 86(1):9–22, 2006.
- [EHC07] S. Efroni, D. Harel und I.R. Cohen. Emergent Dynamics of Thymocyte Development and Lineage Determination. *PLoS Computational Biology*, 3(1):127–136, 2007.
- [HFB⁺04] M. Hucka, A. Finney, B.J. Bornstein, S.M. Keating, B.E. Shapiro, J. Matthews, B.L. Kovitz, M.J. Schilstra, A. Funahashi, J.C. Doyle und H. Kitano. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *Systems Biology*, 1(1):41–53, June 2004.
- [HK04] M. Heiner und I. Koch. Petri Net Based Model Validation in Systems Biology. *Proc. Int. Conf. on Application and Theory of Petri Nets*, Seiten 216–237, June 2004.
- [HR04] S. Hardy und P.N. Robillard. Modeling and Simulation of Molecular Biology Systems Using Petri Nets: Modeling Goals of Various Approaches. *J Bioinform Comput Biol*, 2(4):595–613, 2004.
- [Jen97] K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use*. Springer, Berlin, 2nd. Auflage, 1997.
- [JTGV⁺05] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D’Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney und L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(suppl.1):D428–432, 2005.
- [KGGH⁺06] M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki und M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(suppl.1):D354–357, 2006.
- [KHK⁺04] N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E.J.A. Hubbert und M.J. Stern. Formal Modelling of *C. elegans* Development - A Scenario-Based Approach. In G. Ciobanu und G. Rozenberg, Hrsg., *Modelling in Molecular Biology*, Seiten 151–173. Springer, Berlin, 2004.

- [KPV⁺06] M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis und E. Wingender. TRANSPATH[®]: An Information Resource for Storing and Visualizing Signaling Pathways and their Pathological Aberrations. *Nucleic Acids Research*, 34(suppl.1):D546–551, 2006.
- [LHN04] C.M. Lloyd, M.D.B. Halstead und Poul F. Nielsen. CellML: its future, present and past. *Progress in Biophysics and Molecular Biology*, 85(2-3):433–450, Jun-Jul 2004.
- [Mil99] R. Milner. *Communicating and Mobile Systems. Pi-calculus*. Cambridge University Press, 1999.
- [NDMM05] M. Nagasaki, A. Doi, H. Matsuno und S. and Miyano. Petri Net Based Description and Modeling of Biological Pathways. *Algebraic Biology - Computer Algebra in Biology*, Seiten 19–31, 2005.
- [NJD⁺06] M. Nagasaki, E. Jeong, A. Doi, A. Saito und S. Miyano. *CSML 3.0 - Cell System Markup Language - Basic Concept and Specification*, 2006.
- [OMG05] OMG. *MOF QVT Final Adopted Specification*. Number OMG document formal/2005-11-01. Object Modeling Group, June 2005.
- [PCC06] A. Phillips, L. Cardelli und G. Castagna. A Graphical Representation for Biological Processes in the Stochastic pi-Calculus. In C. Priami, A. Ingólfssdóttir, B. Mishra und H. Riis Nielson, Hrsg., *T. Comp. Sys. Biology*, LNCS 4230, Seiten 123–152. Springer, 2006.
- [PRA05] M. Peleg, D. Rubin und R.B. Altman. Using Petri Net tools to study properties and dynamics of biological systems. *J Am Med Inform Assoc*, 12(2):181–99, 2005.
- [RS04] A. Regev und E. Shapiro. The pi-calculus as an Abstraction for Biomolecular Systems. In Gabriel Ciobanu und Grzegorz Rozenberg, Hrsg., *Modelling in Molecular Biology*, Seiten 219–266. Springer, Berlin, 2004.
- [TE07] C. Täubner und S. Eckstein. Signal Transduction Pathways as Concurrent Reactive Systems - A Modeling and Simulation Approach Using LSCs and the Play-Engine. In *From Biology To Concurrency and back. Satellite Workshop of Concur 2007*, Seiten 1–15, Lisbon, Portugal, September 8 2007 2007.
- [TEK⁺04] C. Talcott, S. Eker, M. Knapp, P. Lincoln und K. Laderoute. Pathway Logic: Modeling of Protein Functional Domains in Signal Transduction. In R.B. Altman, A.K. Dunker, L. Hunter, T.A. Jung und T.E. Klein, Hrsg., *Proc. of the Pacific Symposium on Biocomputing*, Seiten 568–580, Singapore, 2004. World Scientific.
- [Täu08] Claudia Täubner. *Modellierung und Simulation von Signaltransduktionswegen - Anwendung von diskreten Modellierungssprachen in der Systembiologie*. Reihe Informatik. Sierke Verlag, in press, Göttingen, 2008. Dissertation, TU Braunschweig.
- [Win04] E. Wingender. TRANSFAC[®], TRANSPATH[®] and CYTOMER[®] as starting points for an ontology of regulatory networks. *In Silico Biol*, 4(1):55–61, 2004.
- [Wit07] Ole Witthöft. *Pathway Modeler - Implementierung einer Umgebung zur automatischen Generierung von Pathwaymodellen in Java und Integration von Teilanwendungen in das Eclipse Modeling Framework*. Diplomarbeit, TU Braunschweig, 2007.
- [Zie07] Mareike Ziegler. *Implementierung eines Transformationsmoduls in Java zur Abbildung von Signaltransduktions-Instanzen auf CPN-Instanzen*. Diplomarbeit, TU Braunschweig, 2007.