

Methoden und Algorithmen für die Systems Biology Graphical Notation

Dissertation
zur Erlangung des
Doktorgrades der Naturwissenschaften (Dr. rer. nat.)

der



Naturwissenschaftlichen Fakultät III
Agrar- und Ernährungswissenschaften,
Geowissenschaften und Informatik

der Martin-Luther-Universität Halle-Wittenberg

vorgelegt von

Herrn Czauderna, Tobias
Geb. am 08.12.1977 in Magdeburg

Gutachter: Prof. Dr. Falk Schreiber
Prof. Dr. Ursula Kummer

Verteidigung am: 12.01.2015

Danksagung

Mein besonderer Dank gilt Prof. Dr. Falk Schreiber für die Betreuung dieser Arbeit, für die Einführung in die Systems Biology Graphical Notation (SBGN) sowie für die fortwährende Unterstützung und Motivation.

Ganz herzlich bedanken möchte ich mich bei allen (aktuellen und ehemaligen) Kollegen vom Leibniz-Institut für Pflanzengenetik und Kulturpflanzenforschung (IPK) Gatersleben, insbesondere Dr. Eva Grafahrend-Belau, Anja Hartmann, Dr. Astrid Junker, Dr. Christian Klukas, Matthias Klapperstück, Dr. Hendrik Rohn und Torsten Vogt, die mich durch Diskussionen, fachlichen Rat und technische Hilfestellung bei der Erstellung dieser Arbeit unterstützt haben.

Mein Dank geht auch an die Mitglieder der SBGN-Community, speziell an alle SBGN-Redakteure, die durch ihre Mitarbeit, ihr Engagement und durch angeregte Diskussionen zur Entwicklung von SBGN beigetragen haben und immer noch beitragen. Weiterhin bedanke ich mich bei Dr. Michael Wybrow, Prof. Kim Marriott und Dr. Tim Dwyer für eine konstruktive Zusammenarbeit, zahlreiche Diskussionen und die Möglichkeit für einige Gastaufenthalte an der Monash University (Melbourne, Australien). Ein Dankeschön für letzte Korrekturen und Verbesserungsvorschläge geht an Matthias Klapperstück.

Abschließend möchte ich mich herzlich bei meiner Familie, vor allem bei meinen Eltern, für die ununterbrochene Unterstützung, den Zuspruch und die Motivation bedanken.

Inhaltsverzeichnis

Abkürzungsverzeichnis	v
1 Einführung	1
1.1 Motivation und Zielstellung	1
1.2 Struktur der Arbeit	3
1.3 Veröffentlichungen	4
2 Grundlagen	7
2.1 Grundlagen der Biologie	7
2.1.1 Biologische Grundbegriffe	7
2.1.2 Zelluläre Prozesse	9
2.1.3 Biologische Netzwerke	10
2.1.4 Systembiologie	11
2.2 Standardisierung in der Systembiologie	12
2.2.1 Systems Biology Markup Language	12
2.2.2 Biological Pathway Exchange	12
2.2.3 Systems Biology Graphical Notation	13
2.2.4 Systems Biology Ontology	13
2.3 Grundlagen der Mathematik und Informatik	13
2.3.1 Mengen	13
2.3.2 Graphentheorie	14
2.3.3 Zeichnen von Graphen	15
2.3.4 Darstellung biologischer Netzwerke	17
2.4 Zusammenfassung	19
3 Systems Biology Graphical Notation	21
3.1 Der SBGN-Entwicklungsprozess	24
3.2 Die <i>Process Description</i> -Sprache	25
3.3 Die <i>Entity Relationship</i> -Sprache	27
3.4 Die <i>Activity Flow</i> -Sprache	29
3.5 Regeln für SBGN-Karten	31

3.5.1	Konzepte	31
3.5.2	Syntax	32
3.5.3	Semantik	32
3.5.4	Layout-Richtlinien	33
3.6	Repositorien und Software-Anwendungen für SBGN-Karten	34
3.7	Diskussion und Zusammenfassung	37
4	Übersetzung von KEGG-Karten in SBGN PD-Karten	39
4.1	Übersetzung von KEGG in SBGN	40
4.2	Layout-Prozess	43
4.2.1	Bestimmung von Knotenpositionen	45
4.2.2	Bestimmung der Kantenpfade	48
4.3	Beispiele	49
4.4	Diskussion und Zusammenfassung	50
5	Umwandlung von SBGN PD-Karten in SBGN AF-Karten	53
5.1	Regeln zur Umwandlung	54
5.1.1	Umwandlung von Knoten	55
5.1.2	Umwandlung von Kanten	57
5.2	Templates	60
5.2.1	Template „Enzym-Aktivitäten“	61
5.2.2	Template „Metabolit-Aktivitäten“	62
5.2.3	Template „Enzym- und Metabolit-Aktivitäten“	62
5.2.4	Template „Mittelbare Enzym-Aktivitäten“	63
5.3	Ablauf der Umwandlung	63
5.4	Beispiele	64
5.5	Diskussion und Zusammenfassung	65
6	SBGN Bricks	67
6.1	Biologische Prozesse und ihre Repräsentation als SBGN Brick	68
6.1.1	Bricks für Genregulationsnetzwerke	71
6.1.2	Das Bricks-Wörterbuch	73
6.2	Beispiel	74
6.3	Zusammenfassung	74
7	Import und Export von SBGN-Karten	77
7.1	Das Dateiformat SBGN-ML	77
7.2	Die Software-Bibliothek LibSBGN	80
7.3	Vergleich der Darstellung in Software-Anwendungen	81
7.4	Diskussion und Zusammenfassung	83

8 Implementierung von SBGN-ED	85
8.1 Die Software-Architektur	86
8.2 Methoden für SBGN-Karten	87
8.3 Dateiformate für SBGN-Karten	90
8.4 Externe Software-Bibliotheken	91
8.5 Diskussion und Zusammenfassung	92
9 Anwendungsbeispiele	93
9.1 Repräsentation von Stoffwechselprozessen: MetaCrop	93
9.1.1 MetaCrop 1.0	94
9.1.2 MetaCrop 2.0	95
9.1.3 Zentralstoffwechsel in Pflanzen	95
9.2 Repräsentation medizinischen Wissens: Diabetische Retinopathie	96
9.3 Repräsentation von Modellen biologischer Prozesse: Path2Models	99
9.4 Weitere Anwendungsbeispiele	103
9.5 Zusammenfassung	104
10 Zusammenfassung und Ausblick	105
10.1 Zusammenfassung	105
10.2 Ausblick	106
Literaturverzeichnis	109
A Beispiele für die Übersetzung von KEGG-Karten in SBGN PD-Karten	125
B Abbildungsvorschriften für die Umwandlung von SBGN PD-Karten in SBGN AF-Karten	133
C Beispiele für die Umwandlung von SBGN PD-Karten in SBGN AF-Karten	141
D Abbildungen der SBGN Bricks	145
E Übersicht über SBGN-ML	151

Abkürzungsverzeichnis

AF	Activity Flow
API	<i>application programming interface</i> (Programmierschnittstelle)
BioPAX	Biological Pathway Exchange
COMBINE	'COmputational Modeling in BIology' NEtwork
ER	Entity Relationship
GML	Graph Modelling Language
GraphML	Graph Markup Language
GUI	<i>graphical user interface</i> (grafische Benutzeroberfläche)
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
JPG	Joint Photographic Group
KEGG	Kyoto Encyclopedia of Genes and Genomes
KGML	KEGG Markup Language
mEPN	modified Edinburgh Pathway Notation
mSBGN	modified Systems Biology Graphical Notation
MIM	Molecular Interaction Map
PD	Process Description
PDF	Portable Document Format
PDN	Process Diagram Notation
PNG	Portable Network Graphics
REST	Representational State Transfer
SBGN	Systems Biology Graphical Notation
SBGN-ML	SBGN Markup Language
SBML	Systems Biology Markup Language
SBO	Systems Biology Ontology
SVG	Scalable Vector Graphics
SWIG	Simplified Wrapper and Interface Generator
SysML	Systems Modeling Language
UML	Unified Modeling Language
XML	Extensible Markup Language
XSD	XML Schema Definition

1,3 BPG	1,3-Bisphosphoglycerat
2 PG	2-Phosphoglycerat
3 PG	3-Phosphoglycerat
ADP	Adenosindiphosphat
ATP	Adenosintriphosphat
CoA	Coenzym A
DHA-P	Dihydroxyacetonphosphat
DNA	<i>deoxyribonucleic acid</i> (Desoxyribonukleinsäure)
FH	Fumarat-Hydratase
Fructose-6P	Fructose-6-phosphat
Fructose-1,6P	Fructose-1,6-bisphosphat
GA-3P	Glycerinaldehyd-3-phosphat
GAPDH	Glycerinaldehyd-3-phosphat-Dehydrogenase
Glucose-6P	Glucose-6-phosphat
MAPK	Mitogen-aktivierte Proteinkinase
mRNA	<i>messenger</i> RNA (Boten-RNA)
NAD ⁺	Nicotinamid-Adenin-Dinukleotid (oxidierte Form)
NADH	Nicotinamid-Adenin-Dinukleotid-Hydrid
PEP	Phosphoenolpyruvat
P _i	anorganisches Phosphat
RNA	<i>ribonucleic acid</i> (Ribonukleinsäure)

Kapitel 1

Einführung

1.1 Motivation und Zielstellung

In den letzten Jahren kam es in der biologischen Forschung zu einem Paradigmenwechsel vom reduktionistischen Ansatz der deskriptiven Untersuchung einzelner Komponenten einer Zelle oder eines Organismus hin zu einer Untersuchung der Struktur und Dynamik auf zellulärer Ebene oder auf Ebene des Organismus, um biologische Systeme in ihrer Gesamtheit zu verstehen. Möglich wurde dieser Wechsel durch die Verfügbarkeit moderner Hochdurchsatz-Verfahren wie z. B. in der Genomsequenzierung und der Massenspektrometrie, die zu einer immer größeren Menge komplexer, zusammenhängender Daten in der Biologie führen [58, 86, 121].

Die Visualisierung dieser Daten ist sowohl für das Verständnis der Daten als auch für die Analyse der Daten wichtig. Methoden der Visualisierung werden schon lange genutzt, um biologische Systeme zu verstehen, dabei werden zur Darstellung biologischen Wissens Netzwerke (z. B. Stoffwechselnetzwerke, Signalnetzwerke, Genregulationsnetzwerke oder Protein-Protein-Interaktionsnetzwerke) genutzt oder Daten (z. B. Genexpressionsdaten oder Metabolitkonzentrationen) werden im Kontext biologischer Netzwerke repräsentiert [58].

Die Biologie hat eines der höchsten Verhältnisse von visueller (grafischer) zu textueller Repräsentation von Wissen, dennoch gab es bisher keine standardisierte Notation für die grafische Darstellung [100] und Wissenschaftler mussten sich immer wieder Kenntnisse über neue Arten der Darstellung aneignen [127]. Der Austausch biologischen Wissens über grafische Darstellungen erfordert, dass sich Autor und Leser über die verwendeten Symbole, die Regeln für ihre Anordnung und die Interpretation der Darstellung verständigen [100].

Mit der Systems Biology Graphical Notation (SBGN) wurde ein Standard für die visuelle Darstellung biologischer Prozesse und Netzwerke vorgeschlagen. Drei Sprachen (*Process Description*, *Entity Relationship* und *Activity Flow*) ermöglichen es, verschiedene Aspekte biologischer Systeme darzustellen sowie unterschiedliche Sichten auf biologische Systeme zu repräsentieren. Darüber hinaus soll der Austausch biologischen Wissens erleichtert werden.

Das Ziel der vorliegenden Arbeit ist die Entwicklung von Methoden und Algorithmen für die Systems Biology Graphical Notation, die für die Anwendung der Notation zur standar-

disierten Darstellung biologischen Wissens genutzt werden und damit zu einem Erkenntnisgewinn in der Biologie beitragen. Im Rahmen dieser Dissertation sollen neuartige Methoden und Algorithmen entwickelt werden:

1. für die Übersetzung vorhandener Darstellungen biologischen Wissen aus der KEGG-Datenbank in eine standardisierte Darstellung in SBGN unter Erhalt des Layouts der ursprünglichen Darstellung,
2. für die Umwandlung einer Darstellung in der SBGN *Process Description*-Sprache in eine Darstellung in der SBGN *Activity Flow*-Sprache, um unterschiedliche Sichten auf ein biologisches System zu ermöglichen,
3. für eine Vereinfachung der Anwendung von SBGN zur Darstellung biologischen Wissens und
4. für einen einfachen Austausch und eine standardisierte Ablage einer Darstellung in SBGN in elektronischer Form.

Im Ergebnis der Arbeit sollen die entwickelten Methoden und Algorithmen in einer Implementierung verfügbar sein, um zum einen im Rahmen der vorliegenden Arbeit auf domänen-spezifischen Fragestellungen angewendet zu werden und zum anderen von Anwendern aus der Biologie nachfolgend genutzt werden zu können.

Die Thematik der Arbeit ist an der Schnittstelle zwischen der Informatik und der Biologie angesiedelt. Um adäquate Methoden und Algorithmen für die Anwendung in der Biologie zu entwickeln, ist ein umfassendes Wissen der Anwendungsdomäne notwendig und es müssen entsprechende Anforderungen aus der Anwendungsdomäne erhoben, formalisiert und berücksichtigt werden. So ist z. B. für die Übersetzung von Darstellungen aus der KEGG-Datenbank in eine Darstellung in SBGN das Layout durch Erwartungen der Anwender aus der Biologie sowie durch Anforderungen aus der entsprechenden SBGN-Spezifikation vorgegeben. Unter Anwendung gängiger Layout-Verfahren können diese Erwartungen und Anforderungen nicht erfüllt werden. Deshalb muss entweder ein neues Verfahren entwickelt werden oder ein generisches Verfahren muss so angepasst und erweitert werden, dass es die Erwartungen und Anforderungen erfüllen kann. Da es verschiedene Möglichkeiten der Abstraktion von der *Process Description*- in die *Activity Flow*-Sprache gibt, müssen mit Anwendern Regeln für die Umwandlung einer Darstellung in der *Process Description*-Sprache in eine Darstellung in der *Activity Flow*-Sprache erhoben und in einem regelbasierten System umgesetzt werden. Für eine Vereinfachung der Anwendung von SBGN muss biologisches Wissen aus der Anwendungsdomäne für eine Darstellung in SBGN strukturiert aufbereitet werden. Dazu müssen in Zusammenarbeit mit Anwendern aus der Biologie biologische Prozesse identifiziert und so formalisiert werden, dass sie eine eindeutige Repräsentation in SBGN haben.

1.2 Struktur der Arbeit

In Kapitel 2 werden Grundlagen aus der Biologie, der Mathematik und der Informatik sowie Standardisierungsansätze aus der Systembiologie eingeführt, die zum Verständnis der Arbeit beitragen. Es werden in kurzer Form Grundbegriffe aus der Biologie, zelluläre Prozesse sowie biologische Netzwerke beschrieben und es wird ein Überblick über die Graphentheorie, das Zeichnen von Graphen sowie zur Darstellung biologischer Netzwerke gegeben. Da die Systems Biology Graphical Notation die Basis der vorliegenden Arbeit darstellt, wird sie in Kapitel 3 ausführlicher vorgestellt, zusätzlich wird eine Übersicht über Repositorien für SBGN-Karten gezeigt und es wird ein Vergleich von Software-Anwendungen für SBGN-Karten im Kontext dieser Arbeit durchgeführt.

Kapitel 4 beschreibt eine Methode zur Übersetzung von KEGG-Karten aus der KEGG-Datenbank in SBGN PD-Karten unter Anwendung eines Layout-Algorithmus, der das originale, auf Expertenwissen basierende Layout erhält und gleichzeitig SBGN Layout-Richtlinien berücksichtigt. Eine auf Regeln basierende Methode zur Umwandlung von SBGN PD-Karten in SBGN AF-Karten, die unterschiedliche Sichten auf ein biologisches System ermöglicht, wird in Kapitel 5 erläutert. Das Konzept der SBGN Bricks wird in Kapitel 6 vorgestellt. Die SBGN Bricks stellen in allen drei SBGN-Sprachen Templates bereit, die eine Vielzahl biologischer Prozesse repräsentieren und leicht zu verschiedenen Arten biologischer Netzwerke zusammengesetzt werden können und dadurch den Aufwand zur Erstellung von SBGN-Karten verringern.

Das Dateiformat SBGN-ML und die Software-Bibliothek LibSBGN für den Import und Export von SBGN-Karten und die damit verbundene Möglichkeit zum Austausch von SBGN-Karten zwischen Software-Anwendungen wird in Kapitel 7 beschrieben. In Kapitel 8 wird ein Überblick über die Implementierung der Software-Anwendung SBGN-ED gegeben, die alle in den vorherigen Kapiteln dargestellten Methoden und Konzepte implementiert und zusätzliche Methoden zum Erstellen, Bearbeiten und Validieren von SBGN-Karten bereitstellt.

Im Rahmen dieser Dissertation erarbeitete Anwendungsbeispiele zur Repräsentation von Stoffwechselprozessen, zur Repräsentation medizinischen Wissens und zur Repräsentation von Modellen biologischer Prozesse werden in Kapitel 9 dargestellt, es wird außerdem ein Überblick über weitere Anwendungsbeispiele gegeben.

Diese acht Kapitel der Arbeit lassen sich den vier Bereichen Grundlagen (Kapitel 2 und 3), Methoden (Kapitel 4, 5 und 6), Implementierung (Kapitel 7 und 8) und Anwendung (Kapitel 9) zuordnen, dies ist in einer Übersicht in Abbildung 1.1 dargestellt.

Kapitel 10 fasst die vorliegende Arbeit zusammen und gibt einen Ausblick auf weiterführende Entwicklungen.

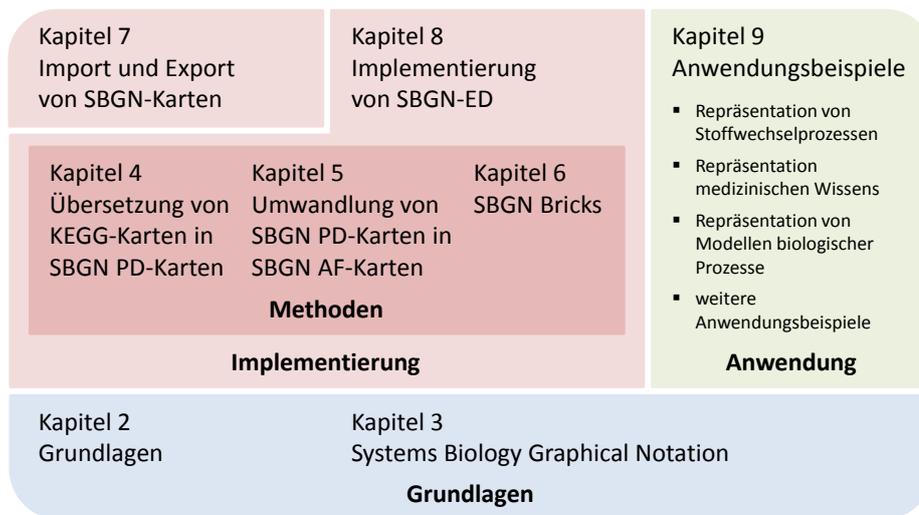


Abbildung 1.1: Struktur der vorliegenden Arbeit. Die einzelnen Kapitel lassen sich den vier Bereichen Grundlagen, Methoden, Implementierung und Anwendung zuordnen.

1.3 Veröffentlichungen

Teile der vorliegenden Dissertation wurden vom Autor unter Erst- oder Mitautorschaft bereits in Fachzeitschriften veröffentlicht:

Kapitel 4

CZAUERNA, Tobias ; WYBROW, Michael ; MARRIOTT, Kim ; SCHREIBER, Falk: Conversion of KEGG metabolic pathways to SBGN maps including automatic layout. In: *BMC Bioinformatics* 14 (2013), Nr. 1, S. 250.1–17

Kapitel 5

VOGT, Torsten ; CZAUERNA, Tobias ; SCHREIBER, Falk: Translation of SBGN maps: Process Description to Activity Flow. In: *BMC Systems Biology* 7 (2013), Nr. 1, S. 115.1–19

Kapitel 6

JUNKER, Astrid ; SOROKIN, Anatoly ; CZAUERNA, Tobias ; SCHREIBER, Falk ; MAZEIN, Alexander: Wiring diagrams in biology: towards the standardized representation of biological information. In: *Trends in Biotechnology* 30 (2012), Nr. 11, S. 555–557

Kapitel 7

VAN IERSEL, Martijn P. ; VILLÉGER, Alice C. ; CZAUERNA, Tobias ; BOYD, Sarah E. ; BERGMANN, Frank T. ; LUNA, Augustin ; DEMIR, Emek ; SOROKIN, Anatoly ; DOGRUSÖZ, Ugur ; MATSUOKA, Yukiko u. a.: Software support for SBGN maps: SBGN-ML and LibSBGN. In: *Bioinformatics* 28 (2012), Nr. 15, S. 2016–2021

Kapitel 8

CZAUDERNA, Tobias ; KLUKAS, Christian ; SCHREIBER, Falk: Editing, validating and translating of SBGN maps. In: *Bioinformatics* 26 (2010), Nr. 18, S. 2340–2341

Kapitel 9

JUNKER, Astrid ; ROHN, Hendrik ; CZAUDERNA, Tobias ; KLUKAS, Christian ; HARTMANN, Anja ; SCHREIBER, Falk: Creating interactive, web-based and data-enriched maps with the Systems Biology Graphical Notation. In: *Nature Protocols* 7 (2012), S. 579–593

SCHREIBER, Falk ; COLMSEE, Christian ; CZAUDERNA, Tobias ; GRAFAHREND-BELAU, Eva ; HARTMANN, Anja ; JUNKER, Astrid ; JUNKER, Björn H. ; KLAPPERSTÜCK, Matthias ; SCHOLZ, Uwe ; WEISE, Stephan: MetaCrop 2.0: managing and exploring information about crop plant metabolism. In: *Nucleic Acids Research* 40 (2012), Nr. 1, S. D1173–D1177

BÜCHEL, Finja ; RODRIGUEZ, Nicolas ; SWAINSTON, Neil ; WRZODEK, Clemens ; CZAUDERNA, Tobias ; KELLER, Roland ; MITTAG, Florian ; SCHUBERT, Michael ; GLONT, Mihai ; GOLEBIEWSKI, Martin u. a.: Path2Models: large-scale generation of computational models from biochemical pathway maps. In: *BMC Systems Biology* 7 (2013), Nr. 1, S. 116.1–19

Kapitel 2

Grundlagen

In diesem Kapitel sollen einige Grundlagen dargestellt werden, die zum Verständnis der Arbeit beitragen. Im ersten Abschnitt des Kapitels werden in einer Übersicht Grundlagen der Biologie zusammengefasst. Das nachfolgende Kapitel beschreibt verkürzt Standardisierungsansätze in der Systembiologie mit einem Bezug zur vorliegenden Arbeit. Im letzten Abschnitt des Kapitels werden in vereinfachter Form Grundlagen aus der Mathematik und Informatik erläutert. Für weiterführende Darstellungen wird in den einzelnen Abschnitten jeweils auf die genutzte Primärliteratur verwiesen.

2.1 Grundlagen der Biologie

2.1.1 Biologische Grundbegriffe

In diesem Abschnitt werden einige biologische Grundbegriffe, die zum weiteren Verständnis beitragen, in einer stichpunktartigen Übersicht aufgelistet. Ausführliche Erläuterungen sowie weiterführende Darstellungen sind in der verwendeten Literatur zu finden [3, 123, 146].

Dimer Ein Dimer ist eine Struktur, die aus zwei gleichen Untereinheiten besteht. Es wird auch die Bezeichnung Heterodimer verwendet, wenn die Untereinheiten nicht identisch sind.

DNA Die DNA ist der Träger der genetischen Information, DNA (Desoxyribonukleinsäure) ist ein doppelsträngiges Polymer aus Desoxynukleotiden.

Enzym Ein Enzym ist meist ein Protein, einige wenige sind RNA-Moleküle. Enzyme katalysieren (beschleunigen) jeweils eine spezifische chemische Reaktion durch Verringerung der für die Reaktion erforderlichen Aktivierungsenergie ohne dabei selbst verändert zu werden.

Gen Ein Gen ist ein Abschnitt auf der DNA, der die Information zur Synthese eines spezifischen Proteins oder eines spezifischen RNA-Moleküls enthält.

Kinase Eine Kinase ist ein Enzym, das Phosphatgruppen von einem Nucleosidtriphosphat auf ein anderes Molekül überträgt.

Kleines Molekül Ein kleines Molekül ist ein Molekül mit einer molekularen Masse zwischen 100 und 1000 Dalton, kleine Moleküle wie z.B. Zucker, Fettsäuren, Aminosäuren und Nucleotide sind die Bausteine von Makromolekülen.

Kompartimentierung Kompartimentierung beschreibt die Unterteilung einer Zelle in kleinere, von Membranen umgebene, Einheiten (Kompartimente), die innerhalb einer Zelle spezielle Funktionen erfüllen.

Komplex Ein Komplex ist eine strukturierte Ansammlung von Molekülen (Makromolekülen), die durch eine nicht kovalente Bindung zusammengehalten werden.

Kovalente Bindung Eine kovalente Bindung ist eine stabile chemische Bindung zwischen zwei Molekülen, die durch die gemeinsame Nutzung von einem oder mehreren Elektronenpaaren entsteht.

Makromolekül Ein Makromolekül ist ein Polymermolekül, z. B. ein Protein, eine Nucleinsäure (z. B. DNA oder RNA) oder ein Polysaccharid, mit einer molekularen Masse über einigen tausend Dalton.

mRNA Die mRNA ist ein RNA-Molekül, dessen Sequenz komplementär zur Sequenz eines proteinkodierenden Gens ist, sie spezifiziert die Aminosäuresequenz eines Proteins.

Metabolit Ein Metabolit ist ein Reaktionspartner (ein Edukt, ein Intermediat oder ein Produkt) einer Stoffwechselreaktion.

Molekül Eine Molekül bezeichnet eine Gruppe von Atomen, die durch kovalente Bindungen zusammengehalten werden.

Multimer Ein Multimer ist eine Struktur, die aus zwei oder mehr Untereinheiten besteht.

Nicht kovalente Bindung Eine nicht kovalente Bindung ist eine schwache chemische Bindung zwischen zwei Molekülen, bei der keine Elektronen gemeinsam genutzt werden.

Nucleotid Ein Nucleotid ist aus einer Pentose, einer Purin- oder Pyrimidinbase und einer Phosphorsäure zusammengesetzt. Biologisch wichtige Nucleotide sind die Polynucleotide oder Nucleinsäuren DNA und RNA. Als Pentose kommen hier Desoxyribose (DNA) und Ribose (RNA) vor.

Polymer Ein Polymer ist ein Molekül, das aus zahlreichen kleineren Einheiten besteht, die in geordneter Weise miteinander verbunden sind.

Protein Ein Protein ist ein Polymer aus Aminosäuren, die in einer spezifischen Sequenz miteinander verbunden sind. Die Aminosäuresequenz bestimmt die Struktur und die biologische Aktivität des Proteins.

Rezeptorprotein Ein Rezeptorprotein ist ein Protein, das einen Reiz außerhalb einer Zelle erkennt und aufnimmt und dann eine Reaktion innerhalb einer Zelle auslöst.

RNA Die RNA (Ribonukleinsäure) ist ein einsträngiges Polymer aus Ribonukleotiden.

Stoffwechselweg Ein Stoffwechselweg bezeichnet eine definierte Abfolge von biochemischen Reaktionen zum Aufbau, Abbau oder Umbau einer spezifischen Substanz. In einem Stoffwechselweg ist das Produkt der einen Reaktion das Edukt der nächsten Reaktion.

Zelle Die Zelle ist der Grundbaustein der Organismen. Zellen bestehen aus einer wässrigen Lösung organischer Moleküle, die von einer Membran umschlossen werden. Es existieren Organismen, die nur aus einer Zelle bestehen und Organismen, die aus einer Vielzahl von weitgehend differenzierten und spezialisierten Zellen bestehen.

2.1.2 Zelluläre Prozesse

Im Folgenden sollen einige Grundbegriffe zu zellulären Vorgängen (Prozessen) verkürzt erläutert werden. Für eine detaillierte Beschreibung zellulärer Prozesse sei auf die genutzte Primärliteratur verwiesen [3, 78, 83, 123, 146].

Aktiver Transport Ein aktiver Transport ist der Transport eines Moleküls durch eine Membran, getrieben durch die ATP-Hydrolyse oder eine anderen Form von chemischer Energie.

Biochemische Reaktion Eine biochemische Reaktion wandelt ein Edukt oder ein Intermediat in ein Produkt oder ein anderes Intermediat um, biochemische Reaktionen sind zumeist durch ein Enzym katalysiert.

Genexpression Genexpression ist die über Transkription und Translation ablaufende Dekodierung der in einem Gen enthaltenen genetischen Information, um ein Protein oder ein RNA-Molekül zu synthetisieren.

Genregulation Genregulation ist die durch Proteine (Transkriptionsfaktoren) regulierte Transkription von Genen in einer Zelle. Ein Transkriptionsfaktor kann aktivieren (positiver Effekt, das regulierte Gen wird „eingeschaltet“) oder inhibieren (negativer Effekt, das regulierte Gen wird „ausgeschaltet“). Durch Genregulation wird die Genexpression gesteuert.

Passiver Transport Ein passiver Transport ist der Transport eines Moleküls durch eine Membran, der ohne zusätzliche Energie von hoher zu niedriger Konzentration verläuft.

Phosphorylierung Phosphorylierung ist die kovalente Bindung einer Phosphatgruppe an ein kleines Molekül oder an ein Protein.

Stoffwechsel Der Stoffwechsel bezeichnet die Gesamtheit aller Reaktionen, die in einer lebenden Zelle oder einem vielzelligen Organismus stattfinden, er umfasst alle biochemischen Reaktionen sowie Transportprozesse.

Transkription Transkription bezeichnet das Kopieren (Umschreiben) der genetischen Information eines DNA-Strangs in eine komplementäre RNA-Sequenz (ein RNA-Molekül).

Translation Translation ist die Übersetzung der in der Nukleotidsequenz der mRNA kodierten genetischen Information in die Aminosäuresequenz eines Proteins, bei der drei aufeinander folgende Nukleotide der mRNA die Aminosäure bestimmen.

2.1.3 Biologische Netzwerke

Die im vorherigen Abschnitt beschriebenen zellulären Prozesse laufen in einer Zelle nicht unabhängig voneinander ab, sie lassen sich zu biologischen Netzwerken zusammenschalten. In diesem Abschnitt sollen deshalb wichtige biologische Netzwerke vorgestellt werden [3, 78, 83].

Stoffwechselnetzwerk (auch metabolisches Netzwerk) Die Gesamtheit der in einer Zelle oder in ihren Kompartimenten stattfindenden Reaktionen bilden das Stoffwechselnetzwerk, das die Metabolite sowie biochemische Reaktionen und Transportprozesse umfasst. Metabolite sind meist kleine Moleküle, können aber auch Makromoleküle sein. Die biochemischen Reaktionen, die Metabolite in andere Metabolite umwandeln, werden häufig durch Enzyme katalysiert, nur sehr wenige biochemische Reaktionen laufen spontan ab. Ein Stoffwechselweg kann als Teil eines Stoffwechselnetzwerkes mit einer speziellen Funktion betrachtet werden, das Stoffwechselnetzwerk selbst bietet eine umfassendere Sicht auf den Stoffwechsel einer Zelle.

Signalnetzwerk (auch Signaltransduktionsnetzwerke) In einem mehrzelligen Organismus kommunizieren Zellen mit Hilfe zahlreicher extrazellulärer Signale, um auf wechselnde Umwelt- und Entwicklungsbedingungen zu reagieren. Rezeptorproteine an der Zellwand bzw. Zellmembran empfangen diese extrazellulären Signale und wandeln sie in intrazelluläre Signale um. Diese Signale werden dann auf intrazellulären Signalwegen (Signalkaskaden) weitergeleitet. Dies geschieht durch Modifikation von Proteinen (Phosphorylierung) oder durch Bildung von Komplexen. Die Signale werden schließlich z. B. durch Proteine (Transkriptionsfaktoren oder Enzyme) empfangen und die Zelle reagiert auf die Signale.

Protein-Protein-Interaktionsnetzwerk Proteine kontrollieren und vermitteln zahlreiche Prozesse in einer Zelle. Sie agieren als Katalysatoren (Enzyme), transportieren oder speichern andere Moleküle, übermitteln Signale, kontrollieren Wachstum und Entwicklung und vieles mehr. Proteine sind jedoch selten einzeln funktionell aktiv. Für eine biologische Aktivität ist die Wechselwirkung mit weiteren Proteinen oder die Bildung von

größeren Proteinkomplexen notwendig, Proteine müssen miteinander interagieren. Ein Protein kann sich gleichzeitig mit mehreren anderen Proteinen in Interaktion befinden.

Genregulationsnetzwerk Ein Genregulationsnetzwerk ist ein Netzwerk aus Genen, den korrespondierenden Transkripten (der mRNA) und Proteinen in einer Zelle, Wechselwirkungen zwischen Genen und Proteinen spezifizieren regulatorische Aspekte. Ein Protein ist das Produkt der Transkription eines Gens und der Translation der korrespondierenden mRNA, die Transkription wird über Transkriptionsfaktoren (Proteine) reguliert. Ein Protein als Produkt einer Transkription und Translation kann selbst als Transkriptionsfaktor agieren und damit die Transkription anderer Gene regulieren, es bilden sich regulatorische Kaskaden.

2.1.4 Systembiologie

Dieser Abschnitt gibt einen kurzen Überblick über das Gebiet der Systembiologie, für eine umfängliche Einführung sei auf die weiterführende Literatur wie z. B. [2, 85, 89] verwiesen.

In der Literatur lässt sich keine allgemeingültige Beschreibung des Begriffes *Systembiologie* finden [2, 85, 89, 120]. Hiroaki Kitano definiert Systembiologie als einen Bereich der Biologie, der versucht, biologische Systeme in ihrer Gesamtheit zu verstehen [85]. Um dieses Ziel zu erreichen, genügt es nicht, die Eigenschaften einzelner Komponenten einer Zelle oder eines Organismus zu beschreiben, sondern es ist notwendig, die Struktur und Dynamik auf zellulärer Ebene oder auf Ebene des Organismus zu untersuchen [86].

Als Basis für ein gesamtheitliches Verständnis biologischer Systeme werden umfassende experimentelle Daten über die biologischen Systeme und ihre Moleküle benötigt. Durch die Fortschritte in der Molekularbiologie speziell in der Genomsequenzierung und durch Erhebung von experimentellen Daten im Hochdurchsatz stehen entsprechende Datensätze zur Verfügung. Während es auch weiterhin wichtig ist, das Wissen über Gene und Proteine zu erweitern, liegt der Fokus in der Systembiologie darin, die Struktur und Dynamik eines Systems zu verstehen. Da ein System nicht nur eine Anordnung von Genen und Proteinen ist, können seine Eigenschaften nicht vollständig auf der Grundlage von Diagrammen, die die Beziehungen zwischen Genen und Proteinen darstellen, verstanden werden. Diese Diagramme helfen in begrenztem Umfang zu verstehen, wie sich Änderungen an bestimmten Teilen eines biologischen Systems auf andere Teile des Systems auswirken. Um aber zu verstehen, wie ein System funktioniert, ist es notwendig zu untersuchen, wie seine Komponenten dynamisch miteinander interagieren [85, 86].

Ist die Struktur eines biologischen Systems auf der Basis von (biologischen) Netzwerken bekannt und verstanden, kann die Dynamik des Systems untersucht werden. Für die dynamische Analyse eines biologischen Systems muss ein (mathematisches) Modell des Systems erstellt werden. Welche Analysen mit einem Modell durchgeführt werden können, hängt vom vorhandenen biologischen Wissen, das in das Modell einfließen konnte, ab. So kann ein Modell z. B. genutzt werden, um das Verhalten des Systems im Detail zu verstehen oder um Vorher-

sagen über das Verhalten des Systems bei sich ändernden Bedingungen zu machen [85, 86]. Vorhersagen von Modellen können als Hypothesen für weitere Experimente im Labor genutzt werden, die dann zu neuen experimentellen Daten führen, die zur Erweiterung oder Verbesserung von Modellen führen können. Vereinfacht zusammengefasst gibt es also in der Systembiologie idealerweise einen Kreislauf vom Experiment über experimentelle Daten zum Modell und vom Modell über Hypothesen zurück zum Experiment.

2.2 Standardisierung in der Systembiologie

Im Kontext der vorliegenden Arbeit bezieht sich Standardisierung in der Systembiologie auf Software-Infrastruktur, für einen allgemeinen Überblick über Standardisierung in der Systembiologie siehe auch [19]. Da systembiologische Ansätze sehr oft mathematische und computer-gestützte Modellierung einbeziehen, hat die Software-Infrastruktur in der systembiologischen Forschung eine hohe Bedeutung [86]. Besonders wichtig ist dabei die Entwicklung von Daten-Repositoryn sowie Standards und Software-Anwendungen für die Simulation, Analyse und Visualisierung [104]. Einige Standards der Systembiologie mit einem Bezug zur vorliegenden Arbeit sollen deshalb im Folgenden kurz vorgestellt werden.

2.2.1 Systems Biology Markup Language

Die Systems Biology Markup Language¹ (SBML) [67] ist ein frei verfügbares, auf XML-basierendes [113] Dateiformat für die Repräsentation und den Austausch von Modellen biologischer Prozesse. SBML kann z. B. Modelle des Stoffwechsels, der Signaltransduktion und der Genregulation repräsentieren und wird von einer internationalen Community gepflegt und weiterentwickelt. Eine detaillierte Beschreibung von SBML ist in der Spezifikation zu finden [66]. Für eine einfache Nutzung von SBML in Software stehen die beiden Software-Bibliotheken LibSBML [15] und JSBML [41] zur Verfügung, die SBML-Webseite listet derzeit über 250 Software-Anwendungen, die SBML unterstützen.

2.2.2 Biological Pathway Exchange

Biological Pathway Exchange² (BioPAX) [36] ist ein frei verfügbarer Standard zur Repräsentation biologischer Reaktionswege (*biological pathways*) auf molekularer und zellulärer Ebene. BioPAX wird von einer internationalen Community gepflegt und weiterentwickelt. Der Standard ermöglicht es, Stoffwechsel- und Signalwege sowie Interaktions- und Genregulationsnetzwerke zu repräsentieren und vereinfacht die Ablage und den Austausch biologischer Reaktionswege. BioPAX ist im Detail in der Spezifikation [14] beschrieben, um BioPAX in Java-basierten Software-Anwendungen zu nutzen, kann die Bibliothek Paxtools [35] verwendet werden.

¹<http://sbml.org/>

²<http://biopax.org/>

2.2.3 Systems Biology Graphical Notation

Die Systems Biology Graphical Notation³ (SBGN) [100] stellt die Grundlage dieser Arbeit dar, sie wird in Kapitel 3 ausführlicher beschrieben. SBGN bietet Möglichkeiten der Visualisierung für die beiden zuvor genannten Standards (siehe auch [36] Abbildung 6).

2.2.4 Systems Biology Ontology

Die Systems Biology Ontology⁴ (SBO) [28, 99] ist eine Sammlung von sieben orthogonalen, kontrollierten Vokabularien, die in der Systembiologie, speziell in der Modellierung, allgemein genutzte Bezeichnungen und deren Beziehungen zueinander enthalten⁵. Die Verwendung von Bezeichnungen aus der Systems Biology Ontology fügt einem Modell semantische Informationen hinzu, werden die Elemente eines Modells mit einem SBO-*Term* (Bezeichnung) annotiert, ist die Bedeutung dieser Elemente im Modell auf eindeutige Weise festgelegt. Da sowohl SBML als auch SBGN die Verwendung von Bezeichnungen aus der SBO zulassen, kann ein SBML-Modell auf Basis von Bezeichnungen aus der SBO sehr genau in SBGN visualisiert werden [28].

2.3 Grundlagen der Mathematik und Informatik

2.3.1 Mengen

Nachfolgend sollen einige, für diese Arbeit wichtige, Definitionen aus der Mengenlehre aufgeführt werden, weiterführende Darstellungen sind in der genutzten Literatur [131] zu finden.

Menge Eine Zusammenfassung von bestimmten, wohl unterschiedenen Objekten a_1, a_2, \dots, a_n zu einem Ganzen wird Menge $A = \{a_1, a_2, \dots, a_n\}$ genannt. Die Objekte a_i werden als Elemente der Menge A bezeichnet, man schreibt $a_i \in A$. Enthält eine Menge keine Elemente, so heißt diese *leere Menge* $A = \emptyset$. Zwei Mengen A und B sind gleich, wenn jedes Element aus Menge A auch Element der Menge B ist und umgekehrt.

Teilmenge Eine Menge A ist Teilmenge der Menge B , wenn alle Elemente aus A auch Elemente von B sind, man schreibt: $A \subseteq B$. Falls es Elemente in Menge B gibt, die nicht Elemente von Menge A sind, heißt A *echte Teilmenge*, man schreibt: $A \subset B$.

Durchschnitt Der Durchschnitt $A \cap B$ der beiden Mengen A und B ist die Menge aller Elemente x , die sowohl in A als auch in B vorkommen: $A \cap B = \{x \mid x \in A \wedge x \in B\}$. Zwei Mengen heißen disjunkt, wenn sie kein gemeinsames Element besitzen und somit gilt: $A \cap B = \emptyset$.

Vereinigung Die Vereinigung $A \cup B$ der beiden Mengen A und B ist die Menge aller Elemente x , die Element der Menge A oder der Menge B sind: $A \cup B = \{x \mid x \in A \vee x \in B\}$.

³<http://sbgn.org/>

⁴www.ebi.ac.uk/sbo/

⁵Zum Begriff der Ontologie im Allgemeinen und im Kontext der Informatik siehe auch [62].

Differenzmenge Die Differenzmenge $A \setminus B$ der beiden Mengen A und B ist die Menge aller Elemente x , die Element von A und nicht Element von B sind: $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$.

Kartesisches Produkt Das kartesische Produkt $A \times B$ zweier Mengen A und B ist die Menge aller Elementpaare (a, b) , für die gilt, dass a Element der Menge A ist und b Element der Menge B ist: $A \times B = \{(a, b) \mid a \in A, b \in B\}$. Die Elemente von $A \times B$ nennt man Tupel.

Potenzmenge Die Potenzmenge $\mathcal{P}(A)$ ist die Menge aller Teilmengen der Menge A : $\mathcal{P}(A) = \{B \mid B \subseteq A\}$, es gilt $A \in \mathcal{P}(A)$ und $\emptyset \in \mathcal{P}(A)$.

Abbildung Eine Abbildung oder *Funktion* $f : A \rightarrow B$ mit $x \mapsto f(x)$ ordnet jedem $x \in A$ genau ein Element aus B zu, A und B sind nicht leere Mengen.

2.3.2 Graphentheorie

In diesem Abschnitt werden Grundbegriffe der Graphentheorie zusammengefasst. Die Erläuterungen folgen der Darstellung in [72, 78, 141], es sind dort ebenso ausführlichere Betrachtungen zur Graphentheorie zu finden.

Ungerichteter Graph Ein ungerichteter Graph $G = (V, E)$ besteht aus einer Menge von Knoten V und einer Menge von Kanten E , wobei jeder Kante $e \in E$ zwei, nicht notwendigerweise verschiedene, Knoten aus V zugeordnet sind. Eine Kante e wird in der Form $e = \{u, v\}$ beschrieben, wobei u und v als Endknoten der Kante e bezeichnet werden. Eine Kante der Form $e = \{v, v\}$, deren Endknoten also identisch sind, wird als Schlinge bezeichnet. Werden zwei Knoten durch zwei Kanten e und f der Form $e = \{u, v\}$ und $f = \{u, v\}$ verbunden, so heißen die Kanten parallel. Die beiden Mengen V und E sollen im Folgenden als endlich betrachtet werden, die betrachteten Graphen sind daher ebenfalls endlich.

Nachbarschaft Zwei Knoten $u, v \in V$ heißen benachbart (adjazent), wenn sie durch eine Kante $e = \{u, v\}$ verbunden sind. Die Menge Φ aller Knoten, die zu einem Knoten v benachbart sind, heißt Nachbarschaft von v .

Weg, Zyklus, Pfad, Kreis Ein Weg ist eine Folge $(v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k)$ von Knoten und Kanten, so dass jede Kante e_i die Endknoten v_{i-1} und v_i besitzt. Man sagt, der Weg verbindet die Knoten v_0 und v_k , die als Start- und Endknoten des Weges bezeichnet werden. Wenn der Startknoten v_0 und der Endknoten v_k eines Weges identisch sind, spricht man von einem Zyklus. Ein Graph heißt azyklisch, wenn er keinen Zyklus enthält. Falls jeder Knoten v nur einmal auf einem Weg vorkommt, handelt es sich um einen Pfad. Man spricht von einem Kreis, wenn der Startknoten v_0 und der Endknoten v_k eines Pfades identisch sind.

Gerichteter Graph Ein gerichteter Graph $G = (V, E)$ besteht aus einer Menge von Knoten V und einer Menge von Kanten E , wobei jeder gerichteten Kante $e \in E$ ein geordnetes Paar (u, v) von Knoten aus V zugeordnet ist. Knoten $u \in V$ wird als Anfangsknoten der Kante e bezeichnet und Knoten $v \in V$ heißt Endknoten der Kante e , die Kante e wird in der Form $e = (u, v)$ beschrieben⁶. Die Definitionen einer Schlinge und paralleler Kanten gelten analog zur Definition für ungerichtete Graphen. Zusätzlich werden antiparallele Kanten e und f der Form $e = (u, v)$ und $f = (v, u)$ beschrieben, für die der Anfangsknoten der einen Kante dem Endknoten der anderen Kante entspricht.

Bipartiter Graph Ein bipartiter Graph $G = (V, E)$ besteht aus einer Menge von Knoten $V = U \cup W$ und einer Menge von Kanten E , wobei die Knotenmenge V in zwei disjunkte Teilmengen U und W geteilt ist, so dass jede Kante $e \in E$ genau einen Endknoten in U und einen Endknoten in W hat.

Hypergraph Ein Hypergraph $G = (V, E)$ besteht aus einer Menge von Knoten V und einer Menge von Hyperkanten E , wobei jede Hyperkante eine nicht leere Teilmenge von V ist.

Beschrifteter Graph Ein beschrifteter (*labeled*) Graph $G = (V, E)$ besteht aus einer Menge von Knoten V , einer Menge von Kanten E und einer Menge von Knotennamen L . Eine Funktion $f_L : V \rightarrow L$ ordnet jedem Knoten $v \in V$ einen Knotennamen $l \in L$ zu.

2.3.3 Zeichnen von Graphen

Das Zeichnen von Graphen (*Graph Drawing*) ist ein Gebiet der Mathematik und Informatik, das sich mit der Konstruktion geometrischer Darstellungen von Graphen beschäftigt [37, 136]. Eine Zeichnung Γ eines Graphen $G = (V, E)$ ist eine Funktion, die jeden Knoten $v \in V$ auf einen Punkt $\Gamma(v)$ und jede Kante $e \in E$ auf eine Jordan-Kurve $\Gamma(u, v)$ mit den Endpunkten $\Gamma(u)$ und $\Gamma(v)$ abbildet [37]. Die Positionierung der Knoten und die Darstellung der Kanten wird auch als Layout des Graphen bezeichnet [78].

Für einen einzelnen Graphen sind viele unterschiedliche Zeichnungen oder Darstellungen möglich. Für gewöhnlich werden Knoten durch Punkte, Rechtecke, Kreise oder andere geometrische Objekte repräsentiert und Kanten werden durch Jordan-Kurven dargestellt, die diese geometrischen Objekte verbinden. Für die Kanten werden dabei oft gerade Linien (als Spezialfall horizontale oder vertikale Linien) und Linienzüge mit zusätzlichen Stützpunkten verwendet [37].

Die Darstellung eines Graphen kann danach bewertet werden, wie verständlich sie ist bzw. wie gut die Bedeutung der Darstellung vom Leser erfasst werden kann. Eine wichtige Rolle beim Zeichnen von Graphen spielen deshalb Algorithmen, die Darstellungen von Graphen erzeugen, die einfach zu erfassen sind. Diese Algorithmen lassen sich nach verschiedenen Kriterien unterteilen: (1) Klassen von Graphen, die mit einem Algorithmus dargestellt werden

⁶Im Gegensatz zur Knotenmenge $\{u, v\}$ bei ungerichteten Graphen wird hier vom geordneten Paar (u, v) gesprochen.

können (z. B. ungerichtete Graphen, gerichtete azyklische Graphen), (2) Zeichenkonventionen, die von einem Algorithmus erfüllt werden können und (3) Klassen von Verfahren, die zum Zeichnen genutzt werden. Die Zeichenkonventionen lassen sich in zwei Gruppen unterteilen: Konventionen, die beim Zeichnen erfüllt werden müssen (z. B. Darstellung von Knoten als Kreise, Darstellung von Kanten als gerade Linien, Darstellung von Kanten als Linienzüge mit zusätzlichen Stützpunkten, Kanten sollen bei einem gerichteten Graphen in eine bestimmte Richtung zeigen) und Konventionen, die beim Zeichnen erfüllt werden sollen (z. B. minimale Zeichenfläche für den Graphen, minimale Anzahl an Kantenkreuzungen, minimale Anzahl an Kantenstützpunkten) [37]. Im Folgenden sollen drei Zeichen- oder Layout-Verfahren detaillierter beschrieben werden.

Hierarchisches Layout-Verfahren Das hierarchische Layout-Verfahren wird für gerichtete azyklische Graphen verwendet. Die Knoten werden dabei in horizontalen Ebenen angeordnet und Kanten als Linienzüge mit Stützpunkten gezeichnet. Alle Kanten sind dabei abwärts oder aufwärts gerichtet. Das Verfahren läuft prinzipiell in drei Schritten ab:

1. Die Knoten werden den Ebenen L_1, L_2, \dots, L_h zugeordnet, für jede Kante $e = (u, v)$ wird der Knoten u der Ebene L_i und der Knoten v der Ebene L_j zugeordnet, so dass gilt $i = j + 1$. Dazu werden wenn notwendig temporäre Knoten eingefügt, so dass eine Kante nicht über mehr als zwei Ebenen verläuft. Die y-Koordinate der Knoten entspricht der y-Koordinate der Ebene.
2. Die Knoten werden auf einer Ebene so sortiert, dass die Anzahl Kantenkreuzungen minimiert wird.
3. Die x-Koordinate der Knoten wird entsprechend der Sortierung aus dem zweiten Schritt zugewiesen. Temporäre Knoten werden entfernt, die Positionen werden als Positionen für Kantenstützpunkte verwendet.

Gegebenenfalls muss ein Graph in einem Vorverarbeitungsschritt in einen gerichteten azyklischen Graphen umgewandelt werden, um das Verfahren anwenden zu können [37].

Kräftebasiertes Layout-Verfahren Das kräftebasierte Layout-Verfahren wird für ungerichtete Graphen mit Kanten ohne Kantenstützpunkte verwendet. Das Layout-Verfahren wendet ein Kräftemodell, das ein physikalisches Modell für den Graphen definiert, und einen Algorithmus, der das Kräftemodell simuliert und eine Position für die Knoten ermittelt, an. Der Graph wird als ein System von Objekten und Kräften, die zwischen diesen Objekten wirken, betrachtet. Der Algorithmus sucht nach einer Konfiguration für die Objekte mit minimaler lokaler Energie, er sucht für jedes Objekt nach einer Position im System, so dass die Summe der Kräfte auf jedes Objekt null ist [37].

Auf Bedingungen basierendes Layout-Verfahren Ein auf Bedingungen (*constraints*) basierendes Layout-Verfahren versucht ähnlich dem kräftebasierten Verfahren, eine Zielfunktion, die die Qualität des Layouts bewertet, zu minimieren. Im Vergleich zum kräftebasierten Verfahren erfolgt die Minimierung der Zielfunktion unter Beachtung (geometrischer) Bedingungen für die Positionierung der Knoten im Graphen, die erfüllt sein sollen [130]. Eine allgemein verwendete Zielfunktion ist die Stressfunktion [79, 130], für die als Optimierungsverfahren *stress majorization* vorgeschlagen wurde [55]. In [43] wurde diese Technik um die Bedingung *Abstand* (*separation constraint*) erweitert. Die Bedingung Abstand hat die Form $u + d \leq v$ mit den Variablen u und v , die eine horizontale oder vertikale Position für Paare von Knoten beschreiben, und der Konstanten d , die einen minimalen Abstand zwischen u und v beschreibt. Mit der Bedingung Abstand lassen sich zusätzlich zahlreiche höherrangige Bedingungen wie z. B. Ausrichtung (*alignment*), Einkapselung (*containment*), feste (relative) Position (*fixed (relative) position*) und Nicht-Überlappung (*non-overlapping*) für ein Layout festlegen, die auf niederrangige Abstands-Bedingungen abgebildet werden können [43, 130].

Bedingungen können aus einer vorhandenen Darstellung abgeleitet [68, 98] oder durch einen Anwender explizit definiert werden, mit einem auf Bedingungen basierendem Layout-Verfahren ist es möglich, verschiedene Layout-Stile und Anwenderinteressen zu berücksichtigen [130]. Eine Übersicht über zahlreiche auf Bedingungen basierende Verfahren und Techniken in grafischen Anwendungen ist in [65] zu finden.

2.3.4 Darstellung biologischer Netzwerke

Ein biologisches Netzwerk kann als Graph $G = (V, E)$ repräsentiert werden. Die Knoten V des Graphen entsprechen dabei den Elementen eines biologischen Netzwerkes (Gene, Proteine, Metabolite, Reaktionen usw.) und die Kanten E des Graphen entsprechen den Beziehungen und Interaktionen dieser Elemente. Für die Darstellung biologischer Netzwerke können Methoden des Zeichnens von Graphen genutzt werden. Im Folgenden werden Methoden für ausgewählte biologische Netzwerke zusammengefasst, weiterführende Erläuterungen sind in der genutzten Literatur [9] zu finden.

Stoffwechselnetzwerke Stoffwechselnetzwerke werden grundlegend als Hypergraph $G = (V, E)$ beschrieben, die Knoten V entsprechen den Metaboliten und die Hyperkanten E repräsentieren die Reaktionen. Hypergraphen können jedoch als bipartite Graphen dargestellt werden. Stoffwechselnetzwerke werden deshalb oft als gerichteter, bipartiter Graph $G = (V_S, V_R, E)$ beschrieben, wobei V_S die Menge der Metabolite repräsentiert, V_R die Menge der Reaktionen repräsentiert und E die Menge der Umwandlungen (Abbau oder Aufbau) der Metabolite durch die Reaktionen repräsentiert. Etablierte Methoden zur Darstellung von Stoffwechselnetzwerken sind kräftebasierte und hierarchische Layout-Verfahren, es wurden jedoch zahlreiche weitere, angepasste Verfahren vorgeschlagen, da allgemeine Layout-Verfahren die Anforderungen insbesondere für Stoffwechselnetzwerke nicht erfüllen können [9, 128].

In [81] wird ein Verfahren vorgeschlagen, das die Topologie des Graphen untersucht (linear, kreisförmig, verzweigt) und entsprechende Layout-Verfahren auf Teilgraphen anwendet. Ein ähnlicher Ansatz wird in [12] verfolgt, der Fokus liegt dabei auf der Suche nach dem größten kreisförmigen Teilgraphen und der Anwendung eines kräftebasierten Verfahrens. Das in [149] beschriebene Verfahren konzentriert sich ebenfalls auf kreisförmige Strukturen, wobei jedoch beliebige Größen möglich sind. Zusätzlich können Knoten vervielfältigt oder vereinigt werden. Ein anderer Ansatz ist ein erweitertes hierarchisches Verfahren, bei dem Strukturen (Reaktionsknoten mit Enzymknoten und Knoten sekundärer Metabolite) lokal angeordnet werden und während des hierarchischen Layouts durch größere Knoten ersetzt werden [128]. In einem weiteren Ansatz wurde das kräftebasierte Verfahren als Basis genutzt und um die Möglichkeit erweitert, Kompartimente mit zu berücksichtigen [40].

Signalnetzwerke Signalnetzwerke werden üblicherweise als gerichteter Graph $G = (V, E)$ beschrieben, werden Kompartimente mit betrachtet, können Signalnetzwerke auch als *clustered* Graph $C = (G, T)$ beschrieben werden, wobei G ein gerichteter Graph ist und T ein Baum ist, der zur Definition der Kompartimente genutzt wird. Etablierte Methoden zur Darstellung von Signalnetzwerken sind kräftebasierte und hierarchische Layout-Verfahren, in [59] ist ein kräftebasiertes Verfahren beschrieben, das Kompartimente und unterschiedliche Kantentypen mit berücksichtigt.

Protein-Protein-Interaktionsnetzwerke Protein-Protein-Interaktionsnetzwerke werden als gerichteter Graph $G = (V, E, \tau)$ beschrieben, wobei V die Menge der Proteine repräsentiert, E die Menge der (gerichteten) Interaktionen repräsentiert und $\tau : E \rightarrow T$ den Typ jeder Kante definiert. Eine etablierte Methode zur Darstellung von Protein-Protein-Interaktionsnetzwerken ist das kräftebasierte Layout-Verfahren. In [51] wird eine Kombination aus kreisförmigen und kräftebasierten Verfahren vorgeschlagen, die unterschiedliche Kantentypen berücksichtigt.

Genregulationsnetzwerke Genregulationsnetzwerke werden üblicherweise als gerichteter Graph $G = (V, E)$ beschrieben, werden Kompartimente mit betrachtet, können Genregulationsnetzwerke auch als *clustered* Graph $C = (G, T)$ beschrieben werden, wobei G ein gerichteter Graph ist und T ein Baum ist, der zur Definition der Kompartimente genutzt wird. Etablierte Methoden zur Darstellung von Genregulationsnetzwerken sind kräftebasierte und hierarchische Layout-Verfahren, es werden aber auch orthogonale und gitterbasierte Verfahren genutzt.

Neben Layout-Verfahren, die für die Darstellung bestimmter biologischer Netzwerke angepasst wurden, werden auch generische Layout-Verfahren für biologische Netzwerke vorgeschlagen. In [103] wird ein diskretes kräftebasiertes Layout-Verfahren beschrieben, bei dem Knoten nur auf festen Gitterpositionen platziert werden. Das Ziel ist dabei eine verbesserte

Darstellung funktionaler Einheiten in biologischen Netzwerken. Ein ähnlicher Ansatz wird in [92] umgesetzt, hier werden ein gitterbasiertes und kräftebasiertes Verfahren kombiniert sowie Kompartimente berücksichtigt. Der in [130] vorgeschlagene Algorithmus nutzt ein auf Bedingungen basierendes Layout-Verfahren. Die Autoren heben hervor, dass durch diesen Ansatz (1) die speziellen Anforderungen für die Darstellung biologischer Netzwerke besser erfüllt werden können, (2) ein Anwender das Layout durch die Definition weiterer Bedingungen anpassen kann und (3) eine interaktive Navigation und Exploration von Netzwerken sehr gut unterstützt wird, da die *mental map* einer Darstellung erhalten bleibt. Die *mental map* (kognitive Karte) [45, 114] einer Darstellung ist die Wahrnehmung oder das Verständnis dieser Darstellung durch den Leser⁷. In einem interaktiven System oder bei der Anpassung der Darstellung eines biologischen Netzwerkes sollte die *mental map* deshalb so weit wie möglich erhalten bleiben [4].

In der vorliegenden Arbeit wird [43] und [130] folgend ein auf Bedingungen basierendes Layout-Verfahren in Kombination mit einem automatischen Verfahren zur Bestimmung orthogonaler Kantentpfade [152] genutzt. Dieses Verfahren zur Bestimmung der Kantentpfade nutzt eine Technik, die einen orthogonalen Sichtbarkeits-Graphen basierend auf einer Menge von rechteckigen Objekten (den Knoten) bildet. Dieser Graph wird dann verwendet, um für die Kanten orthogonale Pfade minimaler Länge und mit minimaler Anzahl an Stützpunkten zu berechnen. Im Rahmen der vorliegenden Arbeit wurde das Verfahren von den Autoren in einer Kooperation um die Möglichkeit zur Definition von Kontrollpunkten erweitert. Kontrollpunkte definieren Positionen im Layout, die von Kantentpfaden passiert werden müssen. Die Verfahren aus [43] und [152] sind in der Bibliothek *Adaptagrams*⁸ implementiert.

2.4 Zusammenfassung

In diesem Kapitel wurden Grundlagen aus der Biologie, der Mathematik und der Informatik sowie Standardisierungsansätze aus der Systembiologie dargestellt, die zum Verständnis der Arbeit beitragen. Es wurden Grundbegriffe aus der Biologie zusammengefasst, grundlegende zelluläre Prozesse beschrieben und wichtige biologische Netzwerke erklärt. Darüber hinaus wurde ein kurzer Überblick über die Graphentheorie und das Zeichnen von Graphen sowie Möglichkeiten zur Darstellung biologischer Netzwerke gegeben. Für umfängliche Betrachtungen der jeweiligen Themengebiete wurde auf die verwendete Primärliteratur verwiesen.

⁷Zum Begriff der kognitiven Karte siehe z. B. auch [106] und [142].

⁸<http://www.adaptagrams.org/>

Kapitel 3

Systems Biology Graphical Notation

Während in Bereichen wie z. B. der Elektrotechnik (Grafische Symbole für Schaltpläne, IEC (International Electrotechnical Commission) 60617 – Graphical symbols for diagrams¹) oder der Informatik (UML (Unified Modeling Language)²) vereinheitlichte Darstellungen schon seit langem Anwendung finden, werden für die Visualisierung biologischen Wissens oft unterschiedliche Darstellungen verwendet. In [100] beschreiben die Autoren, dass in der Literatur für die Darstellung $X \longrightarrow Y$ acht verschiedene Beschreibungen und für die Beschreibung „X inhibiert Y“ (X *inhibits* Y) neun unterschiedliche Darstellungen gefunden werden konnten. In einer Umfrage [90] gaben 40% der Teilnehmer an, dass sie grafische Darstellungen biologischen Wissens in einer oftmals durch die verwendete Software-Anwendung bestimmten Notation zeichnen. Die Teilnehmer äußern ebenfalls, dass ein Bedarf an einer standardisierten Notation speziell für die Visualisierung biochemischer Netzwerke besteht.

Die Systems Biology Graphical Notation (SBGN)³ definiert eine solche vereinheitlichte Notation für die grafische Darstellung und den Austausch komplexen biologischen Wissens in einer übersichtlichen und eindeutigen Weise [100]. Das SBGN-Projekt wurde 2005 von Hiroaki Kitano (The Systems Biology Institute, Tokio, Japan) initiiert [88]. SBGN wird seitdem von einer internationalen Community, bestehend aus Biologen, Biochemikern und Informatikern, entwickelt und unterstützt. Dabei wurden die drei komplementären Sprachen *Process Description* (PD), *Entity Relationship* (ER) und *Activity Flow* (AF) definiert, die eine Menge von Elementen (genannt *glyphs*) bereitstellen und es ermöglichen, biochemische und zelluläre Prozesse und Netzwerke unter Berücksichtigung unterschiedlicher Aspekte und in unterschiedlichen Detailstufen als SBGN-Karten darzustellen. Die drei Sprachen bieten dabei verschiedene Sichten auf die zu Grunde liegenden biologischen Systeme, die Entscheidung für eine der drei Sprachen basiert auf der Art des darzustellenden biologischen Wissens und/oder der Menge des zur Verfügung stehenden biologischen Wissens [100]. Ein Beispiel für eine SBGN-Karte in jeder der drei Sprachen ist in Abbildung 3.1 zu sehen.

¹<http://std.iec.ch/iec60617>

²<http://www.uml.org/>

³<http://sbgn.org/>

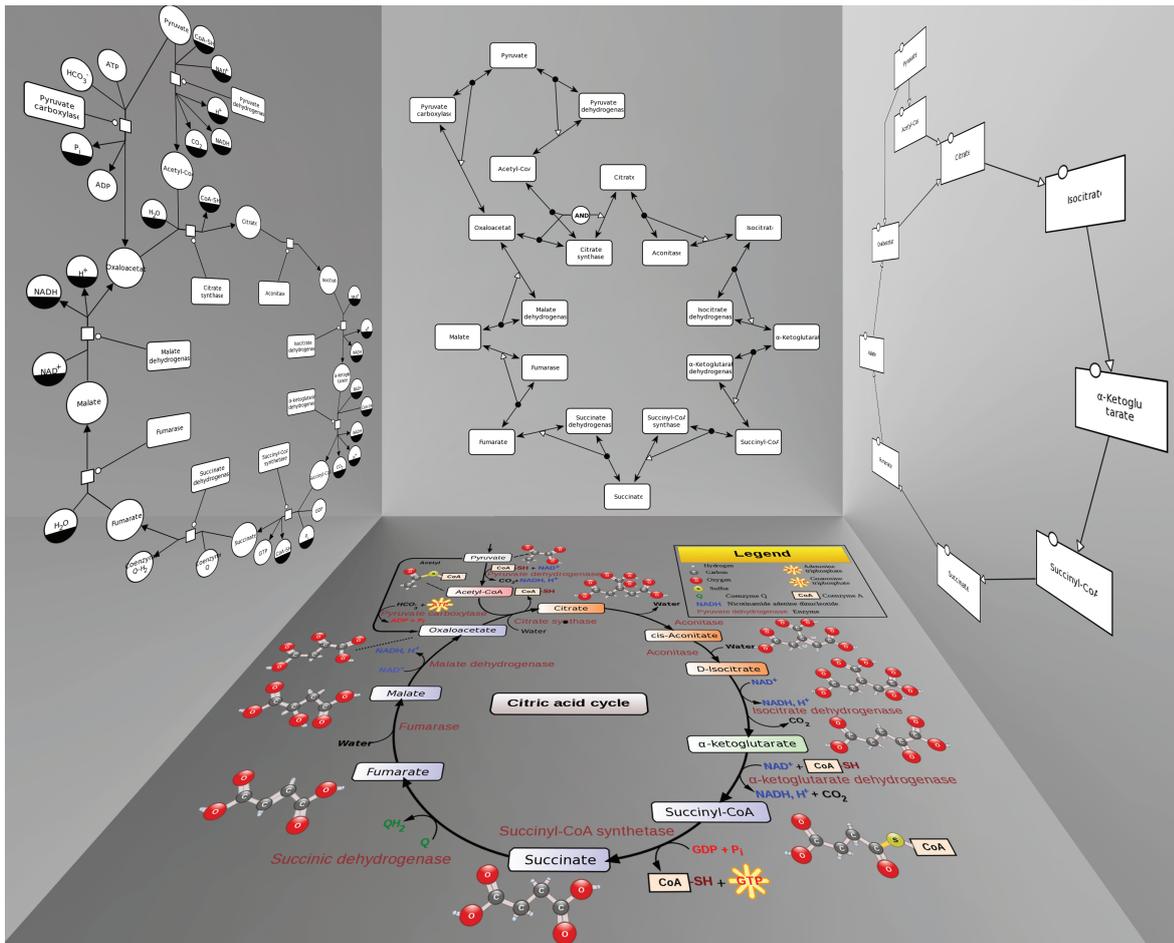


Abbildung 3.1: Darstellung des Zitratzyklus [97] durch Anwendung aller drei SBGN-Sprachen in unterschiedlichen Sichten. Unten: schematische Darstellung des Zitratzyklus [150], links: Darstellung in der *Process Description*-Sprache, oben: Darstellung in der *Entity Relationship*-Sprache, rechts: Darstellung in der *Activity Flow*-Sprache.

Die *Process Description*-Sprache kann genutzt werden, um Prozesse darzustellen, die (1) Entitäten in andere Entitäten umwandeln, (2) den Zustand von Entitäten ändern oder (3) die Lokalisierung von Entitäten ändern. Die Sprache folgt der *Process Diagram Notation* (PDN) [87, 88] und orientiert sich an Ideen in [117]. Die *Entity Relationship*-Sprache kann genutzt werden, um Interaktionen und Beziehungen zwischen Entitäten und die Regeln für diese Interaktionen und Beziehungen darzustellen. Die Sprache basiert auf der *Molecular Interaction Map* (MIM) Notation [91, 105]. Die *Activity Flow*-Sprache kann die Einflüsse, die biologische Aktivitäten aufeinander haben in einer vereinfachten Weise darstellen, die ähnlich zu Darstellungen in der Literatur ist [100].

Zu Beginn des SBGN-Projektes wurden von der Community verschiedene Grundprinzipien für die Entwicklung festgelegt [100], von denen einige wichtige nachfolgend aufgeführt werden.

SBGN soll

- frei verfügbar sein,
- syntaktisch und semantisch konsistent und eindeutig sein,
- visuell konsistent und prägnant sein,
- die Anzahl an Elementen auf ein Minimum beschränken, um das Erlernen zu vereinfachen und
- die Darstellung einer Vielzahl biologischer Entitäten, ihrer Eigenschaften und ihrer Interaktionen unterstützen.

Eine Darstellung in SBGN wird als Karte (*map*) bezeichnet, die Gesamtheit der SBGN-Elemente wird allgemein *glyphs* genannt, die Informationen symbolisch und nonverbal vermitteln. SBGN-Karten werden formal auch als Graphen bezeichnet und es werden Konzepte aus der Graphentheorie verwendet: eine SBGN-Karte ist ein Graph mit einer Menge von Knoten, die durch Kanten verbunden sind. Für die Kanten wird zusätzlich auch der Begriff *arcs* verwendet [100]. Im Rahmen der vorliegenden Arbeit werden Darstellungen als *SBGN-Karte* bezeichnet und für die SBGN-Elemente werden statt *glyph* und *arc* die Begriffe *Knoten* und *Kante* genutzt. Die Systems Biology Graphical Notation definiert, wie die Knoten und Kanten miteinander verbunden werden können, spezifiziert aber nicht das genaue Layout der SBGN-Karten. Die Semantik einer Karte ist unabhängig von der Position der SBGN-Elemente in der Karte. Ebenso haben detaillierte Grafikattribute für die Elemente wie z. B. Farben, Liniestärken, Schriftgrößen usw. keinen Einfluss auf die Semantik, können aber genutzt werden, um unabhängig von SBGN bestimmte Informationen hervorzuheben [100]. Vielen SBGN-Elementen ist ein SBO-Term zugeordnet, dadurch lässt sich SBGN mit anderen Standards der Systembiologie wie z. B. SBML integrieren.

Die drei Sprachen sind detailliert in den Spezifikationen beschrieben [101, 112, 116] (derzeit Level 1 Version 1.3 für die *Process Description*-Sprache, Level 1 Version 1.2 für die *Entity Relationship*-Sprache und Level 1 Version 1.0 für die *Activity Flow*-Sprache). Im folgenden Abschnitt wird ein Überblick über den SBGN-Entwicklungsprozess gegeben. Die nachfolgenden drei Abschnitte führen die SBGN-Sprachen ein, dabei wird die *Activity Flow*-Sprache etwas ausführlicher beschrieben. Für die *Process Description*- und die *Entity Relationship*-Sprache sind Erläuterungen in ähnlicher Form in den Spezifikationen zu finden. Anschließend werden in einem Abschnitt die SBGN-Regeln zusammengefasst. Am Ende des Kapitels werden in zwei Übersichten SGBN-Repositorien und Software-Anwendungen, die SBGN unterstützen, vorgestellt. Die Ausführungen in den folgenden Abschnitten basieren, wenn nicht anders gekennzeichnet, auf [100, 101, 112, 116] oder sind der SBGN-Webseite⁴ entnommen.

⁴<http://sbgn.org/>

3.1 Der SBGN-Entwicklungsprozess

Wie bereits dargestellt, ist das SBGN-Projekt ein internationales Community-basiertes Projekt, an dem Biologen, Biochemiker und Informatiker von zahlreichen Universitäten und Instituten und aus der Industrie mitwirken. Das Projekt ist offen und jeder Interessierte kann sich beteiligen.

Die Entwicklung von SBGN findet im Rahmen von dedizierten SBGN-Tagungen⁵, im Rahmen von Tagungen der COMBINE-Initiative ('COmputational Modeling in BIology' Network)⁶ und durch Diskussion auf der Mailingliste⁷ statt. Informationen zum aktuellen Stand der Entwicklung und zu den Ergebnissen von Entscheidungsprozessen innerhalb der Community sind auf der SBGN-Webseite zu finden.

Das SBGN-Projekt wird von fünf Redakteuren (*Editors*) koordiniert, die von der Community jeweils für einen Zeitraum von drei Jahren gewählt werden, die derzeit aktiven Redakteure sind Huaiyu Mi (University of Southern California, Los Angeles, USA), Stuart L. Moodie (Eight Pillars Ltd, Edinburg, Schottland), Falk Schreiber (Monash University, Clayton, Australien und Martin-Luther-Universität Halle-Wittenberg), Anatoly Sorokin (Institute of Cell Biophysics RAS, Puschtschino, Russland) und Tobias Czauderna (Leibniz-Institut für Pflanzengenetik und Kulturpflanzenforschung (IPK) Gatersleben). Den Redakteuren steht ein wissenschaftlicher Beirat beratend zur Seite, der bei Problem steuernd eingreifen kann ohne jedoch Entscheidungen zu treffen.

Die SBGN-Community hat frühzeitig entschieden, die Entwicklung von SBGN in *Level* und Versionen zu unterteilen, da eine ideale und allumfassende Definition der drei Sprachen in einem Entwicklungsschritt schwierig bis unmöglich erschien. Ein *Level* fasst eine Menge an Funktionalität zusammen, auf die sich die Community verständigt hat und die für bestimmte Aufgaben und Ziele verwendet werden kann. Versionen beinhalten kleinere Änderungen und Korrekturen.

Das Ergebnis eines Schrittes im Entwicklungsprozess sind die aktuellen Versionen der Spezifikationen für die drei SBGN-Sprachen, die von den Redakteuren auf der Basis von Entscheidungen der Community verfasst werden. Die Spezifikationen enthalten detaillierte Beschreibungen der SBGN-Elemente, Anmerkungen zur Syntax, zur Semantik und zum Layout sowie Beispiele für SBGN-Karten in der jeweiligen Sprache.

Für den Austausch von SBGN-Karten zwischen Software-Anwendungen, die SBGN unterstützen, wurde 2010 das Projekt LibSBGN initiiert (siehe auch Kapitel 7).

⁵<http://sbgn.org/Events>

⁶<http://co.mbine.org/events/>

⁷sbgn-discuss@caltech.edu

3.2 Die *Process Description*-Sprache

Die *Process Description*-Sprache (PD) kann genutzt werden, um alle Prozesse, die zwischen biologischen Entitäten stattfinden können, darzustellen. Eine PD-Karte zeigt z. B. wie Entitäten in andere Entitäten umgewandelt werden, wie sich der Zustand von Entitäten ändert oder wie die Lokalisierung von Entitäten geändert wird. Die Abbildungen sind eindeutig und die zeitliche Abfolge von Prozessen wird berücksichtigt. PD-Karten eignen sich besonders zur Darstellung von Stoffwechselnetzwerken. Ein Nachteil der *Process Description*-Sprache ist, dass Entitäten, die in unterschiedlichen Zuständen auftreten, in entsprechender Anzahl auf einer Karte gezeigt werden und Karten dadurch sehr groß werden können.

Die SBGN *Process Description*-Sprache definiert sechs Hauptklassen von Elementen: (1) *Entity Pool Nodes*, (2) *Process Nodes*, (3) *Container Nodes*, (4) *Reference Nodes*, (5) *Connecting Arcs* und (6) *Logical Operators*. Die Referenzkarte der *Process Description*-Sprache, die alle Elemente in einer Übersicht enthält, ist in Abbildung 3.2 dargestellt.

Entity Pool Nodes Ein *Entity Pool* beschreibt eine Menge von Entitäten, die nicht voneinander unterschieden werden können. So formen z. B. alle molekularen Entitäten, die in einem Prozess dieselbe Funktion haben, einen *Entity Pool*. Es sind verschiedene Typen von *Entity Pool Nodes* definiert: *Unspecified Entity*, *Simple Chemical*, *Macromolecule*, *Nucleic Acid Feature*, *Perturbing Agent*, *Source* und *Sink* sowie *Complex*. Zusätzlich sind *Multimer Simple Chemical*, *Multimer Macromolecule*, *Multimer Nucleic Acid Feature* und *Multimer Complex* definiert. Ein *Multimer* beschreibt den Zusammenschluss mehrerer identischer *Entity Pools* durch eine nicht kovalente Bindung.

Process Nodes *Process Nodes* repräsentieren Prozesse, die einen oder mehrere *Entity Pools* in einen oder mehrere *Entity Pools* umwandeln. Es gibt sechs verschiedene *Process Nodes*: *Process*, *Omitted Process*, *Uncertain Process*, *Association*, *Dissociation* und *Phenotype*.

Container Nodes Die SBGN *Process Description*-Sprache definiert das *Compartment*, eine logische oder physische Struktur, die andere SBGN-Elemente enthält.

Reference Nodes Die SBGN *Process Description*-Sprache definiert *Submap*, *Tag* und *Terminal*. Eine *Submap* kann eine Menge von Prozessen einkapseln, die nicht auf der primären SBGN-Karte dargestellt werden. *Terminal*-Knoten auf *Submap*-Knoten sind zur Referenz mit *Entity Pool Node*-Knoten der Karte verbunden. Wenn der Inhalt einer *Submap* als separate SBGN-Karte verfügbar ist, referenzieren *Tag*-Knoten entsprechende *Entity Pool Node*-Knoten der primären SBGN-Karte.

Connecting Arcs Die *Connecting Arcs* beschreiben die Kanten, welche die Knoten verbinden, die *Entity Pool Nodes*, *Process Nodes*, *Reference Nodes* und *Logical Operators* reprä-

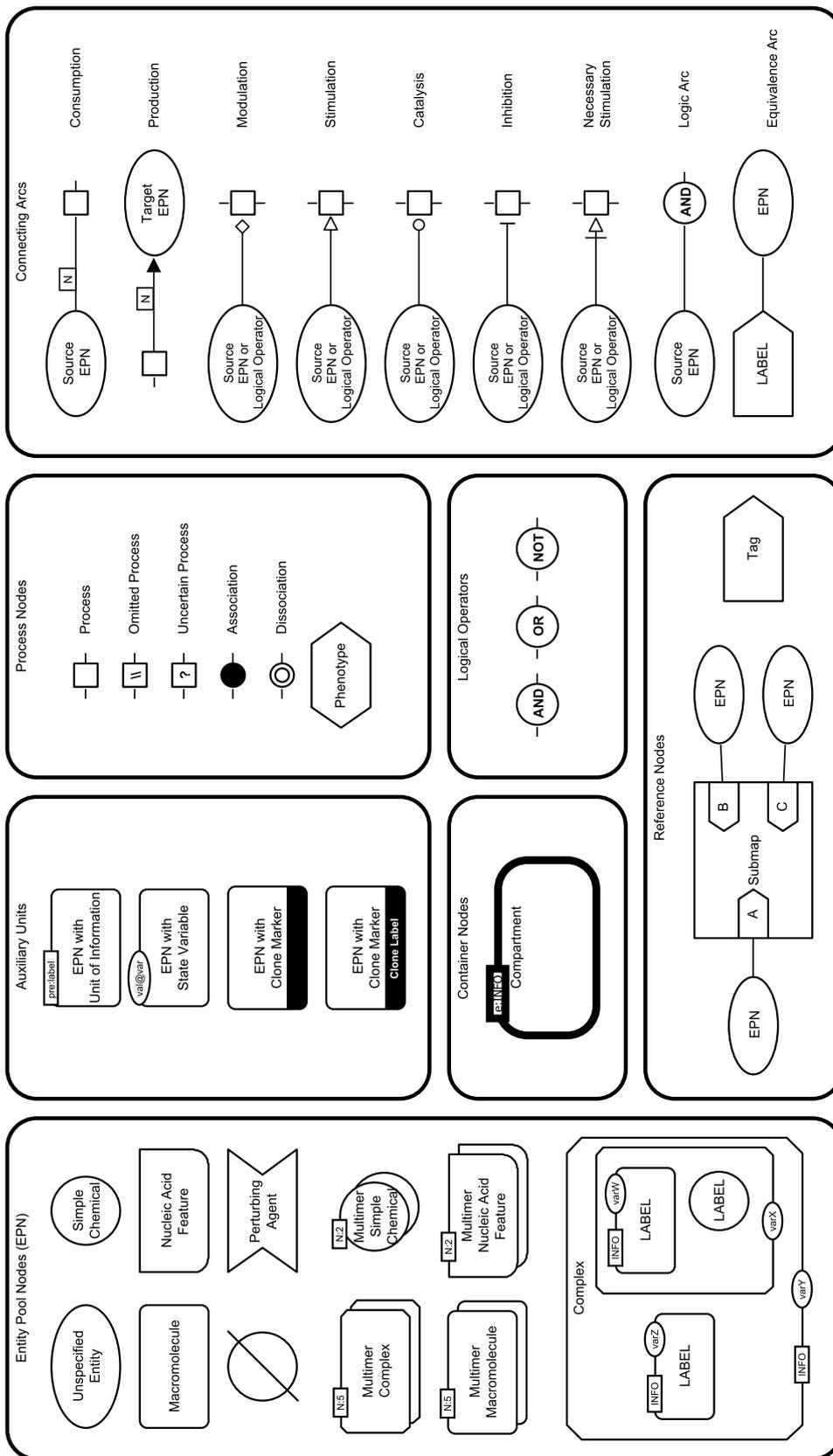


Abbildung 3.2: Referenzkarte der SBGN *Process Description*-Sprache, übernommen und angepasst aus [116].

sentieren. Es sind *Consumption*, *Production*, *Modulation*, *Stimulation*, *Catalysis*, *Inhibition*, *Necessary Stimulation*, *Logic Arc* und *Equivalence Arc* definiert.

Logical Operators Die SBGN *Process Description*-Sprache definiert die drei *Logical Operators* *AND*, *OR* und *NOT*.

Auxiliary Units *Auxiliary Units* dekorieren oder erweitern Knoten durch zusätzliche Informationen. Die SBGN *Process Description*-Sprache definiert *Unit of Information* (zusätzliche Information), *State Variable* (Beschreibung eines Zustandes) und *Clone Marker* (Anzeige der Duplizierung auf der SBGN-Karte).

3.3 Die *Entity Relationship*-Sprache

Die *Entity Relationship*-Sprache (ER) kann genutzt werden, um Interaktionen und Beziehungen zwischen Entitäten und die Regeln für diese Interaktionen und Beziehungen darzustellen. Die Abbildungen sind eindeutig. Da die Interaktionen und Beziehungen unabhängig voneinander sind, kann eine zeitliche Abfolge jedoch nicht gezeigt werden. ER-Karten eignen sich besonders zur Darstellung von Signalnetzwerken und Protein-Protein-Interaktionsnetzwerken. Entitäten werden nur einmal auf einer Karte abgebildet, auch wenn sie in unterschiedlichen Zuständen auftreten.

Die SBGN *Entity Relationship*-Sprache definiert drei Hauptklassen von Elementen: (1) *Entity Nodes* (*Interactors* und *Logical Operators*), (2) *Statements* und (3) *Influences*. Die Referenzkarte der *Entity Relationship*-Sprache, die alle Elemente in einer Übersicht enthält, ist in Abbildung 3.3 dargestellt.

Interactors *Interactors* sind an Interaktionen beteiligt. Die SBGN *Entity Relationship*-Sprache definiert *Entity* und *Outcome*.

Statements *Statements* sind Zuweisungen und Interaktionen, die das Ziel von Einflüssen (*Influences*) sind. Es sind *Assignment*, *Interaction* und *Phenotype* definiert.

Influences *Influences* repräsentieren den Einfluss von *Interactors* und dem *Perturbing Agent* auf *Statements* oder andere *Influences*. Für die *Entity Relationship*-Sprache sind *Modulation*, *Stimulation*, *Necessary Stimulation*, *Absolute Stimulation*, *Inhibition*, *Absolute Inhibition* und *Logic Arc* beschrieben.

Logical Operators Die SBGN *Entity Relationship*-Sprache definiert die vier *Logical Operators* *AND*, *OR*, *NOT* und *Delay*. *Delay* bedeutet, dass ein Einfluss von einem *Interactor* verzögert erzeugt wird.

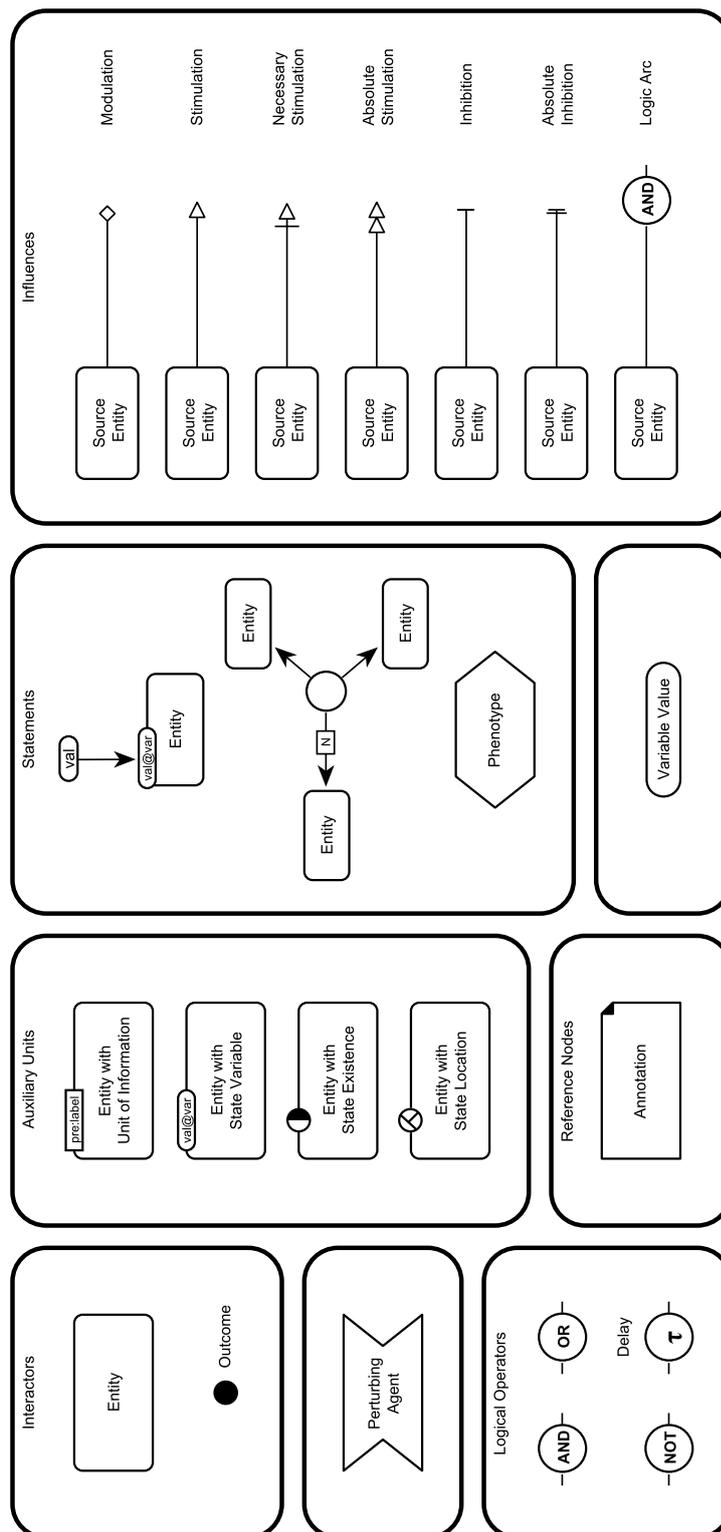


Abbildung 3.3: Referenzkarte der SBN *Entity Relationship*-Sprache, übernommen und angepasst aus [101].

Weitere Knoten Die *Entity Relationship*-Sprache definiert die weiteren Knoten *Perturbing Agent*, *Variable Value* und *Annotation*, die keiner der anderen Klassen zugeordnet sind.

Auxiliary Units *Auxiliary Units* dekorieren oder erweitern Knoten durch zusätzliche Informationen. Die SBGN *Entity Relationship*-Sprache definiert *Unit of Information* (zusätzliche Information), *State Variable* (Beschreibung eines Zustandes), *Existence* (Existenz) und *Location* (Lokalisierung in einem Kompartiment). *Existence* und *Location* sind spezielle Zustände.

3.4 Die *Activity Flow*-Sprache

Die *Activity Flow*-Sprache (AF) kann die Einflüsse, die biologische Aktivitäten aufeinander haben, auf einer konzeptionellen Ebene darstellen. Die Abbildungen sind abstrakt und nicht eindeutig, die zu Grunde liegenden Prozesse sind in einer Abbildung nicht berücksichtigt. Es wird jedoch die zeitliche Abfolge, in der sich biologische Aktivitäten beeinflussen, berücksichtigt. AF-Karten können genutzt werden, wenn biologisches Wissen unvollständig ist, detailliertes biologisches Wissen fehlt, oder biologisches Wissen vereinfacht gezeigt werden soll.

Die SBGN *Activity Flow*-Sprache definiert fünf Hauptklassen von Elementen: (1) *Activity Nodes*, (2) *Container Nodes*, (3) *Reference Nodes*, (4) *Modulating Arcs* und (5) *Logical Operators*. Die Referenzkarte der *Activity Flow*-Sprache, die alle Elemente in einer Übersicht enthält, ist in Abbildung 3.4 dargestellt.

Activity Nodes Eine *Activity Node* repräsentiert die Aktivität einer Entität und nicht die Entität selbst. Neben einer *Activity Node* für physische Entitäten definiert die *Activity Flow*-Sprache auch die beiden konzeptionellen *Activity Nodes* *Perturbation* und *Phenotype*.

- *Biological Activity*: Aktivität einer biologischen Entität.
- *Perturbation*: externer Einfluss, der Aktivitäten beeinflussen kann.
- *Phenotype*: beschreibt einen Phänotyp oder den Einfluss auf andere biologische Prozesse, die unabhängig von der aktuellen SBGN-Karte gezeigt werden.

Container Nodes Die SBGN *Activity Flow*-Sprache definiert das *Compartment*, eine logische oder physische Struktur, die andere SBGN-Elemente enthält.

Reference Nodes Die SBGN *Activity Flow*-Sprache definiert *Submap*, *Tag* und *Terminal*. Eine *Submap* kann eine Menge von Aktivitäten einkapseln, die nicht auf der primären SBGN-Karte dargestellt werden. *Terminal*-Knoten auf *Submap*-Knoten sind zur Referenz mit *Biological Activity*-Knoten der Karte verbunden. Wenn der Inhalt einer *Submap* als separate SBGN-Karte verfügbar ist, referenzieren *Tag*-Knoten entsprechende *Biological Activity*-Knoten der primären SBGN-Karte.

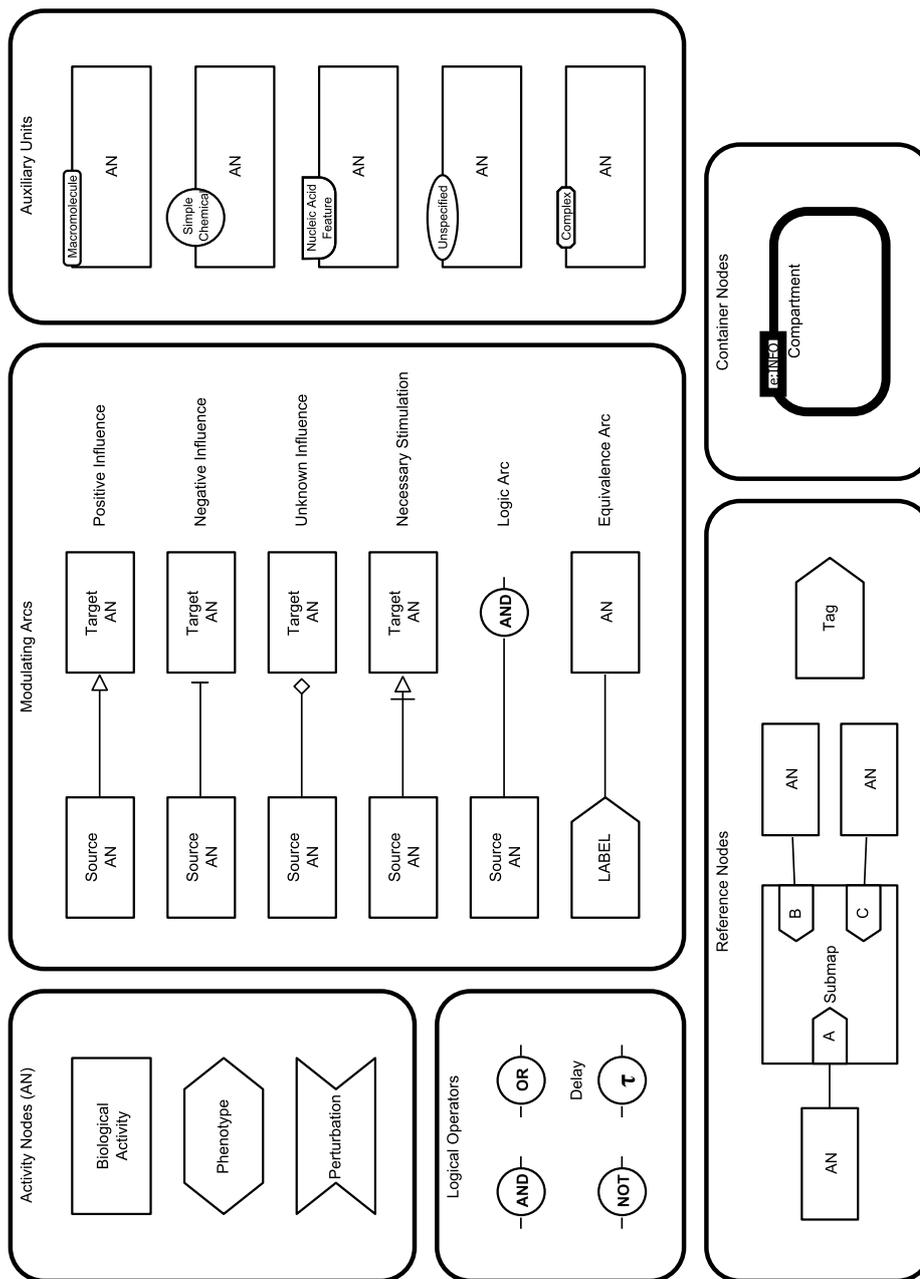


Abbildung 3.4: Referenzkarte der SBGN *Activity Flow*-Sprache, übernommen und angepasst aus [112].

Modulating Arcs Die *Modulating Arcs* beschreiben die Kanten, welche die Knoten verbinden, die *Activity Nodes*, *Reference Nodes* und *Logical Operators* repräsentieren.

- *Positive Influence*: eine Aktivität wird positiv oder aktivierend beeinflusst.
- *Negative Influence*: eine Aktivität wird negativ oder hemmend beeinflusst.
- *Unknown Influence*: eine Aktivität wird beeinflusst, aber es sind keine detaillierten Informationen über den Einfluss vorhanden.

- *Necessary Stimulation*: die Stimulation ist notwendig, damit die Aktivität selbst aktiv sein kann.
- *Logic Arc*: verbindet einen *Biological Activity*-Knoten oder *Logical Operator*-Knoten mit einem *Logical Operator*-Knoten.
- *Equivalence Arc*: verbindet einen *Biological Activity*-Knoten mit einem *Tag*-Knoten.

Logical Operators Die SBGN *Activity Flow*-Sprache definiert die vier *Logical Operators* *AND*, *OR*, *NOT* und *Delay*. *Delay* bedeutet, dass ein Einfluss von einer *Activity Node* verzögert erzeugt wird.

Auxiliary Units *Auxiliary Units* dekorieren oder erweitern Knoten durch zusätzliche Informationen. Die SBGN *Activity Flow*-Sprache definiert die *Unit of Information*, die *Biological Activity*-Knoten um eine Information zur zu Grunde liegenden Entität (*Unspecified*, *Simple Chemical*, *Macromolecule*, *Nucleic Acid Feature*, *Complex*) und *Compartment*-Knoten um zusätzliche Information erweitern kann.

3.5 Regeln für SBGN-Karten

In den letzten drei Abschnitten wurden die SBGN-Sprachen und die Elemente der Sprachen in einem Überblick eingeführt. Neben der Beschreibung der einzelnen SBGN-Elemente ist für eine visuelle Beschreibung biologischer Systeme ein detaillierter Satz von Regeln notwendig, damit (1) eine SBGN-Karte einem Leser genau und einfach vermitteln kann, was dargestellt ist und (2) eine Unterstützung für SBGN durch Software-Anwendungen möglich ist.

Jede der drei Spezifikationen enthält deshalb zwei Abschnitte, in denen die Konzepte, Regeln zur Syntax und Semantik sowie Richtlinien zum Layout für die jeweilige Sprache beschrieben sind. Im Folgenden wird eine Übersicht über die Regeln gegeben, für die *Activity Flow*-Sprache werden die Regeln zur Syntax und Semantik sowie Richtlinien zum Layout in Auszügen dargestellt, für die *Process Description*- und die *Entity Relationship*-Sprache sind sie in ähnlicher Form in den Spezifikationen dargelegt.

3.5.1 Konzepte

Jede der drei SBGN-Sprachen nutzt spezielle Abstraktionen, um biologische Prozesse und Systeme zu beschreiben.

Die *Process Description*-Sprache beschreibt Prozesse, die auf Mengen (*pools*) biologischer Entitäten einwirken. Die Entitäten sind typischerweise Moleküle, Ausnahmen sind aber möglich (z.B. *Perturbing Agent*), und die Prozesse sind typischerweise eine Kombination einer oder mehrerer biochemischer Reaktionen. Prozesse können durch Entitäten gesteuert (reguliert) werden und Entitäten können aus einer Quelle (*Source*) erzeugt werden oder in eine Senke (*Sink*) abgebaut werden.

Die *Entity Relationship*-Sprache beschreibt biologische Interaktionen zwischen biologischen Entitäten. Ein Interaktor, z. B. ein Molekül, beeinflusst das Verhalten anderer Interaktoren durch Beziehungen zu diesen, *Entity Relationships* können als eine Liste unabhängiger Regeln betrachtet werden, die Einflüsse zwischen Interaktoren beschreiben.

Die *Activity Flow*-Sprache beschreibt biologische Aktivitäten von biologischen Entitäten. Eine biologische Aktivität kann durch andere biologische Aktivitäten beeinflusst werden oder selbst andere biologische Aktivitäten beeinflussen. Eine AF-Karte stellt den Fluss der Aktivitäten von einer Entität zu einer anderen Entität oder innerhalb derselben Entität dar. Der den Einflüssen zu Grunde liegende Mechanismus ist entweder nicht bekannt oder wird nicht dargestellt. Wenn der Mechanismus bekannt ist, kann er als PD-Karte oder als ER-Karte dargestellt werden.

3.5.2 Syntax

Die Syntax der drei SBGN-Sprachen definiert, wie die SBGN-Elemente zu einer validen SBGN-Karte zusammengefügt werden können, sie definiert welche SBGN-Knoten über welche SBGN-Kanten miteinander verbunden werden dürfen. Dies wird für jede der Sprachen durch eine Inzidenzmatrix [38] beschrieben, in Tabelle 3.1 ist die Inzidenzmatrix für AF-Karten dargestellt. Für die *Process Description*- und die *Activity Flow*-Sprache wird zusätzlich noch definiert, welche Knoten aufeinander gezeichnet werden können (*containment*). Darüber hinaus gibt es für alle drei Sprachen noch weitere syntaktische Regeln, die sich nicht aus der Inzidenzmatrix ergeben.

So ist in der Spezifikation für AF zusätzlich beschrieben, dass alle Elemente der Sprache auf einen *Compartment*-Knoten gezeichnet werden dürfen (*containment*)⁸ und dass ein *Biological Activity*-Knoten maximal eine *Unit of Information* haben darf.

3.5.3 Semantik

Im Kontext von SBGN beschreiben semantische Regeln die Bedeutung einer SBGN-Karte. Die Semantik ist wichtig, damit ein Autor eine SBGN-Karte erstellen kann, die sein Verständnis eines biologischen Prozesses oder Systems darstellt, und ein Leser eine SBGN-Karte ohne zusätzliche Hilfe verstehen kann.

Für die *Activity Flow*-Sprache sind unter der Annahme, dass eine biologische Aktivität eine Rate hat, z. B. folgende semantische Regeln definiert:

- Eine biologische Aktivität gehört nur zu einem Kompartiment, auch wenn der entsprechende *Biological Activity*-Knoten mehrere *Compartment*-Knoten überlappt. Die Entscheidung über die Zugehörigkeit liegt beim Autor der Karte oder bei der Software-Anwendung.

⁸Entgegen der Spezifikation darf ein *Compartment*-Knoten auf einen *Compartment*-Knoten gezeichnet werden (SBGN-Redakteure, persönliche Mitteilung).

Kante \ Knoten	<i>Biological Activity</i>	<i>Perturbation</i>	<i>Phenotype</i>	<i>Terminal</i>	<i>Tag</i>	<i>AND</i>	<i>OR</i>	<i>NOT</i>	<i>Delay</i>
<i>Positive Influence</i>	-1 & +1	-1	+1	-	-	-1 (1)	-1 (1)	-1 (1)	-1 (1)
<i>Negative Influence</i>	-1 & +1	-1	+1	-	-	-1 (1)	-1 (1)	-1 (1)	-1 (1)
<i>Unknown Influence</i>	-1 & +1	-1	+1	-	-	-1 (1)	-1 (1)	-1 (1)	-1 (1)
<i>Necessary Stimulation</i>	-1 & +1	-1	+1	-	-	-1 (1)	-1 (1)	-1 (1)	-1 (1)
<i>Logic Arc</i>	-1	-	-	-	-	+1	+1	+1 (1)	+1
<i>Equivalence Arc</i>	-1	-	-	+1	+1	-	-	-	-

Tabelle 3.1: Inzidenzmatrix für AF-Karten. Die Matrix beschreibt, welche Knoten mit welchen Kanten verbunden werden dürfen, übernommen und angepasst aus [112] („-1“ Kante beginnt an Knoten, „+1“ Kante endet an Knoten, „-“ Knoten und Kante können nicht verbunden werden, Zahlen in Klammern geben die maximale Anzahl Kanten an).

- Bei *Compartment*-Knoten, die nebeneinander oder aufeinander gezeichnet werden, müssen die Kompartimente nicht physisch nebeneinander lokalisiert sein bzw. das eine Kompartiment muss das andere Kompartiment nicht physisch enthalten.
- Durch eine *Positive Influence* wird die Rate einer biologischen Aktivität erhöht.
- Durch eine *Negative Influence* wird die Rate einer biologischen Aktivität verringert.
- Bei einer *Unknown Influence* ist der Effekt und damit die Änderung der Rate einer biologischen Aktivität unbekannt.
- Es kann nur eine *Necessary Stimulation*-Kante an einem *Biological Activity*-Knoten enden, mehrere *Necessary Stimulation*-Kanten müssen explizit über einen *AND*- oder einen *OR*-Knoten verknüpft werden.

3.5.4 Layout-Richtlinien

Die Layout-Richtlinien beschreiben Anforderungen und Empfehlungen, wie eine SBGN-Karte dargestellt werden soll, sie beschreiben das Erscheinungsbild und die Ästhetik einer Karte. Anforderungen an das Layout müssen erfüllt sein und Empfehlungen für das Layout sollen erfüllt sein. Die Anforderungen an das Layout einer AF-Karte sind z. B.:

- Knoten dürfen sich nicht überlappen mit Ausnahme von *Compartment*-Knoten und *Biological Activity*-Knoten, die *Compartment*-Knoten überlappen.

- Kreuzt eine Kante einen Knoten, dann muss die Kante auf dem Knoten gezeichnet werden.
- Kanten dürfen die Randlinien eines Knotens nicht überlappen.
- Knoten müssen horizontal oder vertikal ausgerichtet sein.

Empfehlungen für das Layout einer AF-Karte sind z. B.:

- Der Knotenname eines Knotens soll horizontal ausgerichtet sein und wenn möglich komplett innerhalb des Knotens dargestellt werden.
- Kanten sollen Knoten nicht kreuzen.
- Die Anzahl Kantenkreuzungen soll minimiert werden.

3.6 Repositorien und Software-Anwendungen für SBGN-Karten

Die Akzeptanz und Anwendung von SBGN hängt u. a. auch von der Verfügbarkeit von Repositorien und Software-Anwendungen für SBGN-Karten ab.

In Tabelle 3.2 ist ein Überblick über zwölf Repositorien für SBGN-Karten dargestellt. Alle Repositorien stellen SBGN-Karten in der *Process Description*-Sprache bereit und ein Repository stellt zusätzlich SBGN-Karten in der *Activity Flow*-Sprache bereit. Derzeit gibt es noch kein Repository für Karten in der *Entity Relationship*-Sprache. Der Download von SBGN-Karten im SBGN-ML Format (siehe auch Kapitel 7) ist in vier Repositorien möglich.

Tabelle 3.3 gibt einen Überblick über Software-Anwendungen, die SBGN unterstützen. Die Anwendungen wurden hinsichtlich verschiedener Kriterien evaluiert: (1) Betrachter oder Editor für SBGN-Karten, (2) Unterstützung der drei SBGN-Sprachen *Process Description*, *Entity Relationship* und *Activity Flow*, (3) Übersetzung von KEGG-Karten in SBGN PD-Karten (siehe auch Kapitel 4), (4) Umwandlung zwischen SBGN-Sprachen (siehe auch Kapitel 5), (5) Unterstützung der SBGN Bricks (siehe auch Kapitel 6) und (6) Austausch der SBGN-Karten im SBGN-ML Format (siehe auch Kapitel 7).

In der Literatur konnten 25 Anwendungen gefunden werden, die SBGN unterstützen. Von diesen 25 untersuchten Anwendungen sind elf Anwendungen Betrachter für SBGN-Karten und 14 Anwendungen sind Editors für SBGN-Karten, die es ermöglichen Karten zu erstellen oder zu bearbeiten. 23 Anwendungen unterstützen die *Process Description*-Sprache, sechs Anwendungen unterstützen die *Entity Relationship*-Sprache und sieben Anwendungen unterstützen die *Activity Flow*-Sprache. Der Austausch von SBGN-Karten im SBGN-ML Format ist mit neun Anwendungen möglich. Bei den beiden Software-Anwendungen geneXplain platform und PathwayLab handelt es sich um kommerzielle Anwendungen, alle anderen sind frei verfügbar.

Repositorium	PD	ER	AF	SBGN-ML
Atlas of Cancer Signalling Networks ⁹	+	-	-	-
AlzPathway [115]	+	-	-	-
BioModels Database ¹⁰ [26]	+	-	-	-
DC-Atlas [24]	+	-	-	-
IPAVS [135]	+	-	-	-
MetaCrop [129]	+	-	-	+
PANTHER Pathway [111]	+	-	-	+
Parkinson's disease map [52]	+	-	-	-
Path2Models [20]	+	-	+	+
Reactome [29]	+	-	-	+
RIMAS [73]	+	-	-	-
SubtiPathways [107]	+	-	-	-

Tabelle 3.2: Überblick über Repositorien für SBGN-Karten. Die Repositorien wurden hinsichtlich zweier Kriterien evaluiert: (1) SBGN-Sprachen (PD – *Process Description*, ER – *Entity Relationship*, AF – *Activity Flow*) und (2) Download der SBGN-Karten im SBGN-ML Format (siehe auch Kapitel 7) („+“ unterstützt, „-“ nicht unterstützt).

Zusammenfassend lässt sich feststellen, dass nur die fünf Software-Anwendungen Biographer, CySBGN, PathVisio, SBGN-ED und SBML Layout alle drei SBGN-Sprachen und SBGN-ML unterstützen, wobei nur Biographer, PathVisio und SBGN-ED das Erstellen und Bearbeiten von SBGN-Karten ermöglichen. Die drei Anwendungen KEGGtranslator, MGV und SBGN-ED erlauben eine Darstellung von KEGG-Karten in SBGN, der Erhalt des Layouts der KEGG-Karten ist dabei nur mit SBGN-ED möglich. SBGN-ED ist auch die einzige Software-Anwendung, die eine Umwandlung zwischen SBGN-Sprachen ermöglicht und die eine Unterstützung für die SBGN Bricks bietet.

⁹<https://acsn.curie.fr/index.html>

¹⁰Path2Models ist Teil der BioModels Database, wird hier aber separat aufgeführt.

Software-Anwendung	B/E	PD	ER	AF	KEGG ↓ SBGN	SBGN ↓ SBGN	SBGN Bricks	SBGN-ML
Arcadia [145]	B	+	-	-	-	-	-	-
Athena [25]	E	+	-	-	-	-	-	-
Beacon Pathway Editor ¹¹	E	-	-	+	-	-	-	-
BIOCHAM [22]	E	+	-	-	-	-	-	-
Biographer [96]	E	+	+	+	-	-	-	+
BioUML [95]	B	+	-	-	-	-	-	-
CellDesigner [53]	E	+	-	-	-	-	-	+
ChiBE [7]	E	+	-	-	-	-	-	-
COPASI [64]	B	+	-	-	-	-	-	-
CySBGN [60]	B	+	+	+	-	-	-	+
Dunnart [44]	E	+	-	-	-	-	-	-
EPE [134]	E	+	+	+	-	-	-	-
geneXplain platform ¹²	E	+	-	-	-	-	-	-
iPathways ¹³	B	+	-	-	-	-	-	-
JWS Online [119]	B	+	-	-	-	-	-	-
KEGGtranslator [151]	B	+	-	-	+	-	-	+
MGV [137]	B	+	-	-	+	-	-	-
Netbuilder' [148]	E	+	-	-	-	-	-	-
Omix [42]	E	-	-	-	-	-	-	+
PathVisio [143]	E	+	+	+	-	-	-	+
PathwayLab [69]	E	+	-	-	-	-	-	-
SBGN-ED [31]	E	+	+	+	+	+	+	+
SBML Layout [34]	B	+	+	+	-	-	-	+
SBMM assistant [124]	B	+	-	-	-	-	-	-
VISIBIOweb [39]	B	+	-	-	-	-	-	+

¹¹<http://bioinformatics.cs.vt.edu/beacon/>

¹²<http://genexplain.com/genexplain-platform-1>

¹³<http://www.ipathways.org/>

Tabelle 3.3: Überblick über Software-Anwendungen für SBGN-Karten. Die Anwendungen wurden hinsichtlich verschiedener Kriterien evaluiert: (1) Betrachter (B) oder Editor (E) für SBGN-Karten, (2) SBGN-Sprachen (PD – *Process Description*, ER – *Entity Relationship*, AF – *Activity Flow*), (3) Übersetzung von KEGG zu SBGN (siehe auch Kapitel 4), (4) Umwandlung zwischen SBGN-Sprachen (siehe auch Kapitel 5), (5) Unterstützung der SBGN Bricks (siehe auch Kapitel 6) und (6) Austausch der SBGN-Karten im SBGN-ML Format (siehe auch Kapitel 7) („+“ unterstützt, „-“ nicht unterstützt).

3.7 Diskussion und Zusammenfassung

Die Systems Biology Graphical Notation ist ein Standard für die visuelle Darstellung biochemischer und zellulärer Prozesse und Netzwerke in der Systembiologie. Die drei Sprachen *Process Description*, *Entity Relationship* und *Activity Flow* ermöglichen es, biologische Systeme in unterschiedlichen Detailstufen unter Berücksichtigung verschiedener Aspekte als SBGN-Karten darzustellen und bieten dabei unterschiedliche Sichten auf die zu Grunde liegenden biologischen Systeme. SBGN hat dabei zusätzlich das Ziel, den Austausch biologischen Wissens zu erleichtern. Da vielen SBGN-Elementen ein SBO-Term zugeordnet ist, lässt sich SBGN auch einfach mit anderen Standards der Systembiologie wie z. B. SBML integrieren.

Neben den bereits erwähnten Ansätzen [87, 88, 91, 105, 117], auf denen SBGN aufbaut, gab es bereits früher schon Ansätze zur Standardisierung der Visualisierung biologischer Systeme [27, 122], die sich jedoch nicht als Standard etablieren konnten [100]. Parallel zur Entwicklung von SBGN wurde mEPN (*modified Edinburgh Pathway Notation*) [50] erarbeitet, die viele Ähnlichkeiten aber auch einige Unterschiede zur *Process Description*-Sprache aufweist. Die Autoren unterstützen SBGN prinzipiell, begründen ihre Entwicklung aber auch mit Schwierigkeiten bei der Implementierung und Anwendung von SBGN in Software. Aufbauend auf SBGN wurde mSBGN (*modified Systems Biology Graphical Notation*) [108] definiert. Im Wesentlichen wurde dabei die *Process Description*-Sprache leicht angepasst, die Syntax für *Phenotype* wurde geändert und Farben werden explizit semantisch verwendet. In [70] werden Unterschiede und Gemeinsamkeiten zwischen SysML (Systems Modeling Language) und SBGN analysiert. Dabei zeigen sich gewisse Überlappungen, die bei einem Gedankenaustausch zwischen den beiden Projekten möglicherweise zu Synergieeffekten führen können.

Bisher unterstützen eine Vielzahl von Repositorien und Software-Anwendungen nur die *Process Description*-Sprache. Um die Akzeptanz und Anwendung von SBGN zu erhöhen, sollte versucht werden, die Unterstützung für die *Entity Relationship*- und *Activity Flow*-Sprache sowohl von Repositorien als auch von Software-Anwendungen zu verbessern. Ebenso sollte versucht werden, den Austausch von SBGN-Karten im SBGN-ML Format zu erweitern, um Karten besser verfügbar zu machen und einfacher zwischen Anwendungen austauschen zu können.

Kapitel 4

Übersetzung von KEGG Stoffwechselweg-Karten in SBGN PD-Karten

Biologen verwenden für ihre Arbeit häufig biologische Netzwerke wie z. B. Stoffwechselnetzwerke. Zahlreiche Datenbanken, die biologische Netzwerke bereitstellen, sind online verfügbar, für eine Übersicht siehe [10]. Die wichtigsten Datenbanken für Stoffwechselwege sind KEGG (Kyoto Encyclopedia of Genes and Genomes) PATHWAY [80], BioCyc/MetaCyc [23], Reactome [29] und PANTHER Pathway [111], sie stellen Stoffwechselwege für zahlreiche Organismen bereit.

Alle diese Datenbanken bieten grafische Darstellungen der Stoffwechselwege. Während einige allgemeine Konventionen für die Darstellung biologischer Netzwerke existieren, kann man sehr häufig beliebige Darstellungen finden [100]. Idealerweise würden die Datenbanken grafische Darstellungen in einer standardisierten Notation anbieten, so dass Wissenschaftler nicht immer wieder neue Notationen erlernen müssten [127]. Reactome und PANTHER Pathway haben bereits damit begonnen, SBGN für die grafische Darstellung von Stoffwechselweg-Karten zu verwenden. Die vielfach genutzte KEGG-Datenbank hingegen bietet Stoffwechselweg-Karten noch immer in einer eigenen Darstellung als statische Bilddateien (für ein Beispiel siehe Abbildung 4.5) und als KGML-Dateien¹ zum Download für die Verwendung in Software an.

Hier soll deshalb eine Methode vorgestellt werden, die KEGG Stoffwechselweg-Karten automatisch in SBGN *Process Description*-Karten übersetzt. Dabei soll das originale Layout der Darstellung soweit wie möglich erhalten werden und die Anforderungen und Empfehlungen zum Layout von SBGN PD-Karten sollen erfüllt werden. Die hier beschriebene Methodik besteht aus drei Schritten: (1) Übersetzung der KEGG-Karte in eine PD-Karte einschließlich notwendiger Änderungen (hinzufügen oder entfernen von Knoten und Kanten), (2) Po-

¹KGML (KEGG Markup Language, <http://www.kegg.jp/kegg/xml/>) ist das Austauschformat der KEGG-Datenbank.

sitionierung der Knoten in der PD-Karte basierend auf dem Layout der KEGG-Karte und (3) Bestimmung orthogonaler Pfade für die Darstellung der Kanten. Diese Schritte werden in den folgenden beiden Abschnitten erläutert und die Methode wird an einigen Beispielen demonstriert.

4.1 Übersetzung von KEGG in SBGN

Die bereits erwähnten KGML-Dateien dienen als Grundlage für die hier dargestellte Übersetzung von KEGG zu SBGN. Sie enthalten alle erforderlichen Informationen für die Rekonstruktion der Stoffwechselweg-Karten. In den nachfolgenden Ausführungen wird die Übersetzung metabolischer Stoffwechselwege aus der KEGG-Darstellung (KGML) in SBGN *Process Description*-Karten erläutert. Die KEGG-Datenbank enthält auch nicht-metabolische Netzwerke, die hier nicht weiter betrachtet werden.

Eine Stoffwechselweg-Karte kann als ein Graph $G = (V, E)$ mit einer Menge von Knoten V und einer Menge von Kanten E betrachtet werden. Die Knoten repräsentieren die Elemente des Stoffwechselweges und jede Kante verbindet genau zwei Knoten und stellt damit eine Beziehung zwischen Elementen des Stoffwechselweges dar. Die KEGG Stoffwechselweg-Karten enthalten mehrere Typen von Knoten und Kanten [5]².

Sei $G_{KEGG} = (V_{KEGG}, E_{KEGG})$ ein Graph mit $V_{KEGG} = V_{GP} \cup V_{OM} \cup V_{AM}$ und $E_{KEGG} = E_{MI} \cup E_{LM}$, der eine Stoffwechselweg-Karte aus der KEGG-Datenbank repräsentiert. In metabolischen Stoffwechselweg-Karten aus KEGG können drei Arten von Knoten gefunden werden: (1) *gene product, mostly protein but including RNA*-Knoten $v_{GP} \in V_{GP}$, (2) *other molecule, mostly chemical compound*-Knoten $v_{OM} \in V_{OM}$ und (3) *another map*-Knoten $v_{AM} \in V_{AM}$. Darüber hinaus werden zwei Arten von Kanten verwendet: (1) *molecular interaction or relation*-Kante $e_{MI} \in E_{MI}$ und (2) *link to another map*-Kante $e_{LM} \in E_{LM}$. Die grafische Darstellung einer KEGG Stoffwechselweg-Karte ist in Abbildung 4.1 (links, Ausschnitt des Pentose-Phosphat-Stoffwechselweges³) zu sehen. Ein *gene product*-Knoten v_{GP} wird als Rechteck gezeichnet, das eine Reaktion einer Stoffwechselweg-Karte repräsentiert; ein *other molecule*-Knoten v_{OM} wird als Kreis gezeichnet, der ein Edukt oder Produkt einer Reaktion darstellt; ein *another map*-Knoten v_{AM} wird als ein Rechteck mit abgerundeten Ecken gezeichnet, eine *molecular interaction or relation*-Kante e_{MI} wird als Kante mit einer gefüllten Pfeilspitze gezeichnet; eine *link to another map*-Kante e_{LM} wird als gestrichelte Kante mit einer leeren Pfeilspitze gezeichnet.

Für alle Elemente einer KEGG Stoffwechselweg-Karte können korrespondierende Elemente in der SBGN *Process Description*-Sprache gefunden werden. Um eine gültige PD-Karte zu erstellen, erhöht sich jedoch während der Übersetzung die Anzahl der Knoten und Kanten. Sei $G_{SBGN} = (V_{SBGN}, E_{SBGN})$ ein Graph mit $V_{SBGN} = V_{SC} \cup V_{SU} \cup V_{TE} \cup V_{MA} \cup V_{PR}$ und $E_{SBGN} = E_{EQ} \cup E_{CA} \cup E_{CO} \cup E_{PR}$, der die SBGN PD-Karte repräsentiert. Die Über-

²Für eine Übersicht siehe http://www.kegg.jp/kegg/document/help_pathway.html.

³<http://www.kegg.jp/kegg/pathway/map/map00030.html>

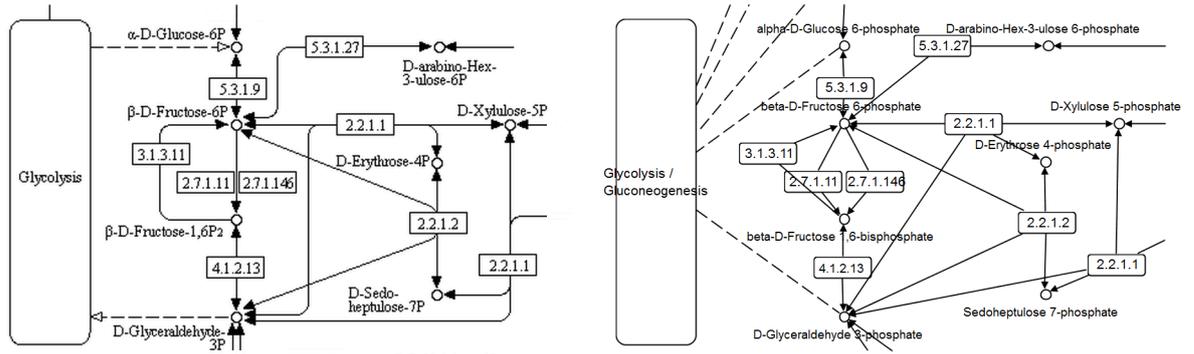


Abbildung 4.1: Ausschnitt des Pentose-Phosphat-Stoffwechselweges aus der KEGG-Datenbank. Links: KEGG-Referenzbild, rechts: KGML-Repräsentation (siehe auch Abbildung 4.2).

setzung eines *other molecule*-Knotens, eines *another map*-Knotens und einer *link to another map*-Kante von KEGG zu SBGN PD erfolgt direkt. Ein *other molecule*-Knoten v_{OM} wird als *Simple Chemical*-Knoten $v_{SC} \in V_{SC}$ abgebildet, ein *another map*-Knoten v_{AM} wird als *Submap*-Knoten $v_{SU} \in V_{SU}$ einschließlich der erforderlichen *Terminal*-Knoten $v_{TE} \in V_{TE}$ abgebildet und eine *link to another map*-Kante e_{LM} wird als eine *Equivalence Arc*-Kante $e_{EQ} \in E_{EQ}$ abgebildet. Eine Reaktion einer KEGG Stoffwechselweg-Karte, dargestellt durch einen *gene product*-Knoten v_{GP} , wird in einen *Macromolecule*-Knoten $v_{MA} \in V_{MA}$ übersetzt und repräsentiert damit die Reaktion katalysierende Enzym. Außerdem wird ein *Process*-Knoten $v_{PR} \in V_{PR}$ hinzugefügt, der die eigentliche Reaktion repräsentiert. Beide Knoten werden durch eine zusätzliche *Catalysis*-Kante $e_{CA} \in E_{CA}$ verbunden. Die Übersetzung eines Reaktionsknotens erhöht daher die Anzahl von Knoten und Kanten. *Molecular interaction or relation*-Kanten e_{MI} werden entsprechend der Reaktion, mit der sie verbunden sind, übersetzt. Im Fall einer irreversiblen Reaktion wird die Kante $e_{MI} \in V_{OM} \times V_{GP}$ vom Eduktknoten zum Reaktionsknoten als *Consumption*-Kante $e_{CO} \in E_{CO}$ übersetzt und die Kante $e_{MI} \in V_{GP} \times V_{OM}$ vom Reaktionsknoten zum Produktknoten als *Production*-Kante $e_{PR} \in E_{PR}$. Für reversible Reaktionen vereinfacht sich die Übersetzung, alle Kanten e_{MI} werden als *Production*-Kanten $e_{PR} \in E_{PR}$ übersetzt. Die entsprechenden Abbildungsvorschriften sind nachfolgend zusammengefasst und in Tabelle 4.1 werden die einzelnen Übersetzungen gezeigt, vollständige Beispiele sind in Abschnitt 4.3 und in Anhang A zu finden.

$$G_{KEGG} = (V_{KEGG}, E_{KEGG})$$

$$V_{KEGG} = V_{GP} \cup V_{OM} \cup V_{AM}$$

$$E_{KEGG} = E_{MI} \cup E_{LM}$$

$$G_{SBGN} = (V_{SBGN}, E_{SBGN})$$

$$V_{SBGN} = V_{SC} \cup V_{SU} \cup V_{TE} \cup V_{MA} \cup V_{PR}$$

$$E_{SBGN} = E_{EQ} \cup E_{CA} \cup E_{CO} \cup E_{PR}$$

$$\begin{aligned}
m_{GP} : V_{GP} &\rightarrow (V_{MA} \times V_{PR}) \times E_{CA} \\
m_{OM} : V_{OM} &\rightarrow V_{SC} \\
m_{AM} : V_{AM} &\rightarrow V_{SU} \times (\mathcal{P}(V_{TE}) \setminus \emptyset) \\
m_{MI_1} : V_{OM} \times V_{GP} &\rightarrow E_{CO} \\
m_{MI_2} : V_{GP} \times V_{OM} &\rightarrow E_{PR} \\
m_{LM} : E_{LM} &\rightarrow E_{EQ}
\end{aligned}$$

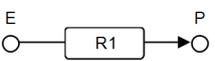
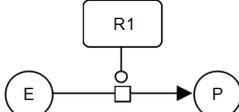
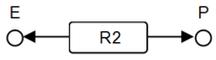
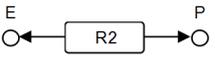
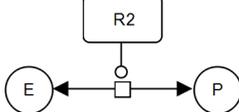
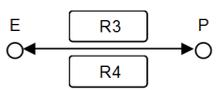
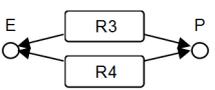
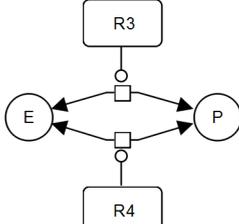
	KEGG	KGML	SBGN PD
irreversible Reaktion			
reversible Reaktion			
mehrere Reaktionen zwischen Reaktionspartnern			
Link zu einer anderen Karte			

Tabelle 4.1: Die Tabelle zeigt die KEGG, KGML und SBGN PD Darstellung für eine irreversible Reaktion, eine reversible Reaktion, mehrere Reaktionen zwischen Reaktionspartnern und einen Link zu einer anderen Karte und deckt damit alle Möglichkeiten der Übersetzung von KEGG Stoffwechselweg-Karten in SBGN PD-Karten unter Verwendung von KGML-Dateien ab.

Während der Übersetzung können Probleme durch Mängel in den KGML-Dateien entstehen. Die Dateien können für einige Stoffwechselweg-Karten Fehler enthalten und/oder nicht alle Informationen bereitstellen, die für eine korrekte automatische Übersetzung notwendig sind. Typische Fehler sind fehlende Informationen über die Reversibilität einer Reaktion oder den Typ eines Reaktionspartners (Edukt oder Produkt) oder generell fehlende Informationen über die Edukte und Produkte einer Reaktion.

4.2 Layout-Prozess

Die in der KEGG-Datenbank enthaltenen Stoffwechselweg-Karten werden manuell gezeichnet [5]. Das Layout wird vom Autor deshalb so gestaltet, dass die biologischen Funktionen hervorgehoben werden. Für jeden Stoffwechselweg der Datenbank ist online ein Referenzbild verfügbar⁴. Die Bilder zeigen die Stoffwechselwege in einer vereinfachten Form, die Anzahl der Kanten wurde reduziert. Grundsätzlich werden für eine Reaktion, die zwischen zwei Reaktionspartnern stattfindet, eine Kante vom Eduktknoten zum Reaktionsknoten und eine Kante vom Reaktionsknoten zum Produktknoten gezeichnet. Wenn mehrere Reaktionen zwischen zwei Reaktionspartnern stattfinden, wird hingegen nur eine Kante zwischen den beiden Reaktionspartnern gezeichnet und die Reaktionsknoten werden oberhalb und unterhalb dieser Kante angeordnet. Die KGML-Dateien enthalten hingegen genügend Informationen, um die Knoten und die Kanten vollständig rekonstruieren zu können. Es sind allerdings nicht die kompletten Layout-Informationen vorhanden. Die Dateien enthalten nur Informationen zu Knotenpositionen und Knotengrößen, die Pfade der Kanten sind nicht enthalten. Eine KEGG Stoffwechselweg-Karte kann deshalb aus einer KGML-Datei unter Erhalt des Layouts mit Ausnahme der Kantenpfade rekonstruiert werden (siehe Abbildung 4.1 (rechts)).

Wie in Abschnitt 4.1 beschrieben, wird die Anzahl der Knoten und Kanten während der Übersetzung erhöht. Dadurch wird das originale Layout durch Knotenüberlappungen beeinflusst (siehe Abbildung 4.2 (links)). Eine mögliche Lösung zur Beseitigung der Knotenüberlappungen wäre es, die Abstände zwischen den Knoten sowohl in x - als auch in y -Richtung zu vergrößern. Diese Methode hat aber den Nachteil, dass die Karten unnötig groß werden können. Außerdem bleibt das Problem der fehlenden Kantenpfade bestehen. Es wird deshalb ein Layout-Algorithmus auf die PD-Karten angewendet, der (1) das vom Autor der Karte eingeführte Layout erhält, (2) orthogonale Pfade für die Kanten berechnet und (3) das Layout so anpasst, dass es den SBGN Layout-Richtlinien folgt.

Die SBGN-Spezifikation für PD-Karten [116] enthält Anforderungen und Empfehlungen für das Layout einer Karte. Wichtige Anforderungen für die Übersetzung von KEGG Stoffwechselweg-Karten zu PD-Karten sind nachfolgend aufgelistet:

1. Knoten dürfen sich nicht überlappen außer ein Knoten enthält einen anderen Knoten;
2. wenn eine Kante einen Knoten kreuzt, muss sie auf dem Knoten gezeichnet werden, es wird empfohlen, dass Kanten Knoten nicht kreuzen;
3. Kanten dürfen die Randlinien von Knoten nicht überlappen;
4. Kanten dürfen sich nicht gegenseitig überlappen (nur kreuzen ist erlaubt);
5. *Consumption*-Kanten und *Production*-Kanten sind mittig auf gegenüberliegenden Seiten mit einem *Process*-Knoten verbunden;

⁴<http://www.kegg.jp/kegg/pathway.html>

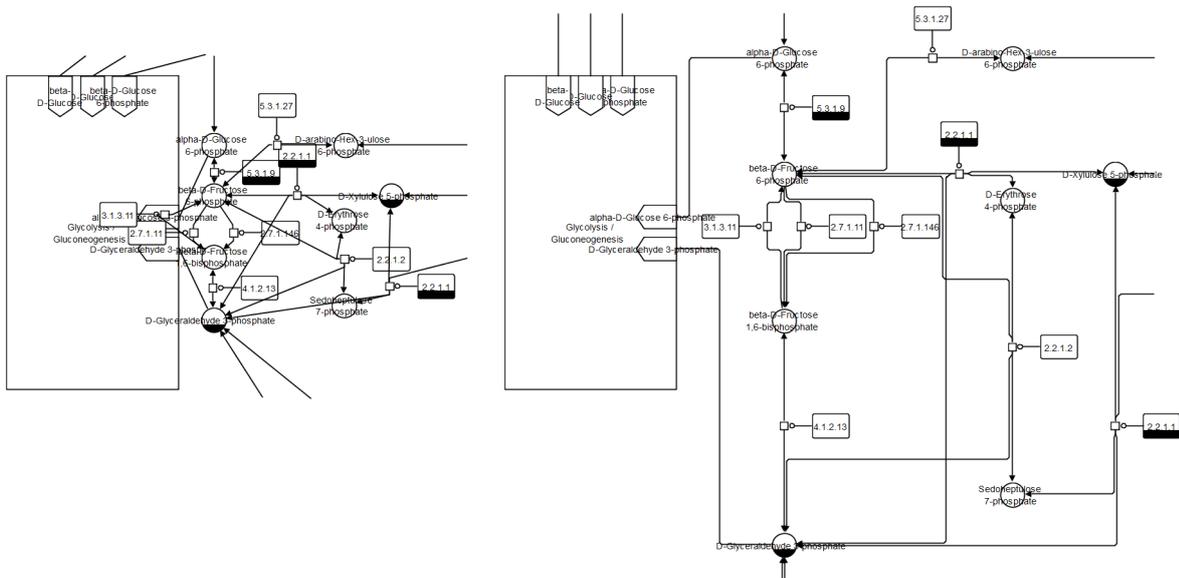


Abbildung 4.2: Ausschnitt des Pentose-Phosphat-Stoffwechselweges aus der KEGG-Datenbank. Links: SBGN-Repräsentation nach der Übersetzung ohne Anpassung des Layouts, rechts: SBGN-Repräsentation nach der Übersetzung mit Anpassung des Layouts (siehe auch Abbildung 4.1).

6. *Catalysis*-Kanten sind an den beiden anderen Seiten mit dem *Process*-Knoten verbunden;
7. mindestens ein Teil des Namens eines Knotens muss sich innerhalb des Knotens befinden;
8. Namen sollten horizontal ausgerichtet sein;
9. die Anzahl Kantenkreuzungen sollte minimiert werden.

Der Layout-Prozess wird in zwei Schritten durchgeführt. Im ersten Schritt wird für jeden Knoten der SBGN-Karte eine Position berechnet. Als Basis dienen die Positionen der Knoten der KEGG-Karte und eine Menge an Bedingungen. Um Knotenpositionen zu bestimmen, die die Bedingungen erfüllen und dabei Knotenüberlappungen vermeiden (Anforderung 1), wird ein auf Bedingungen basierendes Layout-Verfahren genutzt. Im zweiten Schritt werden die berechneten Knotenpositionen sowie Informationen über Kantenverbindungen genutzt, um orthogonale Pfade für die Kanten zu berechnen, die die Anforderungen 2–6 erfüllen. In den nächsten beiden Abschnitten werden diese Schritte detaillierter beschrieben.

Die Anforderungen 7 und 8 werden beim Darstellen der Knotennamen berücksichtigt. Da Knotennamen oft sehr lang sind, lässt es sich nicht immer vermeiden, dass der Text andere Objekte überlappt. Die letzte Anforderung der Minimierung von Kantenkreuzungen ist ein schwer lösbares Problem [56].

4.2.1 Bestimmung von Knotenpositionen

Im ersten Schritt des Layout-Prozesses sollen die Knotenpositionen unter Verwendung eines auf Bedingungen basierenden Layout-Algorithmus gefunden werden. Dabei werden die Größe und die Position jedes Knotens sowie eine Menge an Bedingungen, die erfüllt werden sollen, genutzt. Die Bedingungen sind (1) Ausrichtung und Abstand von Knoten, um das Originallayout des Autors zu erhalten, (2) Nicht-Überlappung von Knoten, (3) Einkapselung und Ausrichtung für hierarchische Knoten und (4) feste relative Positionen für *Macromolecule*-Knoten in Bezug auf die *Process*-Knoten für Reaktionen. Zusätzlich wird die *Meta*-Bedingung Reaktionsgruppe verwendet, die Ausrichtungs- und Abstands-Bedingungen für die Reaktionen einer Reaktionsgruppe zusammenfasst (siehe unten). Diese Bedingungen sind in Abbildung 4.3 für das Beispiel aus Abbildung 4.2 (rechts) dargestellt.

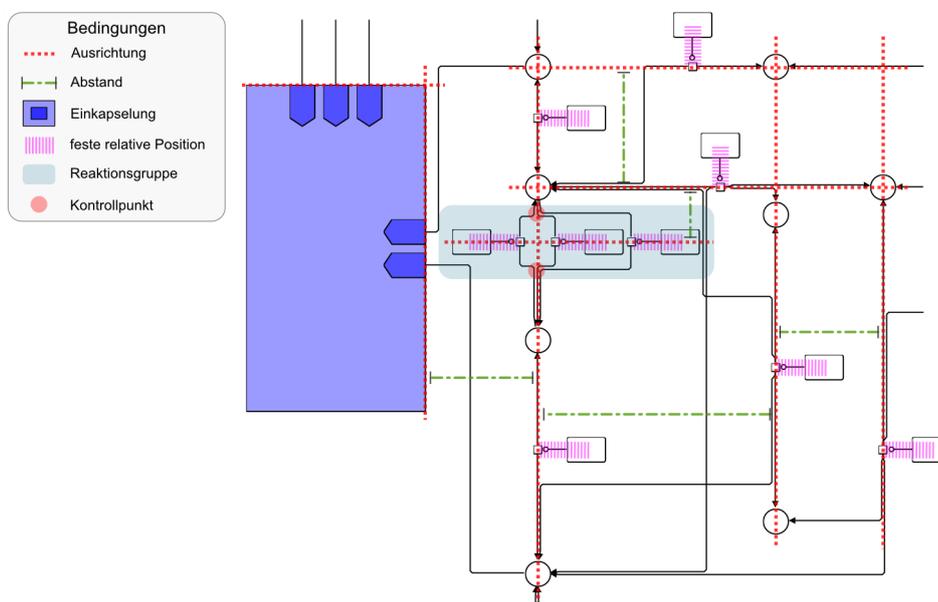


Abbildung 4.3: Ausschnitt des Pentose-Phosphat-Stoffwechselweges aus der KEGG-Datenbank entsprechend Abbildung 4.2 (rechts). Es werden die Bedingungen Ausrichtung, Abstand, Einkapselung, feste relative Position und Reaktionsgruppe sowie Kontrollpunkte (siehe Abschnitt 4.2.2) gezeigt. Knotennamen wurden für eine bessere Darstellung entfernt.

Zunächst werden Gruppen von im originalen KEGG-Layout sichtbar horizontal (x -Richtung) oder vertikal (y -Richtung) ausgerichteten Knoten in der Karte gesucht. Bei dieser Suche wird eine geringe Toleranz $\epsilon = 2,5$ verwendet, um Fälle abzudecken, bei denen eine Ausrichtung sichtbar ist, diese sich aber nicht direkt aus den x - und y -Koordinaten der Knoten ergibt. Die Ausrichtungen werden für alle Edukte und Produkte (*Simple Chemical*-Knoten in SBGN) sowie Reaktionen (*Process*-Knoten in SBGN), die nicht zu einer Reaktionsgruppe gehören (siehe unten), unabhängig davon ermittelt, ob sie über eine Kante miteinander verbunden sind oder nicht. Sobald sämtliche Ausrichtungen bekannt sind, werden Abstände zwischen den Ausrichtungen definiert, um die relative Reihenfolge und die Abstände der Aus-

richtungen im Layout zu erhalten. Für zwei benachbarte vertikale Ausrichtungen ac_u mit den Knoten u_k und ac_v mit den Knoten v_l mit

$$ac_{u_x} = \frac{1}{k} \cdot \sum_{m=1}^k u_{m_x} \quad \text{und} \quad |u_{i_x} - u_{j_x}| \leq \epsilon \quad \text{für} \quad i, j = 1, \dots, k$$

und

$$ac_{v_x} = \frac{1}{l} \cdot \sum_{n=1}^l v_{n_x} \quad \text{und} \quad |v_{i_x} - v_{j_x}| \leq \epsilon \quad \text{für} \quad i, j = 1, \dots, l$$

ist der minimale Abstand $sc_{uv} = ac_{v_x} - ac_{u_x}$. Für horizontale Ausrichtungen ergeben sich Abstände zwischen benachbarten horizontalen Ausrichtungen entsprechend.

Finden mehrere Reaktionen zwischen einem Edukt und einem Produkt statt, werden eine horizontale und eine vertikale Ausrichtung für die Reaktionen definiert, um sie in horizontalen oder vertikalen Reaktionsgruppen anzuordnen (siehe Abbildung 4.3, hier wird eine horizontale Reaktionsgruppe gezeigt).

Sind Edukt und Produkt links und rechts von den Reaktionen positioniert, wird eine vertikale Reaktionsgruppe für alle Reaktionen v_k definiert mit einer vertikalen Ausrichtung ac_v mit

$$ac_{v_x} = \frac{1}{k} \cdot \sum_{m=1}^k v_{m_x}$$

und einer horizontalen Ausrichtung ac_h mit einem Versatz off_v für die Reaktionen relativ zur horizontalen Ausrichtung

$$off_v = \begin{cases} (-2, 5i - 0, 5) \cdot h_{v_{MA}} & \text{für} \quad i = 0, \dots, \lceil \frac{k}{2} \rceil - 1 \\ (2, 5i + 0, 5) \cdot h_{v_{MA}} & \text{für} \quad i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1 \end{cases}$$

mit $h_{v_{MA}}$ Höhe der *Macromolecule*-Knoten und k Anzahl der Reaktionen in der Reaktionsgruppe. Existiert eine gemeinsame horizontale Ausrichtung für das Edukt und Produkt der Reaktionsgruppe, wird ac_h mit dieser horizontalen Ausrichtung vereinigt, sonst werden Abstände zwischen ac_h und benachbarten horizontalen Ausrichtungen definiert.

Sind Edukt und Produkt oberhalb und unterhalb von den Reaktionen positioniert, wird eine horizontale Reaktionsgruppe für alle Reaktionen v_k definiert mit einer horizontalen Ausrichtung ac_h mit

$$ac_{h_y} = \frac{1}{k} \cdot \sum_{m=1}^k v_{m_y}$$

und einer vertikalen Ausrichtung ac_v mit einem Versatz off_h für die Reaktionen relativ zur vertikalen Ausrichtung

$$off_h = \begin{cases} (-2i - 0, 5) \cdot b_{v_{MA}} & \text{für} \quad i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1 \\ (2i + 0, 5) \cdot b_{v_{MA}} & \text{für} \quad i = 0, \dots, \lceil \frac{k}{2} \rceil - 1 \end{cases}$$

mit $b_{v_{MA}}$ Breite der *Macromolecule*-Knoten und k Anzahl der Reaktionen in der Reaktionsgruppe. Existiert eine gemeinsame vertikale Ausrichtung für das Edukt und Produkt der

Reaktionsgruppe, wird ac_v mit dieser vertikalen Ausrichtung vereinigt, sonst werden Abstände zwischen ac_v und benachbarten vertikalen Ausrichtungen definiert.

Die relativen Positionen für einen *Macromolecule*-Knoten und einen *Process*-Knoten, die durch eine *Catalysis*-Kante verbunden sind, werden durch zwei Bedingungen festgelegt: eine gemeinsame Ausrichtung in x -Richtung und einen Abstand $sc = h_{v_{MA}}$ in y -Richtung oder umgekehrt mit $h_{v_{MA}}$ Höhe der *Macromolecule*-Knoten.

Gemäß der Spezifikation für SBGN *Process Description*-Karten enthalten *Submap*-Knoten *Terminal*-Knoten. Dies wird grafisch durch sich übereinander liegende Knoten dargestellt, die auf einer Seite (oben, rechts, unten, links) ausgerichtet sind. Um zu vermeiden, dass die Bedingung Nicht-Überlappung Anwendung findet (siehe unten), wird eine Einkapselung für *Submap*-Knoten und die entsprechenden *Terminal*-Knoten definiert. Darüber hinaus wird für jeden *Terminal*-Knoten zusätzlich noch eine Ausrichtung mit Versatz relativ zum *Submap*-Knoten definiert, um die Position innerhalb des *Submap*-Knotens festzulegen. Für einen *Submap*-Knoten v_{SU} mit *Terminal*-Knoten sei ac_v eine vertikale Ausrichtung mit $ac_{v_x} = v_{SU_x}$ und ac_h eine horizontale Ausrichtung mit $ac_{h_y} = v_{SU_y}$. Für einen *Terminal*-Knoten v_{TE} mit einer Ausrichtung oben am *Submap*-Knoten ist der Versatz

$$off_o = ac_{h_y} - \frac{h_{v_{SU}}}{2} + \frac{h_{v_{TE}}}{2}$$

mit $h_{v_{SU}}$ Höhe des *Submap*-Knotens und $h_{v_{TE}}$ Höhe des *Terminal*-Knotens. Für einen *Terminal*-Knoten v_{TE} mit einer Ausrichtung rechts am *Submap*-Knoten ist der Versatz

$$off_r = ac_{v_x} + \frac{b_{v_{SU}}}{2} - \frac{b_{v_{TE}}}{2}$$

mit $b_{v_{SU}}$ Breite des *Submap*-Knotens und $b_{v_{TE}}$ Breite des *Terminal*-Knotens. Der Versatz für eine Ausrichtung eines *Terminal*-Knotens unten und links am *Submap*-Knoten ergibt sich entsprechend.

Da auf die Bestimmung der Knotenpositionen im zweiten Schritt des Layouts noch die Bestimmung der Kantenpfade folgt, werden die Dimensionen der Knoten für den Algorithmus überhöht definiert, um in der Karte Platz für die Kanten zu erhalten. Die Breite und Höhe aller Knoten außer der *Terminal*-Knoten wird um $h_{v_{SC}}$ vergrößert mit $h_{v_{SC}}$ Höhe der *Simple Chemical*-Knoten.

Knoten sollen zum einen aus ästhetischen Gründen und zum zweiten entsprechend Layout-Anforderung 1 (siehe auch Seite 43) nicht überlappen (Bedingung Einkapselung ausgeschlossen), deshalb wird eine entsprechende Bedingung zur Vermeidung von Überlappungen festgelegt.

Für die Lösung des Layout-Problems wird die Methode aus [43] verwendet. Dabei wird versucht, die Knoten unter Beachtung der definierten Bedingungen so gut wie möglich auf die gewünschte Position zu setzen. Während dieses Prozesses werden zunächst die hier beschriebenen höherrangigen Bedingungen (Ausrichtung, Nicht-Überlappung, Einkapselung, feste relative Position) allesamt auf niederrangige Abstands-Bedingungen abgebildet [43]. Nicht-

Überlappung wird von der Methode automatisch umgesetzt, wenn die entsprechende Bedingung festgelegt wurde.

Prinzipiell ist es möglich, widersprüchliche Bedingungen zu definieren, die nicht erfüllbar sind (z. B. eine horizontale und eine vertikale Ausrichtung und einen Abstand für zwei Knoten). Eine derartige Definition von Bedingungen wird hier ausgeschlossen.

4.2.2 Bestimmung der Kantenpfade

Es gibt auch verschiedene Anforderungen an die Bestimmung der Kantenpfade, um ein Layout der Karten zu erzielen, dass den SBGN Layout-Richtlinien folgt.

Die Kanten sollen entsprechend den KEGG-Referenzbildern orthogonalen Pfaden aus vertikalen und horizontalen Liniensegmenten folgen. Die Gesamtlänge eines Pfades und die Anzahl der Segmente soll dabei minimal sein. Wenn Kanten einem gemeinsamen Pfad folgen, und sich damit überlappen würden, sollen sie parallel laufend gezeichnet werden, so dass sie visuell unterschieden werden können. Diese Festlegung entspricht der Layout-Anforderung 4 (siehe auch Seite 43). Es wird außerdem festgelegt, dass die Kanten die Knoten nicht kreuzen oder berühren dürfen, es sei denn sie sind mit einem Knoten verbunden.

Die Bestimmung der Kantenpfade erfolgt unter Anwendung einer Technik, die einen Sichtbarkeits-Graphen basierend auf einer Menge von rechteckigen Objekten bildet. Dieser Graph wird dann genutzt, um orthogonale Pfade zu berechnen, die die Objekte umgehen [152]. Dazu müssen alle Knoten aus der Karte als Rechtecke definiert werden. Für die Kanten wird definiert (1) mit welchen Knoten sie verbunden sind, (2) ob sie mit einem Knoten aus einer bestimmten Richtung kommend verbunden sind und (3) ob sie mit einem Knoten an einer bestimmten Position verbunden sind.

Für die meisten Knoten wird definiert, dass sie mit Kanten aus einer beliebigen Richtung kommend verbunden werden. Als Position wird dabei der Mittelpunkt des Knotens genutzt (auch wenn die Kante nur bis zum Rand des Knotens gezeichnet wird). Für *Process*-Knoten werden Anschlusspunkte definiert, die eine bestimmte Position und die Richtung für den Anschluss von *Consumption*- und *Production*-Kanten vorgeben. Die Rechtecke für *Process*- und *Macromolecule*-Knoten werden so definiert, dass sie sich berühren und keine andere Kante eine *Catalysis*-Kante kreuzen kann (siehe Abbildung 4.4). Anschlusspunkte werden ebenfalls für *Terminal*-Knoten festgelegt, so dass Kanten an der Seite mit ihnen verbunden werden, an der sie mit einem *Submap*-Knoten ausgerichtet sind.

Für Reaktionsgruppen werden die Kanten jeweils von den beiden Reaktionspartnern gemeinsam bis zu Kontrollpunkten innerhalb der Reaktionsgruppe geführt, an denen sie dann abzweigen (siehe Abbildung 4.3). Kontrollpunkte können als zusätzliche Bedingung für Kanten festgelegt werden, sie definieren Positionen in einer Karte, die von den Kantenpfaden passiert werden müssen.

Nachdem die Bestimmung der Kantenpfade durchgeführt wurde, werden sich überlappende Kanten voneinander separiert. Dazu wird die *Nudging*-Funktion aus [152] genutzt.

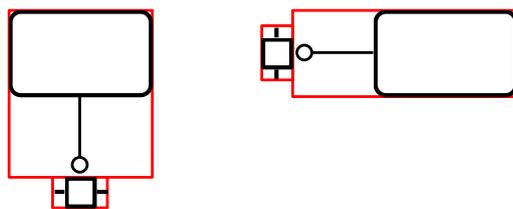


Abbildung 4.4: Die Rechtecke (in Rot) für *Process*- und *Macromolecule*-Knoten werden so definiert, dass sie sich berühren und keine andere Kante eine *Catalysis*-Kante kreuzen kann. Knotennamen wurden für eine bessere Darstellung entfernt.

4.3 Beispiele

Die vorgestellte Methodik kann auf allen metabolischen Stoffwechselweg-Karten der KEGG-Datenbank außer den globalen Übersichtskarten angewendet werden⁵. Hier soll nun das komplette Beispiel des Pentose-Phosphat-Stoffwechselweges aus der KEGG-Datenbank⁶, das für die Erläuterung der Methodik verwendet wurde, dargestellt werden. In den Abbildungen 4.5 und 4.6 sind das Referenzbild aus der KEGG-Datenbank und die entsprechende PD-Karte nach der Übersetzung und Anwendung des Layout-Algorithmus zu sehen. In Anhang A sind weitere Beispiele zu finden.

Das hier vorgestellte Beispiel sowie die Beispiele in Anhang A zeigen sehr gut, dass die vorgestellte Methodik SBGN PD-Karten generiert, deren Layout im Wesentlichen dem Layout in den Referenzbildern entspricht. Besonders auffällig sind dabei horizontale und vertikale Pfade sowie Zyklen, die sehr gut erhalten bleiben. Durch die erhöhte Anzahl an Knoten und Kanten sowie durch die Reaktionsgruppen wird allerdings insgesamt etwas mehr Platz zur Darstellung der Karten benötigt.

Es fällt allerdings auch auf, dass Knotennamen schwer zu lesen sind, da sie auf Grund ihrer Länge oft mit anderen Elementen der Karte überlappen. Momentan erfordert die SBGN-Spezifikation für PD, dass *Simple Chemical*-Knoten als Kreis dargestellt werden und dass der Knotenname zentriert auf dem Knoten gezeichnet wird. Als Folge dieser Definition könnte die Darstellung derzeit nur durch eine Vergrößerung der *Simple Chemical*-Knoten oder durch Abkürzung der Knotennamen verbessert werden. Eine Abkürzung der Knotennamen könnte aber das Verständnis der Karte erschweren und eine Vergrößerung der *Simple Chemical*-Knoten würde die Darstellung der Karte deutlich beeinflussen.

Das Problem der Knotennamen bei *Simple Chemical*-Knoten wurde im SBGN-Projekt diskutiert und wird möglicherweise mit der nächsten Version der Spezifikation für die *Process Description*-Sprache gelöst. Die vorgeschlagene Lösung würde es erlauben, die *Simple Chemical*-Knoten in horizontaler Richtung zu strecken, so dass mehr Platz im Knoten für den Knotennamen zur Verfügung stände.

⁵Die KGML-Dateien für die globalen Übersichtskarten enthalten nicht alle Informationen, die für eine vollständige Rekonstruktion der Darstellung notwendig sind.

⁶<http://www.kegg.jp/kegg/pathway/map/map00030.html>

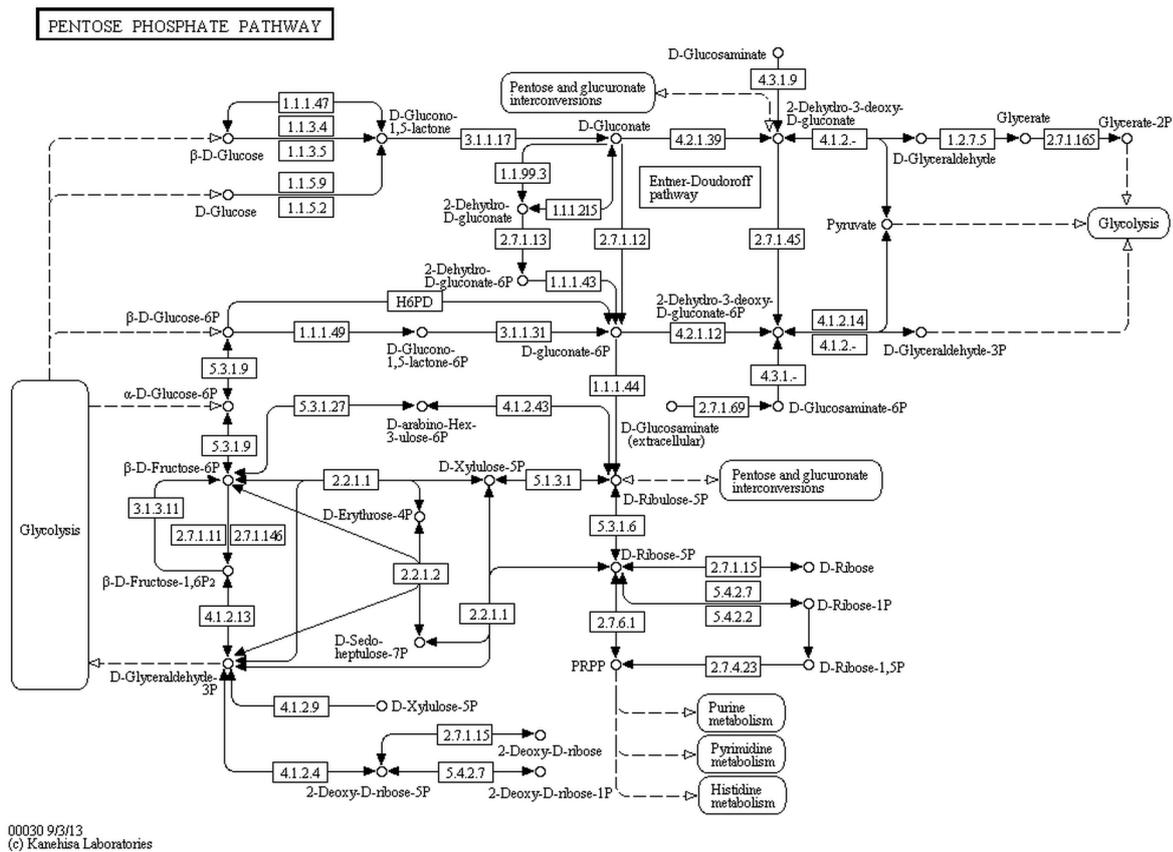


Abbildung 4.5: Referenzbild des Pentose-Phosphat-Stoffwechselweges aus der KEGG-Datenbank.

4.4 Diskussion und Zusammenfassung

In der biologischen Forschung werden oft Datenbanken genutzt, die biologische Netzwerke und entsprechende Darstellungen zur Verfügung stellen. Die grafische Darstellung der Netzwerke folgt oft einigen Konventionen, idealerweise sollten sie aber in einer standardisierten Darstellung angeboten werden.

Hier wurde eine Methodik vorgestellt, die metabolische Stoffwechselweg-Karten aus der vielfach genutzten KEGG-Datenbank in eine Darstellung in der SBGN *Process Description*-Sprache übersetzt. Dabei wird zusätzlich ein Layout-Algorithmus angewendet, der das originale Layout erhält und gleichzeitig den SBGN Layout-Richtlinien folgt. Der Erhalt des Layouts ist besonders wichtig, da die Stoffwechselweg-Karten der KEGG-Datenbank von Hand gezeichnet werden. Das originale Layout basiert also auf Expertenwissen und bestimmte Aspekte einer Karte werden hervorgehoben.

Die beschriebene Methode wurde implementiert (siehe auch Abschnitt 8.2) und kann interaktiv verwendet werden [32]. Eine Stoffwechselweg-Karte aus der KEGG-Datenbank in eine PD-Karte zu übersetzen und den Layout-Algorithmus anzuwenden, dauert für Karten mit einer Größe bis in der Summe 500 Knoten und Kanten (ca. 90% aller Karten) bis zu

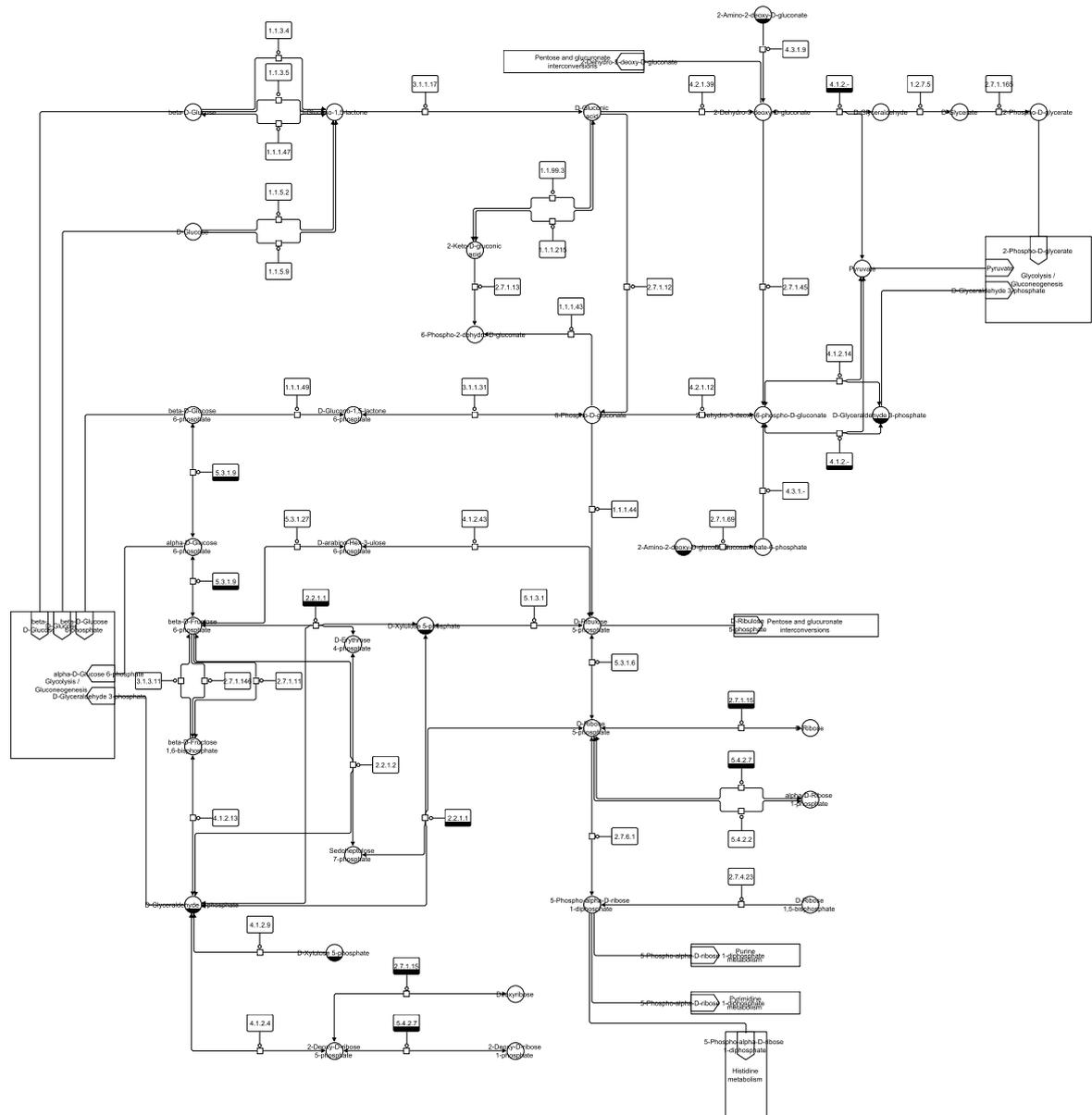


Abbildung 4.6: Der Pentose-Phosphat-Stoffwechselweg aus der KEGG-Datenbank wurde basierend auf der KGML-Datei in eine SBGN PD-Karte übersetzt, anschließend wurde der beschriebene Layout-Algorithmus angewendet.

2,5 Sekunden⁷. Bei größeren Karten aus der KEGG-Datenbank kann der Prozess bis zu zehn Sekunden dauern (siehe Abbildung 4.7).

Im Vergleich mit den Software-Anwendungen KEGGtranslator [151] und MGV [137], die ebenfalls KGML-Dateien laden und in SBGN darstellen können, ist der Erhalt des Layouts basierend auf den Informationen der KEGG-Karte als größter Unterschied zu nennen. Zusätzlich können natürlich auch die weiteren in dieser Arbeit vorgestellten Methoden (siehe auch Kapitel 5 und 7) auf den übersetzten Karten angewendet werden.

⁷Gemessen auf einem Intel® Core™ i7-920 2,67 GHz.

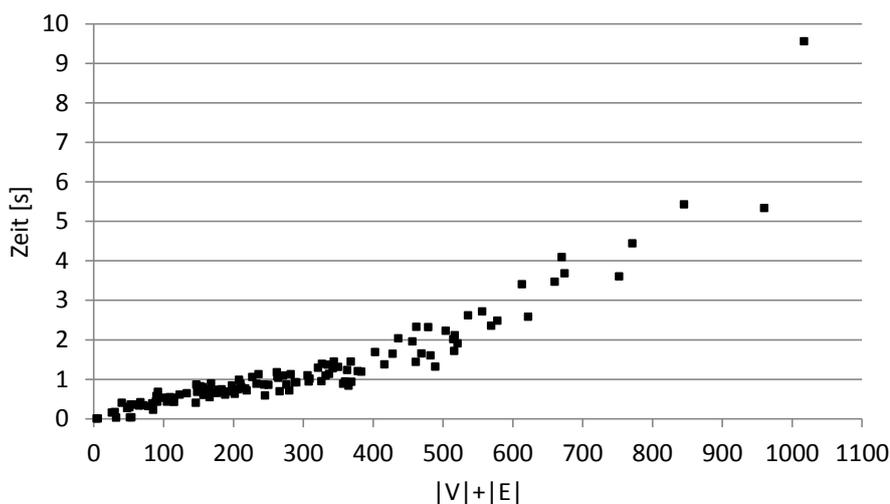


Abbildung 4.7: Laufzeiten der Methode für verschiedene Größen der Stoffwechselweg-Karten aus der KEGG-Datenbank ($|V|$ – Anzahl Knoten, $|E|$ – Anzahl Kanten).

Wie bereits in Abschnitt 4.3 beschrieben, gibt es im Moment Probleme mit der Darstellung von Knotennamen auf *Simple Chemical*-Knoten. Dieses Problem kann gegebenenfalls mit einer neuen Version der Spezifikation für PD-Karten gelöst werden. Die Darstellung einer übersetzten PD-Karte kann außerdem hinsichtlich Platzbedarf verbessert werden. Der Layout-Algorithmus kann möglicherweise durch Definition weiterer Abstands-Bedingungen optimiert werden, so dass die Karten kompakter dargestellt werden können. Ein weiteres bisher nicht betrachtetes Problem ist die Berechnung von Kantenpfaden, wenn zwischen Knoten nicht genügend Platz zur Darstellung der Kanten vorhanden ist. Dadurch kann es in Einzelfällen passieren, dass Kanten ohne sichtbaren Abstand parallel gezeichnet werden und visuell nicht mehr voneinander unterschieden werden können. Diese Fälle müssen erkannt werden und die Bedingungen für das Layout der entsprechenden Knoten so definiert werden, dass genügend Platz zum Zeichnen der Kanten vorhanden ist.

Weitere Anwendungsfälle für den Layout-Algorithmus könnten SBGN-Karten sein, die aus dem SBML- oder dem BioPAX-Format übersetzt wurden. Da diese beiden Formate prinzipiell keine Layout-Informationen enthalten (ausgenommen SBML-Format mit Layout-Erweiterung [57]), müssten initiale Positionen für die Knoten entweder aus dem Kontext der Karte ermittelt werden (z. B. relative Positionen für *Macromolecule*-Knoten und *Process*-Knoten, die durch eine Kante verbunden sind) oder durch Anwendung anderer Algorithmen für das Layout von Graphen berechnet werden. Ein Beispiel für das SBML-Format ist in Abschnitt 9.3 zu finden.

Kapitel 5

Umwandlung von SBGN *Process Description*-Karten in SBGN *Activity Flow*-Karten

SBGN-Karten in der *Process Description*-Sprache sind oft groß und komplex und bieten eine detaillierte Sicht auf ein biologisches System. Eine Umwandlung in die *Activity Flow*-Sprache ermöglicht zum einen eine vereinfachte Sicht auf das dargestellte System und stellt zusätzlich weitere Sichten auf das biologische System bereit. Eine mechanistische Darstellung eines Systems in der *Process Description*-Sprache kann in der *Activity Flow*-Sprache z. B. in einer Weise dargestellt werden, die in ähnlicher Form auch in der Literatur zu finden ist.

Da es verschiedene Möglichkeiten der Abstraktion von der *Process Description*- in die *Activity Flow*-Sprache gibt, ist ein statischer Ansatz für eine Umwandlung nicht ausreichend. Für unterschiedliche Anwendungsfälle können und sollen verschiedene Informationen auf einer Karte hervorgehoben werden. Auf Basis der beiden Sprachen lassen sich jedoch Regeln für die Umwandlung definieren, die dann je nach Anwendungsfall flexibel kombiniert und für eine Umwandlung genutzt werden können. Durch Anwendung der Regeln hat sich empirisch herausgestellt, dass Umwandlungsregeln zu Templates zusammengefasst werden können, die auf bestimmte Aspekte der dargestellten biologischen Prozesse fokussieren. Während der Entwicklung der Methodik konnten so vier Templates beschrieben werden.

Eine flexible Umwandlung von *Process Description*-Karten in *Activity Flow*-Karten auf Basis von Regeln oder vereinfacht auf Basis von Templates bietet gegenüber einer statischen Umwandlung den Vorteil, dass die verschiedenen Möglichkeiten der Abstraktion genutzt werden können. Im folgenden Abschnitt werden die Regeln für die Umwandlung erläutert, danach werden die Templates dargestellt und im Anschluss werden einige Beispiele auf Basis der Regeln und Templates gezeigt.

5.1 Regeln zur Umwandlung

Es gibt prinzipiell viele Möglichkeiten eine PD-Karte in eine AF-Karte umzuwandeln. Deshalb wurde ein Konzept für eine Umwandlung entwickelt, das auf einer Menge von Regeln basiert.

Die beiden Sprachen definieren mehrere gemeinsame Elemente. Dies sind die Knoten *Compartment*, *Phenotype*, *Perturbing Agent* (*Perturbation*), *Submap* (mit *Terminal*-Knoten), *Tag*, *AND*, *OR*, *NOT* sowie die Kanten *Logic Arc* und *Equivalence Arc*. Sie können direkt umgewandelt werden, da sie in beiden Sprachen vorhanden sind und dieselbe Bedeutung haben (Regeln c_1 bis c_8 , siehe Tabelle 5.1). Einen Spezialfall bilden dabei die *Logical Operator*-Knoten *AND*, *OR* und *NOT*. In PD ist das Ziel der ausgehenden Kante eines *Logical Operator*-Knotens immer ein *Process Node*-Knoten oder ein anderer *Logical Operator*-Knoten. Bei einer einfachen Umwandlung könnte ein *Logical Operator*-Knoten mit mehreren ausgehenden Kanten entstehen. Das ist jedoch laut der SBGN-Spezifikationen nicht erlaubt und somit müssen während der Umwandlung mehrere *Logical Operator*-Knoten mit den gleichen eingehenden Kanten und jeweils einer ausgehenden Kante erzeugt werden (siehe Abbildung 5.1).

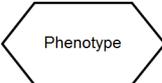
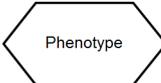
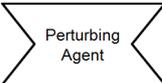
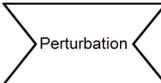
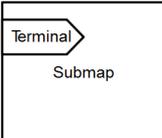
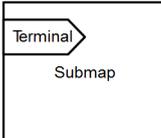
Regel	PD	AF
c_1		
c_2		
c_3		
c_4		
c_5		
c_6	  	  
c_7		
c_8		

Tabelle 5.1: Regeln c_1 bis c_8 für die Umwandlung gemeinsamer Elemente der *Process Description*- und *Activity Flow*-Sprache.

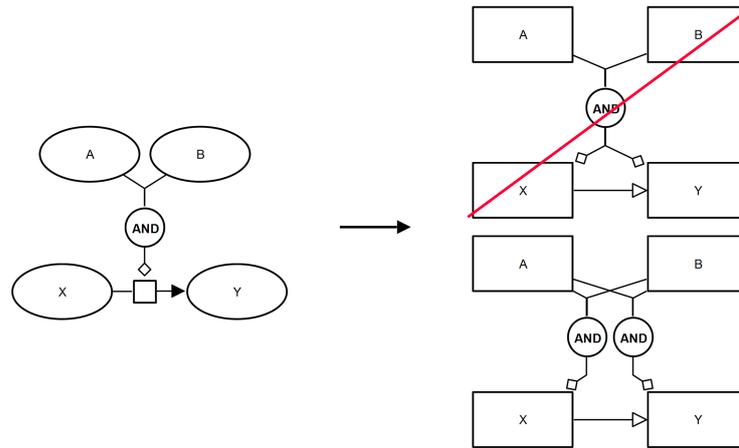


Abbildung 5.1: Beispiel für die Umwandlung einer PD-Karte mit einem *Logical Operator*-Knoten *AND*. Links: Darstellung der PD-Karte, rechts oben: ungültige Umwandlung (zwei ausgehenden Kanten sind in SBGN für einen *Logical Operator*-Knoten nicht erlaubt), rechts unten: gültige Umwandlung.

5.1.1 Umwandlung von Knoten

In Tabelle 5.1 wurde bereits dargestellt, dass zahlreiche Knoten sowohl in der *Process Description*- als auch in der *Activity Flow*-Sprache definiert sind und deshalb direkt umgewandelt werden können. Die *Process Description*-Sprache definiert weiterhin die *Entity Pool Node*- und die *Process*-Knoten, die keine entsprechenden Knoten in der *Activity Flow*-Sprache haben. *Process*-Knoten werden bei der Umwandlung von Kanten betrachtet (siehe Abschnitt 5.1.2), es müssen daher nur Regeln für *Entity Pool Node*-Knoten festgelegt werden (außer *Perturbing Agent*-Knoten, siehe auch Tabelle 5.1). Allgemein werden *Entity Pool Node*-Knoten in *Biological Activity*-Knoten in AF umgewandelt. Diese werden durch eine *Auxiliary Unit* erweitert, die in Form und Bedeutung dem *Entity Pool Node*-Knoten entspricht. Tabelle 5.2 stellt die Umwandlung dar. Während der Umwandlung können Informationen verloren gehen, da z. B. *Multimer*-Knoten nicht in AF dargestellt werden können (Regeln n_1 bis n_4). Außerdem werden die ursprünglichen *Auxiliary Units* der Knoten nicht umgewandelt, da AF keine Möglichkeit bietet, diese zu repräsentieren.

Im Folgenden werden notwendige Erweiterungen zum allgemeinen Konzept der Umwandlung von *Entity Pool Node*-Knoten erläutert.

In PD sind *Entity Pool Node*-Knoten mit gleichem Knotennamen und unterschiedlichen *Auxiliary Units* in einem *Compartment*-Knoten erlaubt, in AF kann ein *Biological Activity*-Knoten nur eine *Auxiliary Unit* haben, die den Typ der Aktivität beschreibt. Für die Umwandlung von Knoten mit gleichen Knotennamen und unterschiedlichen *Auxiliary Units* gibt es zwei Möglichkeiten. Entweder werden diese Knoten in einem *Biological Activity*-Knoten zusammengefasst und die originalen *Auxiliary Units* werden verworfen (Regel a_1) oder die Knoten werden unabhängig voneinander umgewandelt und die Bezeichner der *Auxiliary*

Regel	PD	AF
n_1		
n_2		
n_3		
n_4		
n_5		
n_6		es erfolgt keine Umwandlung

Tabelle 5.2: Regeln n_1 bis n_6 für die Umwandlung von *Entity Pool Node*-Knoten (außer *Perturbing Agent*-Knoten, siehe auch Tabelle 5.1).

Units werden an die Knotennamen angehängt (Regel a_2). Die beiden Möglichkeiten sind in Tabelle 5.3 dargestellt.

Regel	PD	AF
a_1		
a_2		

Tabelle 5.3: Regeln a_1 und a_2 für die Umwandlung von Knoten mit gleichem Knotennamen aber unterschiedlichen *Auxiliary Units*. Zwei *Macromolecule*-Knoten mit gleichem Knotennamen, aber unterschiedlichen *Auxiliary Units* in PD werden in AF zu einem Knoten kombiniert und die *Auxiliary Units* werden verworfen (Regel a_1) oder die zwei Knoten bleiben erhalten und die Bezeichner der *Auxiliary Units* werden an den Knotennamen angehängt (Regel a_2).

Source- und *Sink*-Knoten können nicht übersetzt werden, da es keinen entsprechenden Knoten in AF gibt. In PD repräsentieren sie nicht näher beschriebene Edukte und Produk-

te von Reaktionen. Da in AF Reaktionen nur indirekt beschrieben werden können, werden *Source*- und *Sink*-Knoten bei der Umwandlung nicht betrachtet.

Complex-Knoten in PD können einen Knotennamen und/oder Dekorationen (z. B. *Entity Pool Node*-Knoten) haben. In AF sind Dekorationen für *Biological Activity*-Knoten nicht möglich. Wenn der *Complex*-Knoten nur einen Knotennamen hat, wird dieser für den *Biological Activity*-Knoten übernommen (Regel co_1), sonst werden die Bezeichner der Dekorationen und der Knotennamen verknüpft (Regel co_2). Die Anwendung der beiden Regeln wird in Tabelle 5.4 gezeigt, sie werden implizit angewendet.

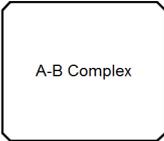
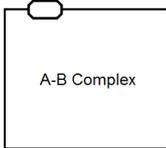
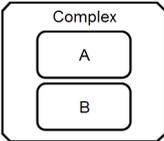
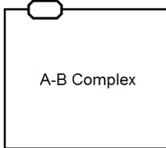
Regel	PD	AF
co_1		
co_2		

Tabelle 5.4: Regeln co_1 und co_2 für die Umwandlung von *Complex*-Knoten. Die beiden möglichen Darstellungen eines *Complex*-Knotens in PD (Knotennamen oder Knotennamen und zusätzliche Dekorationen) werden in eine Darstellung in AF überführt, Knotennamen werden übernommen oder Bezeichner von Dekorationen und Knotennamen werden verknüpft.

In PD werden *Clone Marker* verwendet, um zu zeigen, dass ein *Entity Pool Node*-Knoten mehrmals innerhalb eines *Compartment*-Knotens auftritt. Das Konzept der *Clone Marker* ist in AF nicht vorgesehen. Es gibt verschiedene Möglichkeiten der Umwandlung von Knoten mit einem *Clone Marker*. Es ist möglich, diese Knoten jeweils zu einem Knoten zusammenzufassen und alle ihre Kanten in diesem neuen Knoten starten oder enden zu lassen (Regel cl_1). Des Weiteren können Knoten mit einem *Clone Marker* jeweils fortlaufend nummeriert und ihre Nummer an den Knotennamen angehängt werden (Regel cl_2). Dadurch bleiben alle Knoten mit einem *Clone Marker* erhalten. Zuletzt können alle Knoten mit einem *Clone Marker* entfernt bzw. der AF-Karte nicht hinzugefügt werden (Regel cl_3). Diese Regel kann z. B. bei Kofaktoren angewendet werden, die mehrmals auf einer PD-Karte auftreten, aber zur Darstellung des Flusses von Aktivitäten nicht unbedingt notwendig sind. Die Umwandlung von Knoten mit einem *Clone Marker* ist in Tabelle 5.5 zusammengefasst.

5.1.2 Umwandlung von Kanten

Die *Process Node*-Knoten sind die zentralen Knoten in einer PD-Karte, da sie alle Reaktionen des biologischen Systems repräsentieren. Die Umwandlung der Kanten ist nicht so leicht möglich, da sie in verschiedener Art und Weise mit den *Process Node*-Knoten verbunden sind.

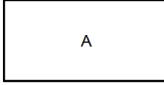
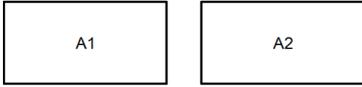
Regel	PD	AF
cl_1		
cl_2		
cl_3		es erfolgt keine Umwandlung

Tabelle 5.5: Regeln cl_1 bis cl_3 für die Umwandlung von Knoten mit einem *Clone Marker*. Knoten in PD mit einem *Clone Marker* werden zu einem Knoten zusammengefasst (Regel cl_1), werden fortlaufend nummeriert (Regel cl_2) oder werden nicht umgewandelt (Regel cl_3).

Process Node-Knoten haben für irreversible Reaktionen immer mindestens eine eingehende *Consumption*-Kante und eine ausgehende *Production*-Kante und für reversible Reaktionen mindestens eine ein- und eine ausgehende *Production*-Kante. Es gibt dadurch eine Beziehung zwischen beiden Seiten der Reaktion. Zusätzlich können eine oder mehrere regulierende Kanten mit jedem *Process Node*-Knoten verbunden sein. Die regulierenden Kanten beeinflussen die Reaktion und haben daher auch eine Beziehung zu beiden Seiten der Reaktion (regulierende Kante \leftrightarrow *Consumption*-Kante und regulierende Kante \leftrightarrow *Production*-Kante), sie beeinflussen sich aber nicht gegenseitig.

Tabelle 5.6 visualisiert, wie diese Beziehungen von der *Process Description*- in die *Activity Flow*-Sprache umgewandelt werden können (Regeln p_1 bis p_4). Es werden außerdem zwei weitere Umwandlungsregeln gezeigt. Die Regel p_5 kann auf benachbarte regulierende Elemente angewendet werden, die ein gemeinsames Edukt/Produkt beeinflussen. Regel p_6 repräsentiert, wie alle Edukte, Produkte und regulierenden Elemente einer Reaktion in AF erhalten werden können. Hat dabei der entsprechende *Process Node*-Knoten mehr als eine eingehende regulierende Kante, können alle regulierenden Kanten des gleichen Typs mit einem *Logical Operator*-Knoten *OR* verbunden werden, der dann mit dem *Logical Operator*-Knoten *AND* verbunden wird.

Für die Darstellung der Regeln p_3 und p_4 in Tabelle 5.6 wird als regulierende Kante jeweils die *Modulation*-Kante verwendet, jede andere regulierende Kante ist aber möglich. Deshalb müssen die Regeln p_3 (regulierende Kante \leftrightarrow *Production*-Kante) und p_4 (regulierende Kante \leftrightarrow *Consumption*-Kante) etwas differenzierter betrachtet werden. Für Regel p_3 gibt es in vielen Fällen eine direkte Umwandlung. Eine *Stimulation*-Kante z. B. sollte in eine *Positive Influence*-Kante umgewandelt werden. Eine *Catalysis*-Kante kann in eine *Positive Influence*-Kante umgewandelt werden, aber auch die Umwandlung in eine *Necessary Stimulation*-Kante wäre möglich. Für Regel p_4 kann jede regulierende Kante (ausgenommen die *Inhibition*-Kante) in eine *Negative Influence*- oder eine *Unknown Influence*-Kante umgewandelt werden. Empfohlene Umwandlungen für die Beziehungen regulierende Kante \leftrightarrow *Production*-Kante und regulierende Kante \leftrightarrow *Consumption*-Kante werden in Tabelle 5.7 gezeigt.

Regel	PD	AF
p_1		
p_2		
p_3		
p_4		
p_5		
p_6		

Tabelle 5.6: Regeln p_1 bis p_6 für die Umwandlung von Kanten. Von oben nach unten: Beziehung zwischen *Consumption*- und *Production*-Kante (irreversible Reaktion); Beziehung zwischen *Production*-Kanten (reversible Reaktion); Beziehung zwischen *Modulation*- und *Production*-Kante; Beziehung zwischen *Modulation*- und *Consumption*-Kante; Beziehung zwischen zwei benachbarten regulierenden Elementen (hier: *Stimulation*), die ein Edukt/Produkt teilen; Berücksichtigung aller Edukte, Produkte und regulierenden Elemente einer Reaktion.

Die Anwendung der Regel p_6 hängt von der Anwendung der Regeln $p_1 - p_4$ ab, nur wenn *Consumption*-, *Production*- und regulierende Kanten entsprechend umgewandelt werden, kann auch Regel p_6 angewendet werden.

Als Ergänzung zu den Umwandlungsregeln in den Tabellen 5.1 bis 5.6 sind in Anhang B Abbildungsvorschriften für die Umwandlung von SBGN *Process Description*-Karten in SBGN *Activity Flow*-Karten zusammengefasst.

	regulierende Kante	regulierende Kante
regulierende Kante	\updownarrow	\updownarrow
	<i>Consumption-Kante</i>	<i>Production-Kante</i>
<i>Modulation</i>	<i>Unknown Influence</i>	<i>Unknown Influence</i>
<i>Stimulation</i>	<i>Negative Influence</i> <i>Unknown Influence</i>	<i>Positive Influence</i>
<i>Catalysis</i>	<i>Negative Influence</i> <i>Unknown Influence</i>	<i>Positive Influence</i>
<i>Inhibition</i>	<i>Positive Influence</i> <i>Unknown Influence</i>	<i>Negative Influence</i>
<i>Necessary Stimulation</i>	<i>Negative Influence</i> <i>Unknown Influence</i>	<i>Necessary Stimulation</i>

Tabelle 5.7: Empfohlene Umwandlungen für die Beziehungen regulierende Kante \leftrightarrow *Production-Kante* und regulierende Kante \leftrightarrow *Consumption-Kante*.

5.2 Templates

Im vorherigen Abschnitt wurden Regeln für die Umwandlung von PD-Karten in AF-Karten beschrieben. Diese Regeln bieten die Möglichkeit für eine flexible Umwandlung. Während der Entwicklung der hier vorgestellten Methodik zur Umwandlung einer PD-Karte in eine AF-Karte konnten vier Templates beschrieben werden, die Umwandlungsregeln für spezielle Anwendungsfälle zusammenfassen (siehe Tabelle 5.8 für eine Übersicht). Diese Templates werden im Folgenden unter Verwendung eines kleinen Beispiels (siehe PD-Karte in Abbildung 5.2) erläutert. Das vollständige Beispiel ist für alle vier Templates in Anhang C gezeigt. Da die Templates auf Stoffwechselnetzwerke fokussieren, werden in den Templates Edukte und Produkte vereinfacht als Metabolite bezeichnet und als regulierende Elemente werden nur Enzyme berücksichtigt.

Template	Regeln	Beispiel
„Enzym-Aktivitäten“	$n_1-n_6, a_2, cl_3, p_3-p_4$	siehe Abbildung 5.3
„Metabolit-Aktivitäten“	$n_1-n_6, a_2, cl_3, p_1-p_2$	siehe Abbildung 5.4
„Enzym- und Metabolit-Aktivitäten“	n_1-n_6, a_2, cl_3, p_6	siehe Abbildung 5.5
„Mittelbare Enzym-Aktivitäten“	n_1-n_6, a_1, cl_3, p_5	siehe Abbildung 5.6

Tabelle 5.8: Übersicht über Templates für die Umwandlung von PD-Karten in AF-Karten.

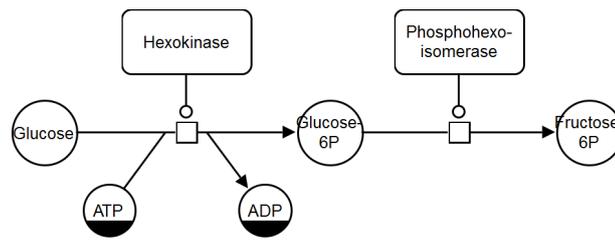


Abbildung 5.2: Beispiel einer PD-Karte, die die ersten beiden Reaktionen der Glykolyse [102] darstellt, übernommen und angepasst aus [116].

5.2.1 Template „Enzym-Aktivitäten“

Das Template „Enzym-Aktivitäten“ kann angewendet werden, um den Einfluss der Enzyme auf die Metabolite zu zeigen.

Im Fall einer irreversiblen Reaktion erzeugt dieses Template in Richtung der Produkte eine regulierende Kante in AF entsprechend der regulierenden Kante in PD, z. B.

- eine *Positive Influence*-Kante für eine *Catalysis*-Kante oder eine *Stimulation*-Kante,
- eine *Negative Influence*-Kante für eine *Inhibition*-Kante,
- eine *Unknown Influence*-Kante für eine *Modulation*-Kante und
- eine *Necessary Stimulation*-Kante für eine *Necessary Stimulation*-Kante.

In Richtung der Edukte wird eine *Unknown Influence*-Kante erzeugt. Im Fall einer reversiblen Reaktion wird für beide Richtungen der Reaktion dieselbe Regel angewendet wie für die Produkte bei einer irreversiblen Reaktion.

Die AF-Karte in Abbildung 5.3 zeigt für das Beispiel aus Abbildung 5.2 die Anwendung des Templates „Enzym-Aktivitäten“.

Regeln: n_1-n_6 , a_2 , cl_3 , p_3-p_4

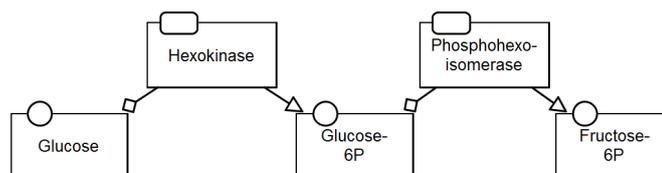


Abbildung 5.3: Beispiel für die Umwandlung in AF unter Anwendung des Templates „Enzym-Aktivitäten“. Die AF-Karte entspricht der PD-Karte in Abbildung 5.2.

5.2.2 Template „Metabolit-Aktivitäten“

Das Template „Metabolit-Aktivitäten“ konzentriert sich auf die *Consumption*- und die *Production*-Kanten der PD-Karte. Es wird ohne Berücksichtigung regulierender Kanten dargestellt, wie die Metabolite umgewandelt werden. Dieses Template kann genutzt werden, wenn der Stofffluss im System gezeigt werden soll. Abbildung 5.4 zeigt die AF-Karte für das Beispiel aus Abbildung 5.2 bei Anwendung des Templates „Metabolit-Aktivitäten“.

Regeln: n_1-n_6 , a_2 , cl_3 , p_1-p_2

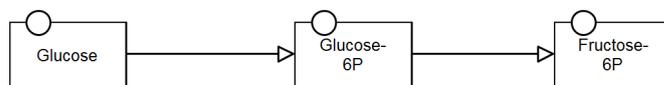


Abbildung 5.4: Beispiel für die Umwandlung in AF unter Anwendung des Templates „Metabolit-Aktivitäten“. Die AF-Karte entspricht der PD-Karte in Abbildung 5.2.

5.2.3 Template „Enzym- und Metabolit-Aktivitäten“

Das Template „Enzym- und Metabolit-Aktivitäten“ kombiniert die beiden zuvor beschriebenen Templates. Die Regeln dieses Templates erzeugen eine AF-Karte, die der ursprünglichen PD-Karte ähnlich ist, die *Process Node*-Knoten werden durch *Logical Operator*-Knoten ersetzt. Ein Nachteil dabei ist, dass *Logical Operator*-Knoten nur eine ausgehende Kante haben dürfen. Wenn ein *Process Node*-Knoten mehr als eine ausgehende *Production*-Kante hat, wird für jede dieser Kanten ein *Logical Operator*-Knoten AND der AF-Karte hinzugefügt. Wenn ein *Process Node*-Knoten mehr als eine eingehende regulierende Kante hat, werden alle regulierenden Kanten des gleichen Typs mit einem neuen *Logical Operator*-Knoten OR verbunden, der dann mit dem *Logical Operator*-Knoten AND verbunden wird.

Die AF-Karte in Abbildung 5.5 zeigt für das Beispiel aus Abbildung 5.2 die Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“.

Regeln: n_1-n_6 , a_2 , cl_3 , p_6

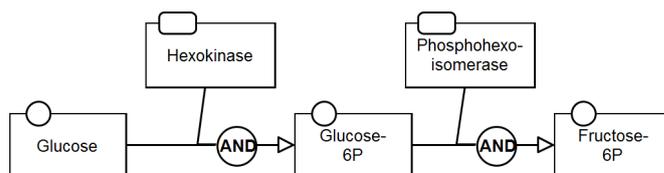


Abbildung 5.5: Beispiel für die Umwandlung in AF unter Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“. Die AF-Karte entspricht der PD-Karte in Abbildung 5.2.

5.2.4 Template „Mittelbare Enzym-Aktivitäten“

Das Template „Mittelbare Enzym-Aktivitäten“ zeigt die zeitliche Abhängigkeit von Enzym-Aktivitäten, ohne dabei Details über die zu Grunde liegenden Prozesse darzustellen. Abbildung 5.6 zeigt die AF-Karte für das Beispiel aus Abbildung 5.2 bei Anwendung des Templates „Mittelbare Enzym-Aktivitäten“.

Regeln: n_1-n_6 , a_1 , cl_3 , p_5

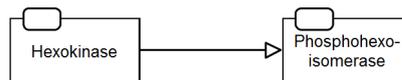


Abbildung 5.6: Beispiel für die Umwandlung in AF unter Anwendung des Templates „Mittelbare Enzym-Aktivitäten“. Die AF-Karte entspricht der PD-Karte in Abbildung 5.2.

5.3 Ablauf der Umwandlung

Im ersten Schritt der Umwandlung werden alle *Entity Pool Node*-Knoten der PD-Karte in die entsprechenden *Biological Activity*-Knoten der zu erstellenden AF-Karte umgewandelt, dabei wird für *Complex*-Knoten implizit die entsprechende Regel (co_1 oder co_2) angewendet. Knoten mit demselben Knotennamen aber unterschiedlichen *Auxiliary Units* werden nach der ausgewählten Regel (a_1 oder a_2) umgewandelt. In Abhängigkeit von der gewählten Regel (cl_1 , cl_2 oder cl_3) werden Knoten mit einem *Clone Marker* entweder in einem *Biological Activity*-Knoten zusammengefasst, fortlaufend nummeriert oder der AF-Karte nicht hinzugefügt.

Im zweiten Schritt werden die Kanten entsprechend der gewählten Regeln umgewandelt. Dabei sind speziell die *Process Node*-Knoten von Bedeutung. In einer Iteration wird für jeden dieser Knoten untersucht, welche Regel (p_1-p_6) für die Kanten angewendet werden muss.

Im letzten Schritt kann es notwendig sein, die erstellte AF-Karte von ungültigen oder redundanten Elementen zu bereinigen. So können z.B. *Logical Operator*-Knoten mehrere ausgehende Kanten haben (siehe auch Abbildung 5.1). In diesen Fällen wird der *Logical Operator*-Knoten einschließlich aller eingehenden Kanten entsprechend der Anzahl ausgehender Kanten vervielfältigt. Darüber hinaus können die *Logical Operator*-Knoten AND und OR nur eine eingehende *Logic Arc*-Kante haben. Dies kann passieren, wenn Knoten während der Umwandlung der AF-Karte nicht hinzugefügt wurden. Diese *Logical Operator*-Knoten sind ungültig und werden von der AF-Karte entfernt. Die Ausgangs- und Zielknoten der Kanten dieser *Logical Operator*-Knoten werden direkt miteinander verbunden.

Der beschriebene Umwandlungsalgorithmus behält die grafische Darstellung aller Knoten in der AF-Karte in Größe und Position entsprechend der Darstellung in der PD-Karte bei, um die *mental map* der PD-Karte während der Umwandlung zu erhalten. Dennoch gibt es einige Ausnahmen für die Positionierung von Knoten in der AF-Karte: (1) wenn Knoten zu

einem neuen Knoten zusammengefasst werden, wird die mittlere Position dieser Knoten als Position für den neuen Knoten berechnet und (2) wenn ein Knoten vervielfältigt wird, werden die neuen Knoten spiralförmig um die Position des ursprünglichen Knotens angeordnet.

5.4 Beispiele

Es wird die Umwandlung der PD-Karte eines Signalweges (MAPK-Kaskade [93, 132, 133]) in eine AF-Karte unter Anwendung eines Templates und einer manuellen Auswahl an Regeln gezeigt.

Die Signalweg-Karte in PD (übernommen und angepasst aus [116]) ist in den Abbildungen 5.7 und 5.8 (jeweils links) gezeigt. Die PD-Karte wurde zunächst unter Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“ in eine AF-Karte umgewandelt (siehe Abbildung 5.7 rechts). Die Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“ vereinfacht die Darstellung leicht, da die Kofaktoren ATP und ADP nicht mit umgewandelt werden. Die Umwandlung der PD-Karte in eine AF-Karte unter Anwendung einer manuellen Auswahl an Regeln ist in Abbildung 5.8 (rechts) dargestellt. Die manuelle Auswahl der Regeln n_1 – n_6 , a_1 , cl_3 und p_3 reduziert die Darstellung auf die vier Enzyme. Diese Darstellung verdeutlicht, wie sich aus einer mechanistischen Beschreibung eines biologischen Systems [84] durch die Anwendung von Regeln eine vereinfachte Darstellung in AF ableiten lässt, die in ähnlicher Form auch in der Literatur gefunden werden kann [93, 133].

In Anhang C ist ein weiteres Beispiel zu sehen. Es wird die Umwandlung einer Stoffwechselweg-Karte (Glykolyse) unter Anwendung aller vier Templates gezeigt.

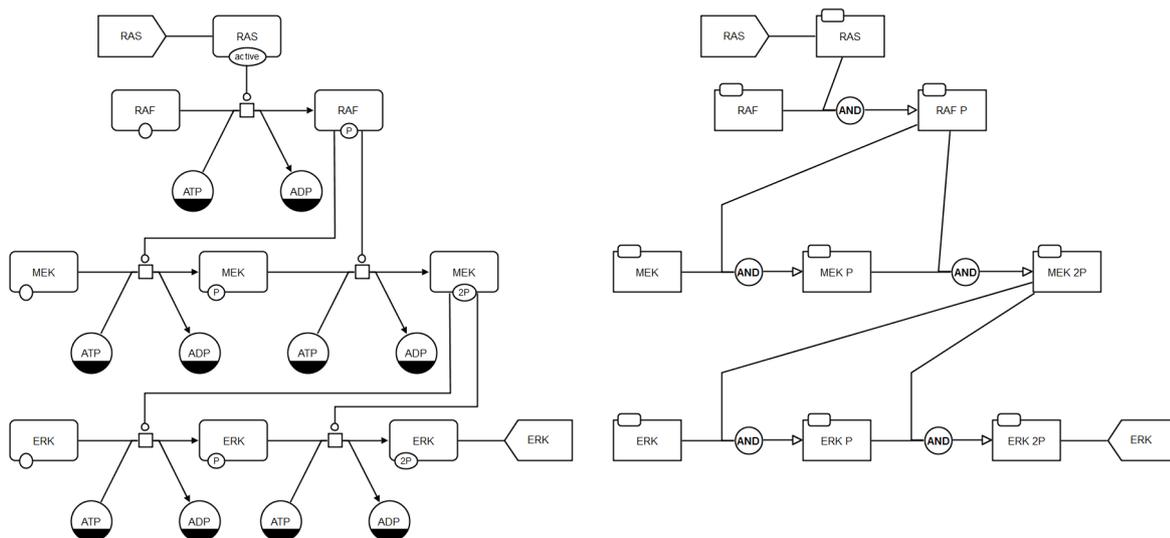


Abbildung 5.7: Umwandlung einer PD-Karte in eine AF-Karte am Beispiel der MAPK-Kaskade, übernommen und angepasst aus [116]. Links: die originale PD-Karte, rechts: die übersetzte AF-Karte unter Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“.

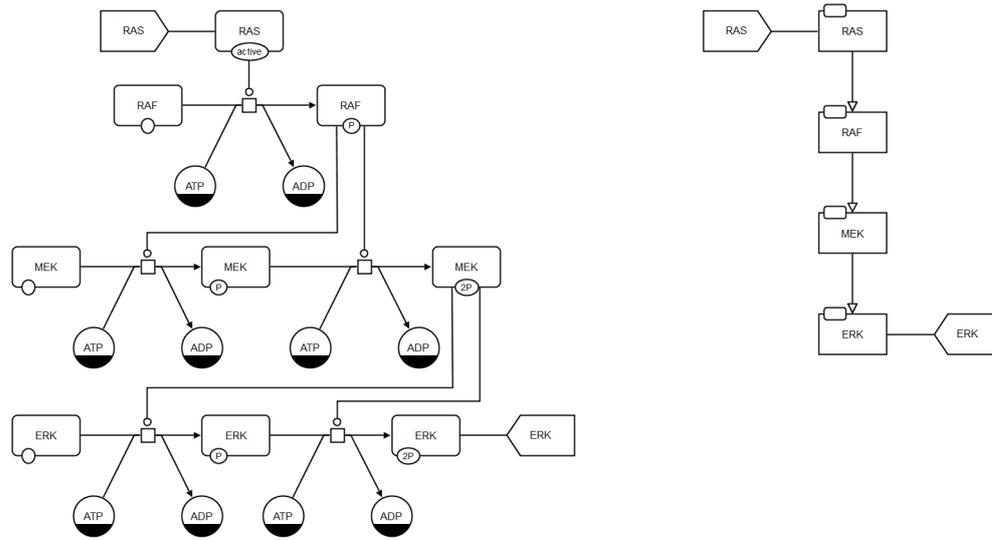


Abbildung 5.8: Umwandlung einer PD-Karte in eine AF-Karte am Beispiel der MAPK-Kaskade, übernommen und angepasst aus [116]. Links: die originale PD-Karte, rechts: die übersetzte AF-Karte. Die Umwandlungsregeln wurden manuell ausgewählt.

5.5 Diskussion und Zusammenfassung

In diesem Kapitel wurde eine Methode zur Umwandlung von *Process Description*-Karten in *Activity Flow*-Karten präsentiert. Die Umwandlung einer PD-Karte in eine AF-Karte ermöglicht die Darstellung biologischen Wissens in einer weiteren Sicht sowie in einer kompakten und übersichtlichen Form.

Für die Umwandlung wurden auf Grundlage der beiden Sprachen Regeln definiert, auf deren Basis vier Templates beschrieben werden konnten. Anhand einiger Beispiele wurde die Umwandlung von *Process Description*-Karten in *Activity Flow*-Karten unter Anwendung der Templates sowie manuell ausgewählter Regeln demonstriert. Momentan beschränken sich die Templates auf Stoffwechselnetzwerke, eine Erweiterung auf andere Netzwerke ist möglich.

Das hier vorgestellte und diskutierte Problem unterscheidet sich von zwei anderen wichtigen Problemen: (1) der Übersetzung von Informationen zu biologischen Netzwerken, zellulären Prozessen und mathematischen Modellen in eine grafische Darstellung in SBGN und (2) der Übersetzung mathematischer Modelle zwischen verschiedenen Repräsentationen. Für (1) gibt es Beispiele zur Übersetzung von SBML, BioPAX und KGML (KEGG) in SBGN: für eine Übersetzung von KGML (KEGG) in SBGN siehe Kapitel 4, für eine Übersetzung von SBML in SBGN siehe Abschnitt 9.3 und für eine Übersetzung von BioPAX in SBGN siehe [35]. Die Übersetzung von BioPAX zu SBML qual [21] und die Übersetzung von Reaktionsgraphen in Aktivierungs-/Inhibitionsgraphen [46] sind Beispiele für (2). Eine SBGN-Karte ist eine grafische Darstellung biologischen Wissens und sollte nicht als simulierbares mathematisches Modell eines biologischen Prozesses interpretiert werden.

Die vorgestellte Methodik wurde implementiert (siehe auch Abschnitt 8.2) und kann angewendet werden [147]. Bei der Anwendung kann es passieren, dass Knoten, wie in Abschnitt 5.3 beschrieben, zusammengefasst oder vervielfältigt werden und neue Positionen für Knoten berechnet werden. Dadurch kann es zu Überlappungen von Knoten kommen. Diese Überlappungen werden nicht algorithmisch beseitigt und müssen gegebenenfalls in einer Nachbearbeitung korrigiert werden, um eine AF-Karte zu erhalten, die den SBGN Layout-Richtlinien folgt. Die Methode könnte deshalb noch um einen Layout-Algorithmus erweitert werden, der Knotenüberlappungen beseitigt, dazu sollte ein Verfahren angewendet werden, das die *mental map* der Karte erhält.

Eine flexible Umwandlung von PD-Karten in AF-Karten und die entwickelte Methodik sind ein erster Schritt bei der Umwandlung zwischen den einzelnen SBGN-Sprachen und ermöglicht neue Anwendungen für SBGN PD-Karten [126]. Darauf aufbauend können weitere Methoden zur Umwandlung zwischen anderen SBGN-Sprachen entwickelt werden.

Kapitel 6

SBGN Bricks

Die Spezifikationen der drei SBGN-Sprachen definieren jeweils eine Anzahl von Elementen, die auf einer Karte in einer der Sprachen verwendet werden können. Zusätzlich wird noch beschrieben, wie diese Elemente untereinander verbunden werden können und welche Hierarchien von Elementen möglich sind, um eine valide SBGN-Karte zu erstellen. Es wird aber nicht im Detail dargelegt, wie bestimmte biologische Prozesse in SBGN dargestellt werden. Anwender können sich an einigen Beispielen in den Spezifikationen orientieren, aber oftmals wird trotzdem nicht deutlich, wie bestimmtes biologisches Wissen in SBGN repräsentiert werden kann und welche der drei Sprachen dafür am besten geeignet ist. Darüber hinaus ist es eine zeitaufwendige Aufgabe, SBGN-Karten auf der Basis einzelner Elemente zu erstellen, die häufig wiederkehrende biologische Prozesse zeigen.

In diesem Kapitel soll deshalb das Konzept der SBGN *Bricks* vorgestellt werden, ein Template-basierter Ansatz zum Erlernen und zum Anwenden von SBGN.

SBGN Bricks sind *Bausteine*, die eine Anzahl grundlegender biologischer Prozesse, die in verschiedenen biologischen Netzwerken (Stoffwechselnetzwerke, Signalnetzwerke, Protein-Protein-Interaktionsnetzwerke und Genregulationsnetzwerke) stattfinden können, in den drei SBGN-Sprachen repräsentieren. Die Bricks ermöglichen es einem Anwender, eine SBGN-Karte zu erstellen, ohne die Spezifikationen im Detail zu kennen. Da sie zeigen, wie bestimmte biologische Prozesse in SBGN dargestellt werden können, unterstützen sie den Anwender bei der Entscheidung für eine der drei Sprachen. Die Anwendung der Bricks verringert zudem die Zeit, die benötigt wird, um eine SBGN-Karte zu erstellen, da die Karte aus den Bricks zusammengesetzt werden kann.

Im nächsten Abschnitt wird das Konzept allgemein beschrieben und für Genregulationsnetzwerke werden die Bricks im Detail dargestellt. Im nachfolgenden Abschnitt werden diese Bricks dann in einem Beispiel angewendet.

6.1 Biologische Prozesse und ihre Repräsentation als SBGN Brick

Zelluläre Vorgänge basieren auf einer Reihe grundlegender biologischer Prozesse, wie z. B. Stoffwechselreaktionen, Transkription oder Translation. Diese Prozesse wiederum bilden die Grundlage für verschiedene biologische Netzwerke, in denen sie wiederholt auftreten können. Solche wiederholt vorkommenden biologischen Prozesse werden hier abstrakt als SBGN Bricks repräsentiert, die verwendet werden können, um vereinfacht SBGN-Karten zu erstellen (siehe Abbildung 6.1).

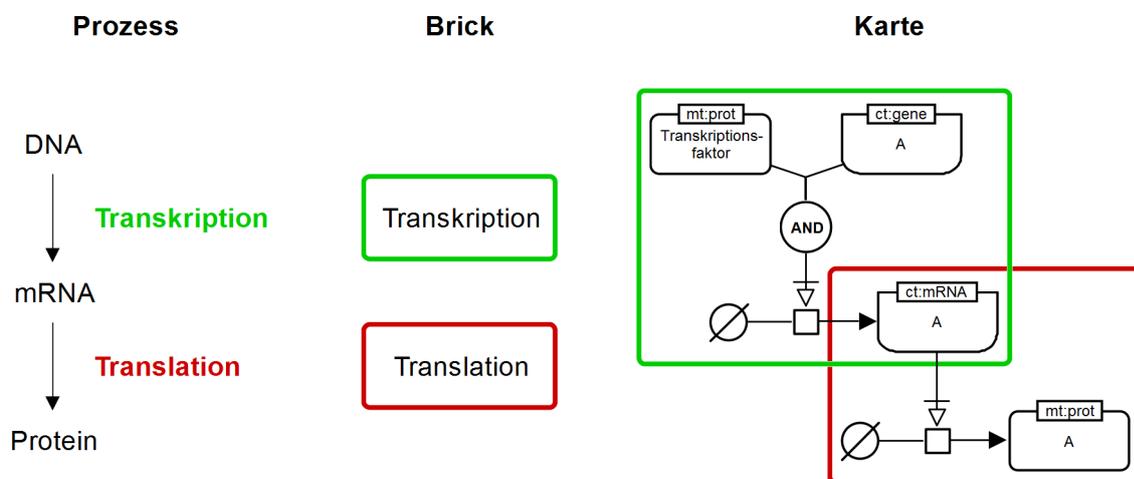


Abbildung 6.1: Das Konzept der SBGN Bricks beschreibt grundlegende biologische Prozesse als Bricks, die zu einer SBGN-Karte zusammengefügt werden können. Hier wird z. B. die Umwandlung von DNA über mRNA in ein Protein (links) abstrakt als die beiden Bricks „Transkription“ und „Translation“ beschrieben (Mitte), die dann zu einem Genregulationsnetzwerk zusammengefügt werden (rechts).

In Tabelle 6.1 sind die Bricks für verschiedene biologische Netzwerke aufgeführt und kurz erläutert. In Abschnitt 6.1.1 werden die Bricks für Genregulationsnetzwerke im Detail vorgestellt. Für die Bricks aller anderen Netzwerke sind Abbildungen in Anhang D zu finden. Tabelle 6.2 gibt zudem einen Überblick zur Verfügbarkeit der Bricks in den drei SBGN-Sprachen. Im Prinzip kann jeder biologische Prozess in jeder der drei Sprachen dargestellt werden, je nach Art des biologischen Prozesses eignen sie sich jedoch unterschiedlich gut z. B. hinsichtlich der Wichtigkeit mechanistischer Details und Beachtung zeitlicher Aspekte. Es wird deshalb zwischen „verfügbar“ und „nicht verfügbar“ unterschieden (in Tabelle 6.2 dargestellt durch „+“ und „-“). Diese Unterscheidung gibt einen Hinweis darauf, wie gut sich eine Sprache für die darzustellenden Prozesse eignet.

So ist der Brick „Katalyse“ für die *Entity Relationship*-Sprache nicht verfügbar, da die Repräsentation von Beziehungen zwischen Elementen im Mittelpunkt der Sprache steht. Bio-

Brick	Beschreibung
Stoffwechselnetzwerk	
Katalyse	repräsentiert Stoffwechselreaktionen, steht für eine unterschiedliche Anzahl von Edukten und Produkten sowie für irreversible und reversible Reaktion zur Verfügung
Signalnetzwerk	
Protein-Phosphorylierung	repräsentiert einen Prozess, der einem Protein eine Phosphatgruppe hinzufügt
Protein-Protein-Interaktionsnetzwerk	
Komplexbildung	repräsentiert die Bildung eines Proteinkomplexes aus Elementen
Komplexdissoziation	repräsentiert die Auflösung eines Proteinkomplexes in seine Elemente
Oligomerisierung / Homodimerisierung	repräsentiert die Bildung von Dimeren oder Multimeren
Genregulationsnetzwerk	
Transkription	repräsentiert den Prozess der Bildung von mRNA aus DNA
Translation	repräsentiert den Prozess der Bildung eines Proteins aus mRNA
Sonstige	
Funktionelle Beziehung	repräsentiert den Zusammenhang zwischen dem Einfluss eines (biologischen) Ereignisses auf ein Element, das dann selbst ein anderes (biologisches) Ereignis beeinflussen kann
Transport	repräsentiert die aktive (mit Transporter) oder passive (durch Diffusion) Bewegung von Elementen aus oder in ein Kompartiment

Tabelle 6.1: Übersicht über die SBGN Bricks für verschiedene biologische Netzwerke. In Tabelle 6.3 sind die Bricks für Genregulationsnetzwerke abgebildet. Für die Bricks aller anderen Netzwerke sind Abbildungen in Anhang D zu finden.

chemische Reaktionen in Stoffwechselnetzwerken müssten als Interaktionen zwischen Edukten von Reaktionen und den diese Reaktionen katalysierenden Enzymen dargestellt werden. Diese Interaktionen würden dann dazu führen, dass den *Auxiliary Units Existence* der Produkte dieser Reaktionen der Wert „wahr“ oder „falsch“ zugewiesen wird. Derartig komplizierte Darstellungen erschweren die Interpretation einer Karte und entsprechende Bricks wurden nicht

		PD	ER	AF
Stoffwechselnetzwerk				
Katalyse	irreversibel (1 Edukt, 1 Produkt)	+	-	+
	irreversibel (2 Edukte, 1 Produkt)	+	-	+
	irreversibel (1 Edukt, 2 Produkte)	+	-	-
	irreversibel (2 Edukte, 2 Produkte)	+	-	-
	reversibel (1 Edukt, 1 Produkt)	+	-	-
	reversibel (2 Edukt, 1 Produkt)	+	-	-
	reversibel (1 Edukte, 2 Produkte)	+	-	-
	reversibel (2 Edukte, 2 Produkte)	+	-	-
Signalnetzwerk				
Protein-Phosphorylierung		+	+	+
Protein-Protein-Interaktionsnetzwerk				
Komplexbildung		+	+	+
Komplexdissoziation		+	+	+
Oligomerisierung / Homodimerisierung		+	+	+
Genregulationsnetzwerk				
Transkription	mit Bindung	+	+	+
	ohne Bindung	+	+	+
Translation		+	+	+
Sonstige				
Funktionelle Beziehung		-	-	+
Transport	aktiv (Transporter)	+	+	+
	passiv (Diffusion)	+	+	+

Tabelle 6.2: Verfügbarkeit der SBGN Bricks in den drei SBGN-Sprachen für verschiedene biologische Netzwerke.

berücksichtigt. Ähnliches gilt für die Darstellung biochemischer Reaktionen in der *Activity Flow*-Sprache. Entsprechend der Anzahl Produkte einer Reaktion müssten der Karte logische Operatoren hinzugefügt werden. Bei mehr als einem Produkt führt auch das zu einer komplizierten Darstellung.

Als Besonderheit ermöglicht die *Activity Flow*-Sprache die abstrakte Darstellung biologischer Prozesse mit einer eingeschränkten Anzahl an Details. Der Brick „Funktionelle Bezie-

hung“ lässt sich deshalb nicht für die anderen beiden Sprachen darstellen, da für diese ein höheres Maß an Details notwendig wäre.

6.1.1 Bricks für Genregulationsnetzwerke

In diesem Abschnitt sollen die Bricks für Genregulationsnetzwerke für jede SBGN-Sprache detaillierter vorgestellt werden. Für die *Activity Flow*-Sprache existieren oftmals mehrere Möglichkeiten zur Repräsentation eines biologischen Prozesses (siehe auch Kapitel 5), hier wird jedoch nur eine Möglichkeit dargestellt.

Für Genregulationsnetzwerke wurden die drei Bricks „Transkription mit Bindung“, „Transkription ohne Bindung“ und „Translation“ definiert. Diese Bricks sind in Tabelle 6.3 dargestellt und werden im Folgenden näher erläutert. Zusätzlich sind in Tabelle 6.3 den Prozessen (und damit auch den Bricks) ein oder mehrere GO-Terme aus der Hierarchie der *Gene Ontology* [6, 138] zugeordnet. Dadurch ist es möglich, die Definition der Bricks zu standardisieren und die Prozesse unabhängig von den SBGN-Sprachen zu beschreiben.

Prozess	Brick in PD	Brick in ER	Brick in AF
Transkription mit Bindung GO:0006351 GO:0006355 GO:0009299 GO:0065004			
Transkription ohne Bindung GO:0006351 GO:0006355 GO:0009299			
Translation GO:0006412			

Tabelle 6.3: SBGN Bricks für Genregulationsnetzwerke (PD – *Process Description*, ER – *Entity Relationship*, AF – *Activity Flow*).

Brick „Transkription mit Bindung“

- *Process Description*-Sprache: Das *Protein Transkriptionsfaktor* und *Gen A* binden zu einem Komplex. Dieser stimuliert den Prozess der Transkription (notwendige Stimulation), der aus einer Menge von Edukten, die nicht näher spezifiziert werden, zur Bildung der *mRNA* von *Gen A* führt. Eine Bindung zwischen Transkriptionsfaktor und *Gen* ist aus der Literatur bekannte (sie konnte experimentell verifiziert werden).
- *Entity Relationship*-Sprache: Die *Entity Transkriptionsfaktor* mit dem Typ *Protein* und die *Entity A* mit dem Typ *Gen* interagieren, sie binden. Diese Bindung wird durch ein *Outcome* repräsentiert. Das *Outcome* stimuliert die Zuweisung des Wertes *T* (*true*, wahr) zur *Auxiliary Unit* vom Typ *Existence* der *Entity A* mit dem Typ *mRNA* (notwendige Stimulation). Eine Bindung zwischen Transkriptionsfaktor und *Gen* ist aus der Literatur bekannte (sie konnte experimentell verifiziert werden).
- *Activity Flow*-Sprache: Die *Biological Activity Transkriptionsfaktor* mit dem Typ *Protein* und der *Auxiliary Unit* vom Typ *Macromolecule* und die *Biological Activity A* mit dem Typ *Gen* und der *Auxiliary Unit* vom Typ *Nucleic Acid Feature* stimulieren zusammen die *Biological Activity Transkriptionsfaktor - A* mit der *Auxiliary Unit* vom Typ *Complex*. Diese *Biological Activity* repräsentiert die Bindung und stimuliert die *Biological Activity A* mit dem Typ *mRNA* und der *Auxiliary Unit* vom Typ *Nucleic Acid Feature* (notwendige Stimulation). Eine Bindung zwischen Transkriptionsfaktor und *Gen* ist aus der Literatur bekannte (sie konnte experimentell verifiziert werden).

Brick „Transkription ohne Bindung“

- *Process Description*-Sprache: Das *Protein Transkriptionsfaktor* und *Gen A* stimulieren zusammen den Prozess der Transkription (notwendige Stimulation), der aus einer Menge von Edukten, die nicht näher spezifiziert werden, zur Bildung der *mRNA* von *Gen A* führt. Eine Bindung zwischen Transkriptionsfaktor und *Gen* ist aus der Literatur nicht bekannt (sie konnte experimentell nicht verifiziert werden), deshalb wird der *Logical Operator AND* verwendet, um die Wechselwirkung zwischen Transkriptionsfaktor und *Gen* zu repräsentieren.
- *Entity Relationship*-Sprache: Die *Entity Transkriptionsfaktor* mit dem Typ *Protein* und die *Entity A* mit dem Typ *Gen* stimulieren zusammen die Zuweisung des Wertes *T* (*true*, wahr) zur *Auxiliary Unit* vom Typ *Existence* der *Entity A* mit dem Typ *mRNA* (notwendige Stimulation). Eine Bindung zwischen Transkriptionsfaktor und *Gen* ist aus der Literatur nicht bekannt (sie konnte experimentell nicht verifiziert werden), deshalb wird der *Logical Operator AND* verwendet, um die Interaktion zwischen Transkriptionsfaktor und *Gen* zu repräsentieren.

- *Activity Flow*-Sprache: Die *Biological Activity Transkriptionsfaktor* mit dem Typ *Protein* und der *Auxiliary Unit* vom Typ *Macromolecule* und die *Biological Activity A* mit dem Typ *Gen* und der *Auxiliary Unit* vom Typ *Nucleic Acid Feature* stimulieren zusammen die *Biological Activity A* mit dem Typ *mRNA* und der *Auxiliary Unit* vom Typ *Nucleic Acid Feature* (notwendige Stimulation). Eine Bindung zwischen Transkriptionsfaktor und Gen ist aus der Literatur nicht bekannt (sie konnte experimentell nicht verifiziert werden), deshalb wird der *Logical Operator AND* verwendet, um die Wechselwirkung zwischen Transkriptionsfaktor und Gen zu repräsentieren.

Brick „Translation“

- *Process Description*-Sprache: Die *mRNA* von Gen *A* stimuliert den Prozess der Translation (notwendige Stimulation), der aus einer Menge von Edukten, die nicht näher spezifiziert werden, zur Bildung des *Proteins A* führt.
- *Entity Relationship*-Sprache: Die *Entity A* mit dem Typ *mRNA* stimuliert die Zuweisung des Wertes *T* (*true*, wahr) zur *Auxiliary Unit* vom Typ *Existence* der *Entity A* mit dem Typ *Protein* (notwendige Stimulation).
- *Activity Flow*-Sprache: Die *Biological Activity A* mit dem Typ *mRNA* und der *Auxiliary Unit* vom Typ *Nucleic Acid Feature* stimuliert die *Biological Activity A* mit dem Typ *Protein* und der *Auxiliary Unit* vom Typ *Macromolecule* (notwendige Stimulation).

6.1.2 Das Bricks-Wörterbuch

Das Konzept der SBGN Bricks soll es Anwendern erleichtern, SBGN anzuwenden und zu erlernen. Dies setzt natürlich voraus, dass die Bricks auch frei verfügbar und öffentlich zugänglich sind. Es wurde deshalb ein Wiki eingerichtet, das eine initiale Zusammenstellung an Bricks (siehe auch Tabellen 6.1 und 6.2) enthält¹. Das Wiki enthält eine Übersicht aller Bricks mit einer detaillierten Erklärung jedes Bricks und kann deshalb auch als *Bricks-Wörterbuch* betrachtet werden. Die Bereitstellung der Bricks in einem Wiki hat den Vorteil, dass sie bei entsprechendem Bedarf durch Anwender aus der Community erweitert oder angepasst werden können.

Zusätzlich stehen die SBGN Bricks im Wiki zum Download im SBGN-ML Format (siehe auch Kapitel 7) zur Verfügung. Software, die Unterstützung für SBGN und SBGN-ML anbietet, kann die Bricks integrieren und Anwender können sie dann zum vereinfachten Erstellen von SBGN-Karten verwenden (siehe auch Kapitel 8).

¹<http://sbgnbricks.sourceforge.net>

6.2 Beispiel

Das Erstellen von SBGN-Karten in allen drei Sprachen unter Anwendung der SBGN Bricks soll in diesem Abschnitt am Beispiel eines Genregulationsnetzwerkes gezeigt werden. Dabei werden die im Abschnitt 6.1.1 im Detail dargestellten Bricks „Transkription mit Bindung“ und „Translation“ genutzt.

In Abbildung 6.2 ist ein Genregulationsnetzwerk in allen drei SBGN-Sprachen zu sehen, dass eine zweistufige Regulationskaskade zeigt. Es wurden in jeder Sprache zwei Bricks „Transkription mit Bindung“ (grün umrandet) und ein Brick „Translation“ (rot umrandet) zur Erstellung der Karten verwendet. Im ersten Schritt der Regulationskaskade stimuliert der *Transkriptionsfaktor 1* die Transkription von *Gen A* in die *mRNA* von *Gen A*. Im zweiten Schritt (der Translation) wird die *mRNA* von *Gen A* in *Protein A* übersetzt und stimuliert dann als *Transkriptionsfaktor 2* im dritten Schritt die Transkription von *Gen B* in die *mRNA* von *Gen B*.

Das Zusammensetzen einer SBGN-Karte aus SBGN Bricks erfordert im Wesentlichen drei Schritte in der Nachbearbeitung: (1) die Knotennamen müssen entsprechend angepasst werden, (2) identische Knoten aus verschiedenen Bricks müssen zusammengeführt werden und (3) in der *Process Description*-Sprache müssen gegebenenfalls *Clone Marker* hinzugefügt werden, um eine valide Karte zu erhalten.

Im Beispiel in Abbildung 6.2 wurden jeweils der Knoten, der die *mRNA* von *Gen A* repräsentiert (das Ergebnis der ersten Transkription und gleichzeitig der Ausgangspunkt der Translation), und der Knoten, der *Protein A* repräsentiert (das Ergebnis der Translation und gleichzeitig der Transkriptionsfaktor der zweiten Transkription), zusammengeführt.

In einem letzten Schritt ist eventuell noch das Layout der SBGN-Karte anzupassen.

6.3 Zusammenfassung

Das Konzept der SBGN Bricks wurde entwickelt, um Anwendern aus der Systembiologie das Erstellen von SBGN-Karten zu erleichtern. Die SBGN Bricks vereinfachen die Entscheidung für eine der drei Sprachen, um bestimmte biologische Prozesse darzustellen und ermöglichen es, eine Karte zu erstellen, ohne die SBGN-Spezifikationen im Detail zu kennen [76].

SBGN Bricks stellen Templates in allen drei SBGN-Sprachen zur grafischen Darstellung einer Vielzahl von grundlegenden biologischen Prozessen bereit, die leicht zu verschiedenen Arten biologischer Netzwerke (Stoffwechselnetzwerke, Signalnetzwerke, Protein-Protein-Interaktionsnetzwerke und Genregulationsnetzwerke) zusammengesetzt werden können und dabei den Aufwand zur Erstellung einer SBGN-Karte verringern.

Ein Bricks-Wörterbuch ist in einer ersten Version als Wiki online verfügbar und kann bei entsprechendem Bedarf erweitert oder angepasst werden². Die SBGN Bricks stehen dort ebenfalls zum Download im SBGN-ML Format (siehe auch Kapitel 7) bereit und können

²<http://sbgnbricks.sourceforge.net>

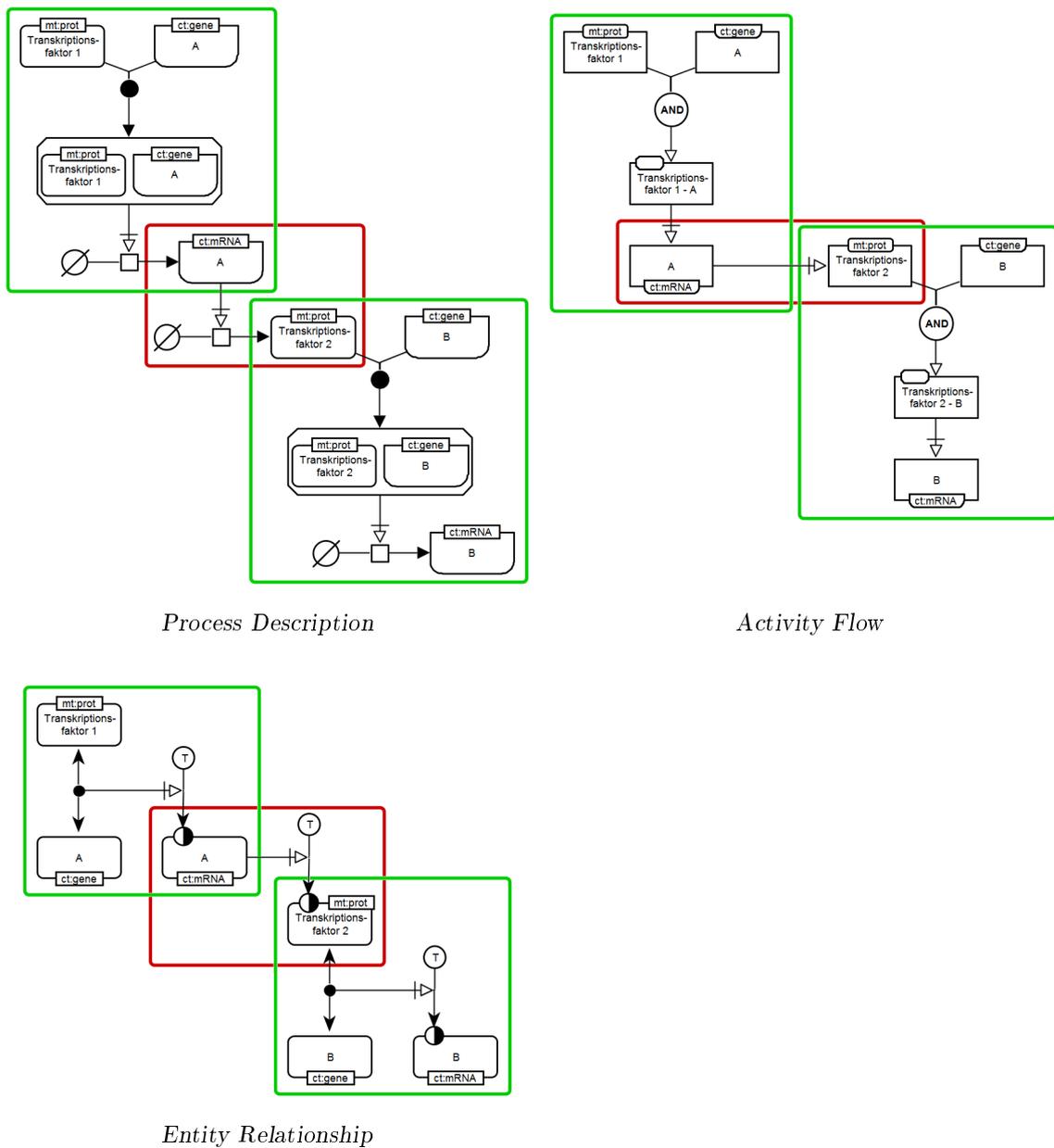


Abbildung 6.2: Beispiel zur Anwendung der SBGN Bricks. Es ist ein Genregulationsnetzwerk in allen drei SBGN-Sprachen dargestellt, das jeweils aus drei SBGN Bricks zusammengesetzt wurde: zwei Bricks „Transkription mit Bindung“ (grün umrandet) und ein Brick „Translation“ (rot umrandet).

in Software, die Unterstützung für SBGN und SBGN-ML bietet, zum Erstellen von SBGN-Karten verwendet werden (siehe auch Kapitel 8).

Kapitel 7

Import und Export von SBGN-Karten

Für eine Akzeptanz und Anwendung von SBGN in der Systembiologie oder der Biologie im Allgemeinen innerhalb kurzer Zeit ist es unabdingbar, dass Software-Anwendungen eine entsprechende Unterstützung anbieten. Es gibt bereits einige Anwendungen, die SBGN in unterschiedlicher Art und Weise unterstützen (siehe auch Abschnitt 3.6). Allerdings sind derzeitige Implementierungen von SBGN in Software oft unvollständig und teilweise nicht korrekt. Eine vollständige und korrekte Implementierung der kompletten SBGN-Spezifikationen ist eine aufwendige Aufgabe. Ein zusätzliches Problem dabei ist, dass SBGN nur definiert, wie biologisches Wissen grafisch dargestellt werden soll, aber nicht wie SBGN-Karten zwischen Software ausgetauscht werden sollen. Die mit einer bestimmten Software erstellten oder bearbeiteten SBGN-Karten können also in der Regel nicht mit einer anderen Software wieder- oder weiterverwendet werden. Neben der standardisierten Darstellung von Information ist aber der Austausch von Informationen eines der Ziele von SBGN. Da andere in der Systembiologie verwendete Formate, wie z. B. SBML oder BioPAX, sich als ungeeignet erwiesen, ergab sich die Notwendigkeit ein spezielles SBGN-Dateiformat zu entwickeln.

Aus diesem Grund wurde das Projekt LibSBGN¹ etabliert, welches das Ziel hat, den Austausch von SBGN-Karten zwischen SBGN-kompatibler Software zu ermöglichen. Dies führte zur Entwicklung des Dateiformates *SBGN-ML* und der Software-Bibliothek *LibSBGN*. Jede dieser beiden Entwicklungen wird in den folgenden beiden Abschnitten erläutert.

7.1 Das Dateiformat SBGN-ML

SBGN-ML ist ein XML-basiertes [113] Dateiformat zum einfachen Austausch von SBGN-Karten zwischen SBGN-kompatibler Software, dass es ermöglicht, sowohl die biologischen als auch die grafischen Informationen einer SBGN-Karte in elektronischer Form abzulegen. Prinzipiell soll SBGN-ML drei Anforderungen erfüllen.

¹<http://www.sbgn.org/LibSBGN>

Es soll

- für einen Computer einfach zu analysieren und zu interpretieren sein,
- für einen Computer einfach darzustellen sein und
- für einen Menschen einfach zu lesen sein.

Der erste Punkt umfasst die Syntax und Semantik von SBGN. Das SBGN-ML Format nutzt dieselben Bezeichner für Elemente wie die SBGN-Spezifikationen (siehe auch Kapitel 3). Dadurch können auch die biologischen Informationen der Karte interpretiert werden. Software-Anwendungen, die eine SBGN-Karte aus einer SBGN-ML Datei grafisch darstellen wollen, müssen die Bezeichner benutzen, um Elemente entsprechend den SBGN-Spezifikationen zeichnen zu können. Das Dateiformat selbst definiert nicht, wie Elemente grafisch repräsentiert werden sollen. Diese Informationen müssen in einer Anwendung implementiert sein.

Aus der SBGN-ML Syntax lassen sich die hierarchischen Beziehungen zwischen Elementen ermitteln. Die Elemente eines *Compartment*-Knotens, die Dekorationen eines *Complex*-Knotens und weitere Elemente wie *Unit of Information*, *State Variable* und *Clone Marker* werden beschrieben. Alle diese Elemente sind Teilelemente eines Knotens oder einer Kante. Des Weiteren lässt sich die Topologie des zu Grunde liegenden biologischen Netzwerkes ableiten, da die Verbindungen zwischen Knoten und Kanten aus den abgelegten Informationen rekonstruiert werden können. Sowohl die hierarchischen Informationen als auch die Topologie können für weitere Analysen genutzt werden.

Die letzten beiden Punkte beziehen sich auf die grafische Darstellung von SBGN. Der Inhalt einer SBGN-ML Datei (die SBGN-Karte) soll einfach auf einem Bildschirm darzustellen sein und Software-Anwendungen sollen keine Berechnungen durchführen, um die Karte anzeigen zu können. Alle notwendigen Informationen wie z. B. die Größe und die Position jedes Knotens sowie die Pfade der Kanten werden mit abgelegt. Dadurch bleibt das Layout einer SBGN-Karte durch die elektronische Ablage im SBGN-ML Format erhalten. Das ist eine wichtige Eigenschaft von SBGN-ML, da das Layout einer SBGN-Karte durch den Autor hinsichtlich Inhalt, Aussehen und Lesbarkeit der Karte bewusst gewählt wurde. Man spricht auch von der *mental map* einer Darstellung. Der Erhalt des Layouts ist wesentlich, um es dem Leser einer SBGN-Karte zu vereinfachen, die dargestellten biologischen Informationen zu verstehen.

Es können SBGN-Karten aller drei Sprachen im SBGN-ML Dateiformat abgelegt werden. In der derzeitigen Spezifikation ist es allerdings nur möglich, alle Informationen abzulegen, die notwendig sind, um die Position und Größe von Knoten und die Pfade von Kanten zu rekonstruieren. Detaillierte Grafikattribute für die Elemente einer SBGN-Karte, wie z. B. Farben, Liniestärken, Schriftgrößen usw., können momentan nicht definiert werden.

Die gesamte Entwicklung von SBGN-ML basiert auf der Definition einer zugehörigen XML Schema-Definition (XSD) [113] für alle drei SBGN-Sprachen, das die Spezifikation der notwendigen Elemente enthält. Zusätzlich wurde ein Element „note“ definiert, das für Kommentare

genutzt werden kann. Einzelne Software-Anwendungen können SBGN-ML durch Elemente in eigenen Namensräumen erweitern, um z. B. zusätzlich benötigte Informationen, die nicht von SBGN-ML definiert sind, zu Importieren und zu Exportieren.

Nachfolgend soll die SBGN-ML Syntax an einem Beispiel näher erläutert werden. Die Abbildung 7.1 zeigt eine SBGN *Process Description*-Karte mit den ersten drei Schritten der Glykolyse [102].

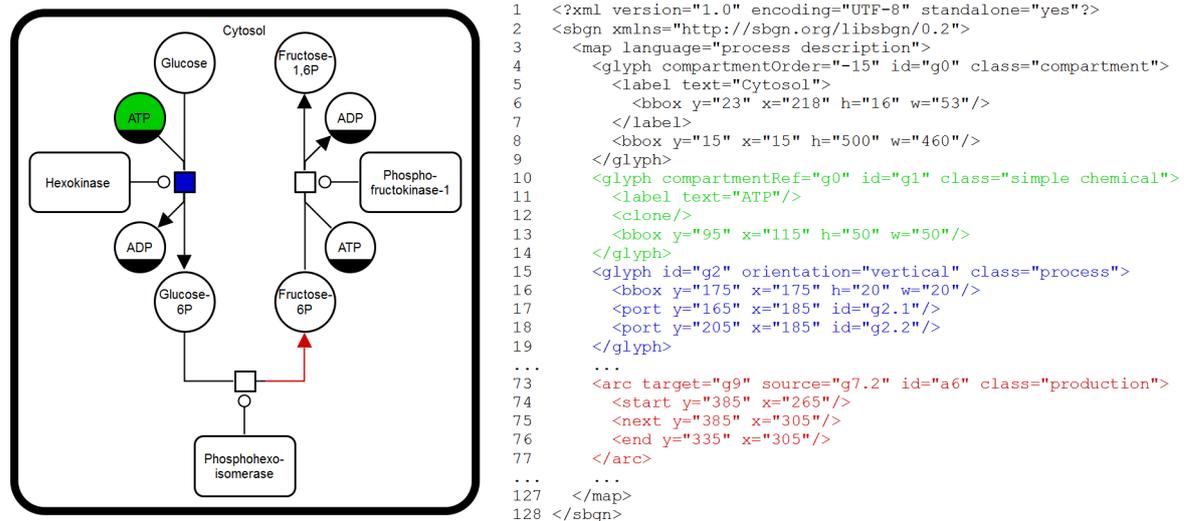


Abbildung 7.1: Es sind die ersten drei Schritte der Glykolyse [102] als SBGN PD-Karte (links, übernommen und angepasst aus [116]) und als SBGN-ML (rechts) dargestellt. Die verwendeten Farben haben keine Bedeutung in SBGN. Es soll lediglich der Zusammenhang zwischen den grafischen Elementen in der Karte und der Darstellung in SBGN-ML gezeigt werden (*Simple Chemical*-Knoten in Grün, *Process*-Knoten in Blau, *Production*-Kante in Rot).

Das Wurzelement einer SBGN-ML Datei heißt „sbgn“ (Zeile 2). Danach folgt ein Element „map“ mit einem Attribut „language“, das die verwendete SBGN-Sprache angibt (Zeile 3). Dann folgen „glyph“- und „arc“-Elemente, die den Knoten und Kanten der Karte entsprechen. Jedes „glyph“-Element hat ein Attribut „class“, das den Typ des entsprechenden Knotens der SBGN-Karte bezeichnet. Zusätzlich hat jedes „glyph“-Element ein Attribut „id“, um innerhalb der Datei von anderen Elementen referenziert werden zu können. Außerdem definieren Elemente vom Typ „glyph“ ein Element „bbox“ (*bounding box*, umfassendes Rechteck). Die Koordinaten beschreiben das kleinste umfassende Rechteck und definieren damit Position und Größe des Knotens (siehe z. B. *Process*-Knoten Zeilen 15–19).

Kompartimente werden in SBGN-ML ebenfalls als „glyph“ beschrieben (Zeilen 4–9). Bestimmte Elemente vom Typ „glyph“ (z. B. „simple chemical“ oder „macromolecule“) verweisen auf ihr umgebendes Kompartiment durch ein „compartmentRef“-Attribut (Zeile 10). *Consumption*- und *Production*-Kanten werden mit *Process*-Knoten an „port“-Elementen verbunden, die eine Position definieren und ebenfalls eine „id“ zur Referenzierung besitzen (Zeilen 17–18). Ein „glyph“-Element der Klasse „simple chemical“ wurde für ATP definiert (Zeilen 10–14).

Es hat ein „label“-Element, das einen Namen deklariert, der im Zentrum des Knotens dargestellt werden soll, wenn nicht anders angegeben (Zeile 11). Knoten, die mehrfach auf einer Karte auftreten, müssen einen *Clone Marker* tragen. Dies wird durch das „clone“-Element beschrieben (Zeile 12).

Ähnlich den „glyph“-Elementen definiert auch jedes „arc“-Element ein Attribut „class“ und ein Attribut „id“. Zusätzlich wird noch ein Attribut „source“ und ein Attribut „target“ deklariert. Diese beiden Attribute referenzieren ein „glyph“- oder „port“-Element, an dem die Kante startet und ein „glyph“- oder „port“-Element, an dem die Kante endet. Im Beispiel ist dies für eine *Production*-Kante gezeigt (Zeilen 73–77). „Arc“-Elemente müssen außerdem einen Start- und einen Endpunkt festlegen. Das ist notwendig, da Start- und Endpunkt nicht immer direkt aus dem umfassenden Rechteck eines „glyph“-Elementes abgeleitet werden können und z. B. zusätzliche Berechnungen erfordern (siehe Abbildung 7.2). Entsprechend den Anforderungen an das SBGN-ML Format soll das aber vermieden werden (siehe oben). Die Definition eines „arc“-Elementes kann zusätzliche Positionen zur Beschreibung des Pfades einer Kante enthalten (Zeile 75).

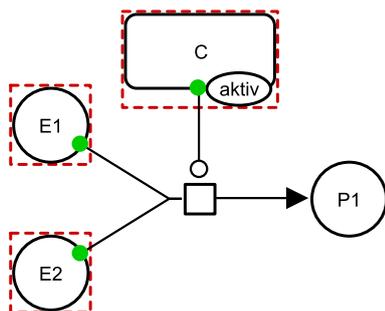


Abbildung 7.2: Start- und Endpunkte für Kanten können nicht immer direkt aus dem umfassenden Rechteck eines Knotens abgeleitet werden und müssen explizit festgelegt werden: Die Startpunkte (grüner Punkt) der beiden *Consumption*-Kanten und der *Catalysis*-Kante liegen nicht auf dem umfassenden Rechteck (rot gestrichelt) der *Simple Chemical*-Knoten bzw. des *Macromolecule*-Knotens.

Eine Übersicht über weitere SBGN-ML Elemente ist in Anhang E als Ergänzung zum Beispiel in Abbildung 7.1 zu finden. In den Tabellen E.1 bis E.4 sind den SBGN-Referenzkarten (siehe Abbildungen 3.2, 3.3 und 3.4) folgend allgemeine SBGN-ML Elemente, die für mehr als eine der drei SBGN-Sprachen genutzt werden, sowie spezifische SBGN-ML Elemente für jede der drei Sprachen gezeigt.

7.2 Die Software-Bibliothek LibSBGN

LibSBGN ist eine Software-Bibliothek zum Importieren und zum Exportieren von SBGN-Karten unter Verwendung des in Abschnitt 7.1 beschriebenen Dateiformates SBGN-ML. Die Bibliothek besteht aus zwei parallelen Implementierungen in C++ und Java und vereinfacht

es Software-Entwicklern, SBGN in ihrer Anwendung zu unterstützen. Die beiden Implementierungen basieren auf demselben Objekt-Modell, so dass auch Implementierungen in weiteren Programmiersprachen denkbar sind.

Das Ziel bei der Entwicklung von LibSBGN war es, die Einbindung der Bibliothek in bestehende Software-Anwendungen zu erleichtern und damit den Aufwand für Entwickler zu verringern. Deshalb

- wurde eine Implementierung vermieden, die spezifisch für eine Software-Anwendung ist. Derartige Implementierungen führen zu Schwierigkeiten bei der Anbindung der Bibliothek in anderen Software-Anwendungen. Die Entwicklung von LibSBGN basierte deshalb von Anfang an auf der Mitarbeit von Entwicklern unterschiedlicher Anwendungen, die eine Unterstützung für SBGN und SBGN-ML anbieten.
- wurde keine gemeinsame Grafik-Implementierung zur Darstellung der SBGN-Karten verwendet. Viele Anwendungen benutzen eine eigene Grafik-Implementierung, die ein wichtiger Bestandteil der Software ist und nicht leicht durch eine gemeinsame Implementierung ersetzt werden kann. Die Bibliothek beschränkt sich deshalb auf den Import und den Export von SBGN-ML Dateien. Software-Anwendungen bilden das Objekt-Modell der Bibliothek auf ihr internes Objekt-Modell ab und benutzen ihre eigene Grafik-Implementierung zur Darstellung.

Der Quelltext wird sowohl für die C++- als auch für die Java-Version der Bibliothek automatisch auf Basis der XML Schema-Definition generiert. Dadurch kann der Entwicklungsaufwand reduziert werden und Fehler müssen nur in der Schema-Definition korrigiert werden. Der erzeugte Java-Code und einige Hilfsklassen bilden eine eigenständige Java-Bibliothek. Der Ansatz, nur eine C++-Bibliothek zu entwickeln und einen Java-*Wrapper* für die Java-Version der Bibliothek zu nutzen, wurde verworfen. Die Erfahrungen im Projekt LibSBML [15] haben gezeigt, dass es einen Bedarf für eine eigenständige Java-Bibliothek gibt. Dies führte zur Entwicklung von JSBML [41] und nun müssen zwei ähnliche Projekte in verschiedenen Programmiersprachen weitergeführt werden. Durch die automatische Erzeugung des Codes für die LibSBGN-Bibliothek kann der Aufwand einer Doppelentwicklung vermieden werden.

7.3 Vergleich der Darstellung in Software-Anwendungen

Für jede der drei SBGN-Sprachen wurden zahlreiche Testbeispiele erzeugt, die alle Aspekte der Syntax abdecken. Zu jedem Testbeispiel gibt es eine Referenzdarstellung als Bilddatei im PNG-Format² und eine entsprechende SBGN-ML Datei. Es wurde ein System für den Vergleich der Darstellung dieser Testbeispiele in verschiedenen Software-Anwendungen umgesetzt. Derzeit sind in dieses System vier Software-Anwendungen integriert: CySBGN [60], PathVisio [143], SBGN-ED [31] (siehe auch Kapitel 8) und SBML Layout [34]. Die Ausgabe des

²<http://www.libpng.org/pub/png/>

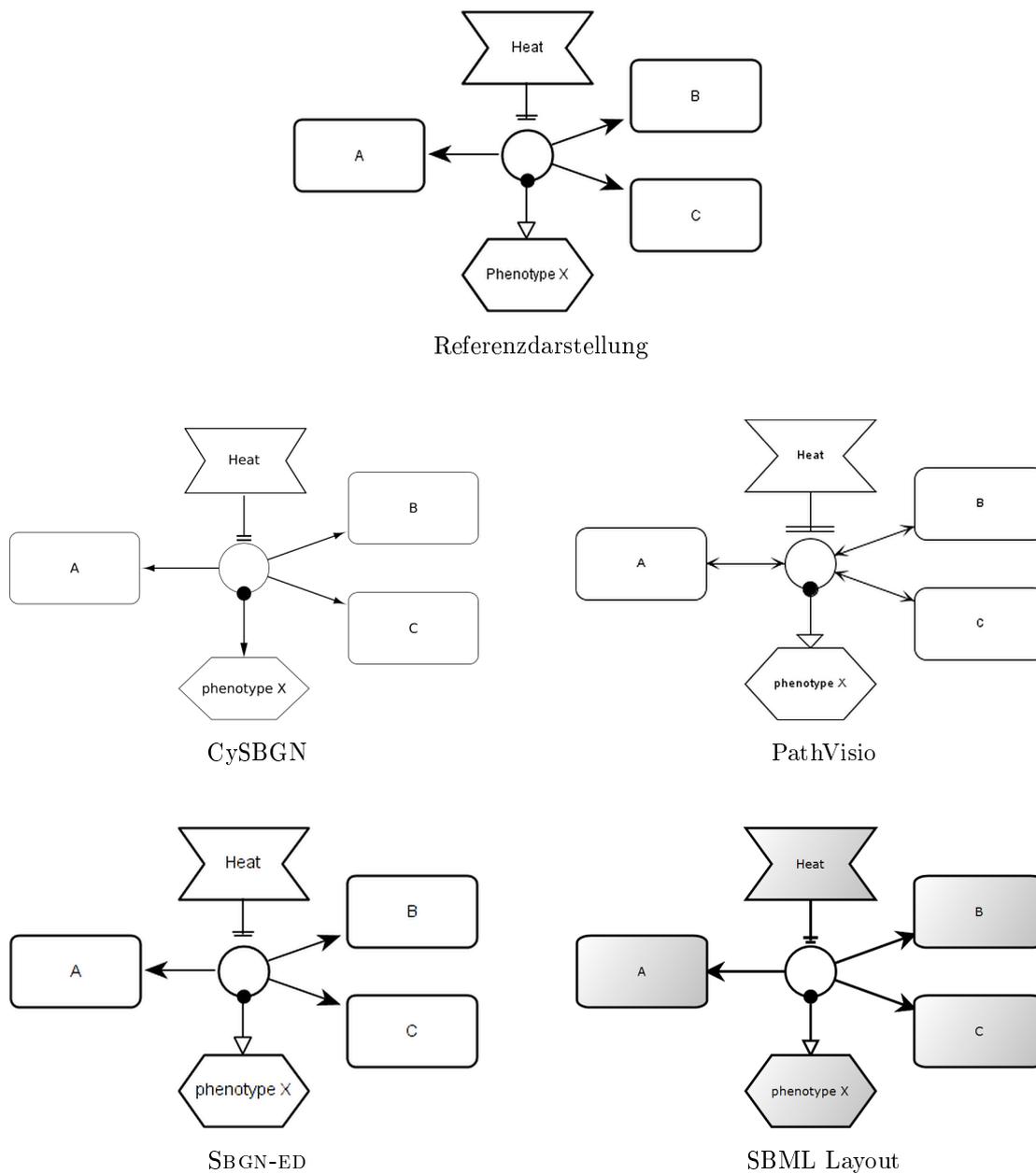


Abbildung 7.3: Vergleich der Darstellung eines Testbeispiels in den vier Software-Anwendungen CySBGN, PathVisio, SBGN-ED und SBML Layout, die Darstellungen wurden übernommen und angepasst³.

Systems ist online verfügbar, es wird für jedes Testbeispiel die Referenzdarstellung zusammen mit den Darstellungen der vier Anwendungen gezeigt³ (siehe Abbildung 7.3).

Dieses System zum Vergleich der Darstellung war während der Entwicklung von LibSBGN sehr hilfreich. Ein Unterschied in der Darstellung eines Testbeispiels in einer der vier Anwendungen und der Referenzdarstellung konnte unterschiedliche Ursachen haben. Oftmals konnte ein Unterschied in der Darstellung auf eine fehlerhafte Implementierung in der entsprechen-

³http://libsbgn.sourceforge.net/render_comparison/

den Anwendung zurückgeführt werden. Eine weitere mögliche Fehlerquelle war eine fehlerhafte Definition in der SBGN-ML Spezifikation, die dann nach einer Diskussion im Projekt zu einer Korrektur der Spezifikation führte. In einigen Fällen konnten Probleme auf Mehrdeutigkeiten in den SBGN-Spezifikationen zurückgeführt werden, die dann vom SBGN-Projekt korrigiert wurden. Auf diese Weise hat LibSBGN auch zu einer Verbesserung der SBGN-Spezifikationen beigetragen.

7.4 Diskussion und Zusammenfassung

Zielstellung war es, einen (elektronischen) Austausch von SBGN-Karten zwischen Software-Anwendungen zu ermöglichen. Dieses Ziel wurde durch das Dateiformat SBGN-ML und die Software-Bibliothek LibSBGN realisiert. Sowohl das Dateiformat als auch die Bibliothek werden bereits in zahlreichen Software-Anwendungen verwendet und beides wird aktiv weiterentwickelt [144].

Ein möglicher Kritikpunkt an SBGN-ML ist die Entwicklung eines weiteren Dateiformates im Umfeld der Systembiologie. Es wurden deshalb verschiedene Möglichkeiten zur elektronischen Ablage von SBGN-Karten untersucht. Dabei wurden z. B. die Ablage in einem Grafik-Format wie SVG⁴, die Ablage als Graph im GraphML-Format [16, 18] und die Ablage in anderen Dateiformaten im Umfeld der Systembiologie wie dem SBML-Format mit Layout-Erweiterung [57] oder dem BioPAX-Format analysiert. Die ersten beiden Ansätze haben Einschränkungen, da sie vollständig unabhängig von SBGN entwickelt wurden. Um z. B. semantische Informationen einer SBGN-Karte in SVG ablegen zu können, müsste die SVG Dokumenttypdefinition⁵ (*Document Type Definition*) erweitert werden, da SVG diese Informationen nicht nativ unterstützt. Ähnlich müsste die GraphML XML Schema-Definition um eine eigene XML Schema-Definition erweitert werden, um komplexe Informationen, wie z. B. *Auxiliary Units* von *Entity Pool Node*-Knoten, im GraphML-Format ablegen zu können [17]. SBML deckt den Bereich der rechnergestützten Modellierung in der Systembiologie ab und BioPAX ist ein Austauschformat für biologische Reaktionswege. SBML und BioPAX hätten möglicherweise hinsichtlich SBGN-Konzepten erweitert werden können, aber es ist nicht immer eine direkte Abbildung von SBGN- auf SBML- oder BioPAX-Elemente möglich, ein Beispiel sind die *Logical Operator*-Knoten. Das SBML-Format mit Layout-Erweiterung z. B. unterstützt nur die semantischen Informationen, die für das SBML-Format definiert wurden, es unterstützt aber keine zusätzlichen semantischen Informationen [57]. Deshalb war die Entwicklung eines neuen Formates notwendig, um alle Eigenschaften von SBGN-Karten unterstützen zu können. SBGN-ML fokussiert dabei auf den Bereich der Visualisierung der SBGN-Konzepte. Dies unterscheidet es besonders von den beiden zuletzt genannten Formaten. Als weiterer Ansatz soll noch die Biological Connection Markup Language (BCML) [13] erwähnt werden. Sie ermöglicht es, die Topologie einer SBGN *Process Description*-Karte abzulegen. Eine Rekonstruktion

⁴<http://www.w3.org/TR/SVG11/>

⁵<http://www.w3.org/TR/SVG11/svgtd.html>

der Darstellung der Karte wird aber nicht unterstützt, die Größe und die Position der Knoten sowie die Pfade der Kanten können nicht mit abgelegt werden.

SBGN-ML wird offiziell vom SBGN-Projekt als Referenzimplementierung unterstützt und stellt die einfachste Möglichkeit dar, um SBGN-Karten zwischen Anwendungen auszutauschen.

Das Dateiformat wird durch die XML Schema-Definition beschrieben. Eine Dokumentation erläutert den Inhalt der XML Schema-Definition und erklärt zusätzliche Regeln und Konventionen (z. B. das verwendete Koordinatensystem). Diese Ressourcen bilden die SBGN-ML Spezifikation. Die Bibliothek (in C++ und Java) und das Dateiformat wurden unter einer Dual-Lizenz veröffentlicht (Lesser General Public Licence Version 2.1 oder höher und Apache Version 2.0) und sind frei verfügbar.

SBGN-ML wird erweitert werden, um die Definition von detaillierten Grafikattributen für die Elemente einer SBGN-Karte zu ermöglichen. LibSBGN ist in erster Linie für den Import und Export von SBGN-Karten gedacht. Weitere Funktionalitäten (z. B. Import oder Export anderer Formate) werden derzeit nicht unterstützt. Durch das genutzte Objekt-Modell ist aber neben C++ und Java eine Erweiterung auf andere Programmiersprachen denkbar.

Kapitel 8

Implementierung von SBGN-ED

Für die Unterstützung von SBGN ist neben der Entwicklung auch die Implementierung von Methoden notwendig. In diesem Kapitel soll deshalb die Software SBGN-ED¹ [31] vorgestellt werden, die eine Implementierung der in den Kapiteln 4 bis 6 beschriebenen sowie weiterer Methoden für die Arbeit mit SBGN-Karten darstellt.

SBGN-ED ist als Add-on für VANTED² [77, 125] implementiert. VANTED wird am Leibniz-Institut für Pflanzengenetik und Kulturpflanzenforschung (IPK) Gatersleben entwickelt und ist eine Software für die Visualisierung und Analyse biologischer Netzwerke und zugehöriger experimenteller Daten. Die Software ermöglicht den Import, die Erstellung und die Bearbeitung biologischer Netzwerke. VANTED erlaubt die Integration und Visualisierung experimenteller Datensätze im Kontext zu Grunde liegender Netzwerke sowie die visuelle Exploration von Netzwerken und Daten. Es bietet außerdem verschiedene Möglichkeiten statistischer Datenanalyse und Simulation. Visualisierungen von Netzwerken und Daten können vom Anwender umfangreich angepasst werden. Die beschriebenen Funktionen werden durch VANTED selbst oder durch entsprechende Add-ons bereitgestellt.

Die Software SBGN-ED wird basierend auf VANTED in der Programmiersprache Java entwickelt und ist frei verfügbar. Die Implementierung der aktuellen Version von SBGN-ED orientiert sich immer an der letzten veröffentlichten Version der SBGN-Spezifikationen (derzeit Level 1 Version 1.3 für die *Process Description*-Sprache, Level 1 Version 1.2 für die *Entity Relationship*-Sprache und Level 1 Version 1.0 für die *Activity Flow*-Sprache).

In den folgenden Abschnitten werden die Software-Architektur von SBGN-ED und VANTED, Methoden und Dateiformate für SBGN-Karten sowie genutzte externe Software-Bibliotheken erläutert. Für eine Übersicht über SBGN-kompatible Software-Anwendungen und einen Vergleich dieser Anwendungen hinsichtlich verschiedener Kriterien siehe auch Abschnitt 3.6.

¹www.sbgn-ed.org

²www.vanted.org

8.1 Die Software-Architektur

Wie bereits erwähnt, ist SBGN-ED als VANTED Add-on implementiert. VANTED selbst basiert auf Gravisto (Graph Visualization Toolkit) [8], einem durch Plug-ins erweiterbaren System zur Visualisierung und Bearbeitung von Graphen, das an der Universität Passau entwickelt wurde. Das Gravisto-System besteht aus drei Schichten: (1) „Core“ stellt grundlegende Graph-Datenstrukturen bereit, (2) „Editor“ stellt die Ereignisverwaltung (Event Listener Manager) und die Plug-in Verwaltung (Plug-in Manager) bereit und (3) „Plug-in“ stellt verschiedene Erweiterungen wie z. B. Algorithmen, Komponenten der Benutzeroberfläche (GUI-Komponenten) sowie Methoden für den Datei-Import und -Export (Dateiformate) bereit. VANTED nutzt die Plug-in basierte Erweiterbarkeit des Systems und ergänzt es u. a. um weitere Algorithmen, GUI-Komponenten und Dateiformate. Zusätzlich erweitert VANTED das Gravisto-System um ein Add-on Konzept (Add-on Manager), das in sich geschlossene Erweiterungen ermöglicht, die zeitlich unabhängig von VANTED als Add-ons veröffentlicht werden können. Sowohl Gravisto als auch VANTED verwenden Funktionalitäten externer Bibliotheken. In Abbildung 8.1 ist die Software-Architektur als Überblick dargestellt.

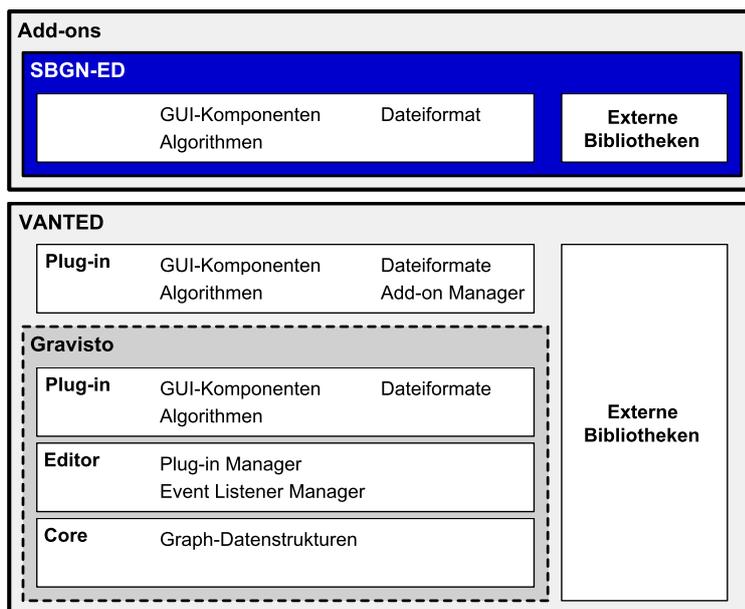


Abbildung 8.1: Überblick über die Software-Architektur: SBGN-ED ist als VANTED Add-on implementiert. VANTED nutzt das Gravisto-System als Basis und erweitert es u. a. um Algorithmen, GUI-Komponenten und Dateiformate. Gravisto (Graph Visualization Toolkit) [8] ist ein durch Plug-ins erweiterbares System zur Visualisierung und Bearbeitung von Graphen.

Gravisto und darauf aufbauend auch VANTED sowie die Add-ons nutzen allgemein das *Observer*-Entwurfsmuster für die Ereignisverwaltung, bei dem ein Objekt eine beliebige Anzahl von abhängigen *Observern* über Änderungen seines Zustands informiert. Im Besonderen wird ein MVC-Design (*Model-View-Controller*) genutzt, welches Datenstrukturen (*Model*), Darstellung (*View*) und Interaktion (*Control*) als entkoppelte Objekte betrachtet [54, 110].

8.2 Methoden für SBGN-Karten

Aus dem Überblick der Software-Anwendungen (siehe auch Abschnitt 3.6), die SBGN unterstützen, ist ersichtlich, dass sich die Anwendungen in zwei Gruppen unterteilen lassen: Betrachter für SBGN-Karten und Editors für SBGN-Karten. Da VANTED umfangreiche Editor-Funktionalitäten bietet und die Grundlage für die Implementierung von SBGN-ED ist, gehört SBGN-ED zur Gruppe der Editors und kann mit den in den Kapiteln 4 bis 7 beschriebenen Methoden und der Editor-Funktionalität zum Erstellen, Bearbeiten, Validieren, Übersetzen, Umwandeln und Austauschen von SBGN-Karten genutzt werden.

SBGN-ED integriert sich im rechten Seitenbereich als Registerkarte mit (untergeordneten) Registerkarten in die Benutzeroberfläche von VANTED. Es gibt eine Registerkarte für jede der drei SBGN-Sprachen und für die SBGN Bricks, eine Registerkarte „Tools“ und eine Registerkarte „Examples“, die Beispiel-Karten für alle drei Sprachen bereitstellt (siehe Abbildung 8.2).

Die Registerkarten für jede drei SBGN-Sprachen enthalten für alle in den Spezifikationen beschriebenen SBGN-Elemente (siehe auch Kapitel 3) Interaktionselemente, die zum Erstellen und Bearbeiten von SBGN-Karten genutzt werden können.

Die Registerkarte „Bricks“ enthält für alle definierten SBGN Bricks (siehe auch Kapitel 6) Interaktionselemente, die zum Erstellen von SBGN-Karten genutzt werden können, wenn die entsprechenden biologischen Prozesse dargestellt werden sollen. Gegebenenfalls können zusätzliche SBGN-Elemente von den drei Registerkarten für die SBGN-Sprachen zur Vervollständigung einer Karte hinzugefügt werden.

In Abbildung 8.3 ist die Registerkarte „Tools“ mit der Validierung („Validate ... map“, siehe unten), der Übersetzung von KEGG-Karten in SBGN-Karten („Translate KEGG to SBGN“, siehe auch Kapitel 4), der Umwandlung von SBGN PD-Karten in SBGN AF-Karten („Translate PD to AF“, siehe auch Kapitel 5) und zusätzlich der Übersetzung von SBML in SBGN („Translate SBML to SBGN“) zu sehen.

Die Validierung ist immer für die aktuell angezeigte SBGN-Karte möglich (hier eine AF-Karte). Nachdem eine Validierung durchgeführt wurde, werden dem Anwender Probleme angezeigt, wenn die Karte nicht der entsprechenden SBGN-Spezifikation folgt, es werden aber keine Vorschläge zur Lösung der Probleme gemacht. Für die Validierung wird ein regelbasiertes System genutzt, das die SBGN-Regeln zur Syntax und Semantik überprüft, die Layout-Richtlinien werden derzeit noch nicht berücksichtigt. In Abbildung 8.4 ist das Beispiel aus Abbildung 8.3 nach der Validierung gezeigt. Zwischen Knoten „p63“ und Knoten „Metastasis Suppressor genes“ wurde eine nicht valide Kante eingefügt, die Kante wird markiert und der Anwender erhält einen entsprechenden Hinweis.

Die Übersetzung von KEGG-Karten basiert auf KGML-Dateien aus der KEGG-Datenbank, die VANTED öffnen und darstellen kann (siehe auch Abschnitt 8.3). In den KGML-Dateien sind für die Edukte und Produkte von Reaktionen keine Namen enthalten sondern *Compound IDs*, die für den Anwender nicht besonders verständlich sind. Die *Compound IDs*

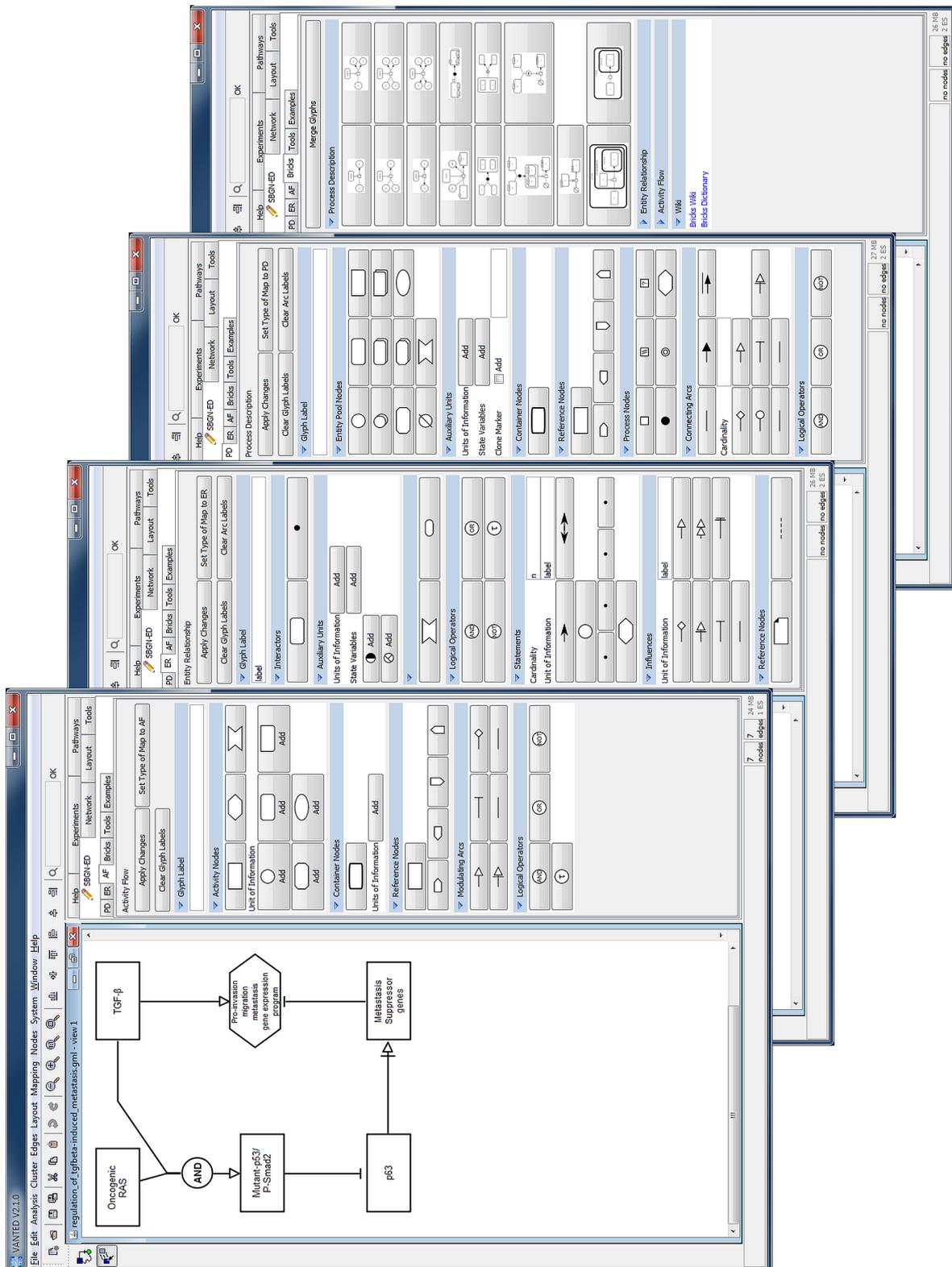


Abbildung 8.2: Bildschirmfotos von SBGN-ED integriert als Add-on in VANTED. Links: Registerkarte für die AF-Sprache und eine AF-Karte, die die Regulierung TGF β -induzierter Metastasierung zeigt [1]; Mitte links: Registerkarte für die ER-Sprache; Mitte rechts: Registerkarte für die PD-Sprache; rechts: Registerkarte für die SBGN Bricks.

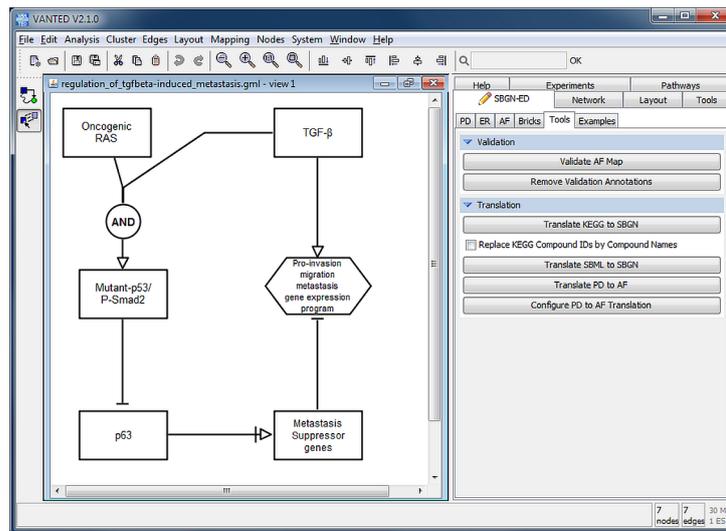


Abbildung 8.3: Bildschirmfoto von SBGN-ED integriert als Add-on in VANTED. Es ist die Registerkarte „Tools“ mit der Validierung, der Übersetzung von KEGG-Karten in SBGN-Karten, der Umwandlung von SBGN PD-Karten in SBGN AF-Karten, der Übersetzung von SBML in SBGN und eine AF-Karte (siehe auch Abbildung 8.2) zu sehen.

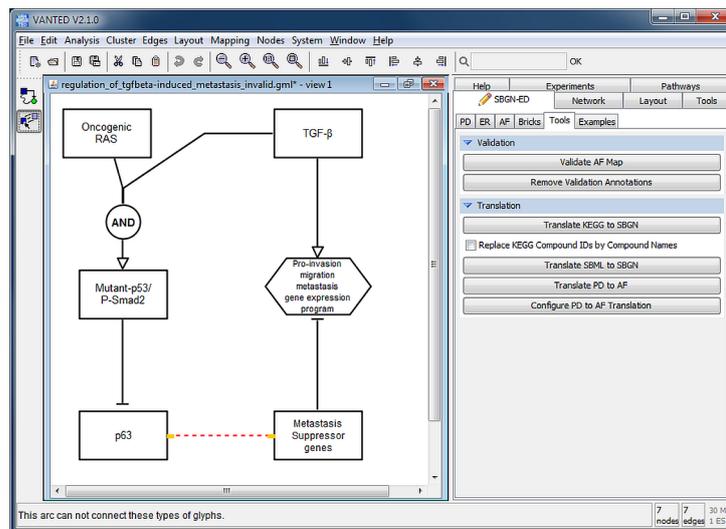


Abbildung 8.4: Beispiel aus Abbildung 8.3 nach der Validierung. Zwischen Knoten „p63“ und Knoten „Metastasis Suppressor genes“ wurde eine nicht valide Kante eingefügt, die Kante wird markiert und der Anwender erhält einen entsprechenden Hinweis (links unten).

können deshalb während der Übersetzung durch *Compound* Namen ersetzt werden („Replace KEGG Compound IDs by Compound Names“, siehe auch Abschnitt 8.4).

Die Umwandlung von PD-Karten in AF-Karten kann durch den Anwender konfiguriert werden („Configure PD to AF Translation“). Dabei kann entweder eines der vier Templates ausgewählt werden oder die anzuwendenden Umwandlungsregeln können über einen Dialog (hier nicht gezeigt) manuell ausgewählt werden, um die gewünschte Darstellung einer AF-Karte zu

erzielen. Die Methode wurde in der Implementierung um Ansätze zur Interaktion ergänzt, die es durch *Brushing & Linking* [139] ermöglichen, zwischen der PD- und der AF-Karte zu navigieren und dabei korrespondierende Elemente in beiden Karten zu markieren.

Zusätzlich steht dem Anwender noch die Übersetzung von SBML in SBGN zur Verfügung. Die dahinter stehende Methodik wird in dieser Arbeit nicht näher betrachtet, da der Ansatz nicht neu ist und andere Software-Anwendungen wie z. B. SBML Layout [34] und Arcadia [145] ähnliche Funktionalität anbieten.

8.3 Dateiformate für SBGN-Karten

Für das Laden und Speichern von SBGN-Karten stehen verschiedene Dateiformate zur Verfügung, die nachfolgend kurz dargestellt werden sollen. Der Austausch von SBGN-Karten mit anderen SBGN-kompatiblen Software-Anwendungen erfolgt über das SBGN-ML Format. Dieses Format wird von SBGN-ED über die Bibliothek LibSBGN bereitgestellt (siehe auch Kapitel 7 und Abschnitt 8.4). Weitere Dateiformate werden von VANTED selbst zur Verfügung gestellt. Da SBGN-ML noch keine Möglichkeit bietet, detaillierte Grafikattribute für die Elemente einer SBGN-Karte wie z. B. Farben, Linienstärken, Schriftgrößen usw. zu speichern, können für die ausschließliche Arbeit in SBGN-ED die beiden Formate GML [63] und GraphML [16, 18] verwendet werden, die diese Möglichkeit bieten. In Kapitel 4 wurde die Übersetzung von KEGG Stoffwechselweg-Karten in SBGN-Karten beschrieben. Diese Übersetzung basiert auf KGML-Dateien, für die VANTED einen Import anbietet.

Die bisher erwähnten Dateiformate ermöglichen es, SBGN-Karten zu erstellen oder zu bearbeiten. Darüber hinaus ist es möglich, SBGN-Karten z. B. für Publikationen in zahlreiche Formate zu exportieren, die VANTED bereitstellt. SBGN-Karten können in verschiedene Bildformate wie JPG³, PNG⁴ und SVG⁵ exportiert werden, außerdem ist der Export als PDF-Datei⁶ und als Webseite [75] möglich.

Zusätzlich zu den beschriebenen Dateiformaten bietet VANTED auch einen Zugriff auf die MetaCrop-Datenbank⁷ [61, 129], die u. a. pflanzenspezifische Stoffwechselweg-Karten in der SBGN *Process Description*-Sprache als GML-Dateien bereitstellt. Der Zugriff ermöglicht es, die GML-Dateien direkt zu öffnen und in SBGN-ED zu verwenden (siehe auch Abschnitt 9.1).

³<http://www.jpeg.org/>

⁴<http://www.libpng.org/pub/png/>

⁵<http://www.w3.org/TR/SVG11/>

⁶http://www.adobe.com/devnet/pdf/pdf_reference.html

⁷<http://metacrop.ipk-gatersleben.de>

8.4 Externe Software-Bibliotheken

In diesem Abschnitt soll ein Überblick über Software-Bibliotheken gegeben werden, die über die von VANTED bereitgestellten Bibliotheken hinaus für die Implementierung von SBGN-ED verwendet wurden.

Wie in Abschnitt 8.3 beschrieben, wird die Bibliothek LibSBGN (siehe auch Kapitel 7) für den Austausch von SBGN-Karten im SBGN-ML Format mit anderen SBGN-kompatiblen Software-Anwendungen genutzt.

Die Übersetzung von KEGG Stoffwechselweg-Karten in SBGN-Karten (siehe auch Kapitel 4) wurde unter Verwendung mehrerer Bibliotheken implementiert. Wie in Kapitel 4 dargelegt, basiert die entwickelte Layout-Methode auf verschiedenen Algorithmen. Diese Algorithmen sind in der Bibliothek Adaptagrams⁸ implementiert, die u. a. Algorithmen und Quellcode zum Zeichnen von Graphen und für das Layout von Graphen, Dokumenten und Diagrammen bereitstellt. Die Software-Bibliothek Adaptagrams wird in der Programmiersprache C++ entwickelt und SBGN-ED wie schon erläutert in der Programmiersprache Java. Um die Bibliothek dennoch nutzen zu können, wurde SWIG⁹ [11] verwendet. SWIG ist ein Software-Entwicklungswerkzeug, mit dem Quellcode generiert werden kann, der eine Nutzung von C- und C++-Quellcode in anderen Programmiersprachen wie z. B. Java ermöglicht.

Für die Übersetzung von KEGG in SBGN werden die von der KEGG-Datenbank bereitgestellten KGML-Dateien genutzt. Diese Dateien enthalten für die Edukte und Produkte einer Reaktion keine Namen sondern IDs (*Compound IDs*). Die IDs sind für einen Anwender nicht direkt verständlich. Die KEGG-Datenbank stellt jedoch eine REST API¹⁰ zur Verfügung, mit der ein RESTful Webservice genutzt werden kann, der es u. a. ermöglicht IDs durch Namen zu ersetzen. Das Architekturprinzip REST [47, 48, 49] ist ein abstraktes Modell einer netzwerkbasierter Architektur, das Prinzipien und Bedingungen zur Einhaltung dieser Prinzipien beschreibt. Als Beispiel für eine REST-Architektur kann HTTP genannt werden. Ein Webservice, der den Prinzipien folgt und die Bedingungen erfüllt, wird als RESTful Webservice bezeichnet. Das Projekt Jersey¹¹ implementiert und erweitert die Spezifikation einer Java API für einen RESTful Webservice. Entsprechende Bibliotheken des Projektes wurden verwendet, um die KEGG REST API zu nutzen und damit IDs durch Namen ersetzen zu können.

⁸<http://www.adaptagrams.org/>

⁹<http://www.swig.org/>

¹⁰<http://www.kegg.jp/kegg/rest/keggapi.html>

¹¹<https://jersey.java.net/>

8.5 Diskussion und Zusammenfassung

In diesem Kapitel wurde die Software SBGN-ED vorgestellt. In der Software sind die in den Kapiteln 4 bis 6 beschriebenen Methoden implementiert. Zusätzlich stellt SBGN-ED Methoden zum Erstellen, Bearbeiten und Validieren von SBGN-Karten in allen drei SBGN-Sprachen bereit. Der Austausch von SBGN-Karten mit anderer SBGN-kompatibler Software wurde über die Software-Bibliothek LibSBGN (siehe auch Kapitel 7) realisiert.

Da SBGN-ED als VANTED Add-on implementiert ist, können bei der Arbeit mit SBGN-Karten auch alle anderen Funktionen von VANTED wie z. B. die Integration und Visualisierung experimenteller Datensätze oder der Export von SBGN-Karten als Bilder und Webseiten genutzt werden [75]. Beispiele für Anwendungen sind in Kapitel 9 zu finden.

Die Implementierung der aktuellen Version von SBGN-ED orientiert sich immer an der letzten veröffentlichten Version der SBGN-Spezifikationen und versucht, diese vollständig umzusetzen. Dabei gibt es derzeit eine Einschränkung. Die Darstellung eines *Compartment*-Knotens ist in SBGN-ED (bedingt durch VANTED als Basis der Implementierung) auf eine rechteckige oder elliptische Form beschränkt. Die SBGN-Spezifikationen beschreiben ein *Compartment* aber als eine beliebige geschlossene Form.

Der Vergleich SBGN-kompatibler Software-Anwendungen (siehe auch Abschnitt 3.6) zeigt, dass SBGN-ED neben Biographer [96] und PathVisio [143] eine der wenigen Anwendungen ist, die eine Erstellung und Bearbeitung von SBGN-Karten in allen drei SBGN-Sprachen sowie einen Austausch der Karten im SBGN-ML Format (siehe auch Kapitel 7) ermöglichen. SBGN-ED ist auch die einzige Software-Anwendung, die eine Methode für die Übersetzung von KEGG-Karten in SBGN PD-Karten unter Erhalt des Layouts (siehe auch Kapitel 4) und eine Methode für die Umwandlung von SBGN PD-Karten in SBGN AF-Karten (siehe auch Kapitel 5) implementiert sowie eine Unterstützung für die SBGN Bricks (siehe auch Kapitel 6) bietet.

Im SBGN-Wettbewerb, der in den Jahren 2010 und 2011 in der SBGN-Community durchgeführt wurde, belegte SBGN-ED in der Kategorie *Best SBGN software support* jeweils den 1. Platz. Um die Qualität der Software zu erhalten oder zu verbessern, ist eine kontinuierliche Pflege und Weiterentwicklung notwendig. So sind gegebenenfalls nach der Veröffentlichung neuer Versionen der SBGN-Spezifikationen Anpassungen der Software erforderlich. Als mögliche Erweiterungen der Software für eine Verbesserung der Anwenderfreundlichkeit sind zwei Funktionen denkbar: (1) die Validierung einer SBGN-Karte schon während des Erstellens oder Bearbeitens kombiniert mit entsprechenden Hinweisen für den Anwender und (2) eine Funktion zum Abspeichern der verwendeten Grafikattribute für die Elemente einer SBGN-Karte wie z. B. Farben, Linienstärken, Schriftgrößen usw., so dass diese auch auf anderen SBGN-Karten angewendet werden können und nicht ausgehend von den Standardwerten angepasst werden müssen. Die Validierung selbst kann noch um die Layout-Richtlinien erweitert werden, so dass diese während der Validierung neben den Regeln zur Syntax und Semantik mit überprüft werden.

Kapitel 9

Anwendungsbeispiele

In den vorherigen Kapiteln wurden bereits zahlreiche Beispiele gezeigt, die unterstützend zur Beschreibung der verschiedenen Methoden genutzt wurden.

In diesem Kapitel sollen nun drei weitere Anwendungsbeispiele vorgestellt werden, die im Rahmen dieser Arbeit unter Verwendung der bisher vorgestellten Methoden und der zugehörigen Implementierung SBGN-ED entstanden. Zusätzlich werden noch Beispiele aus der Literatur aufgelistet, die unabhängig von dieser Arbeit sind und für die SBGN-ED zur Erstellung und Bearbeitung von SBGN-Karten verwendet wurde.

9.1 Repräsentation von Stoffwechselprozessen: MetaCrop – Stoffwechsel in Pflanzen

MetaCrop ist ein am Leibniz-Institut für Pflanzengenetik und Kulturpflanzenforschung (IPK) Gatersleben gehostetes Repositorium, das manuell kurierte Informationen zum Stoffwechsel in Pflanzen bereitstellt. Die Informationen umfassen dabei Stoffwechselweg-Karten, Reaktionen einschließlich Metabolite und Enzyme, Transportprozesse, Stöchiometrien, Lokalisierungen, Reaktionskinetiken und Literaturreferenzen [61, 129].

Derzeit enthält MetaCrop Informationen zu den sieben Nutzpflanzen Gerste, Weizen, Reis, Mais, Kartoffel, Raps [61] und Zuckerrübe [129] sowie den zwei Modellpflanzen Ackerschmalwand und gestutzter Schneckenklee [129]. Die enthaltenen Informationen wurden durch Kuratoren in manueller Arbeit aus Veröffentlichungen und Datenbanken gesammelt, Informationen aus Datenbanken wurden zusätzlich an Hand relevanter Veröffentlichungen überprüft.

Die MetaCrop-Webseite¹ kann zur Navigation und Exploration der Informationen zum Stoffwechsel in Pflanzen auf unterschiedlichen Detailstufen genutzt werden. Es können außerdem Stoffwechsel-Modelle für die Simulation erstellt werden [61, 129]. Stoffwechsel-Modelle können zur Verwendung in entsprechenden Software-Anwendungen im SBML-Format exportiert werden [129].

¹<http://metacrop.ipk-gatersleben.de>

implementiert sondern Skript-Funktionalität von VANTED genutzt. Das Layout wurde bei Knotenüberlappungen ebenso wie die Größe von *Compartment*-Knoten manuell angepasst.

9.1.2 MetaCrop 2.0

In einer erweiterten Version enthält das MetaCrop-Repositoryum aktuell Informationen zu 62 Stoffwechselwegen mit 566 Reaktionen und 63 Transportprozessen aus über 1800 Veröffentlichungen [129]. Die Darstellung der 24 zusätzlichen Stoffwechselwege sollte ebenfalls in der SBGN *Process Description*-Sprache erfolgen. Dazu wurden die Stoffwechselwege über den SBML-Export von MetaCrop exportiert und über den SBML-Import von VANTED importiert (siehe Abbildung 9.2 links). Anschließend wurde ein initiales kräftebasiertes Layout (*force-directed layout*) mit Hilfe von VANTED-Funktionalität erstellt (siehe Abbildung 9.2 Mitte). Diese Darstellung wurde dann mit SBGN-ED und VANTED in die SBGN *Process Description*-Sprache überführt (siehe Abbildung 9.2 rechts).

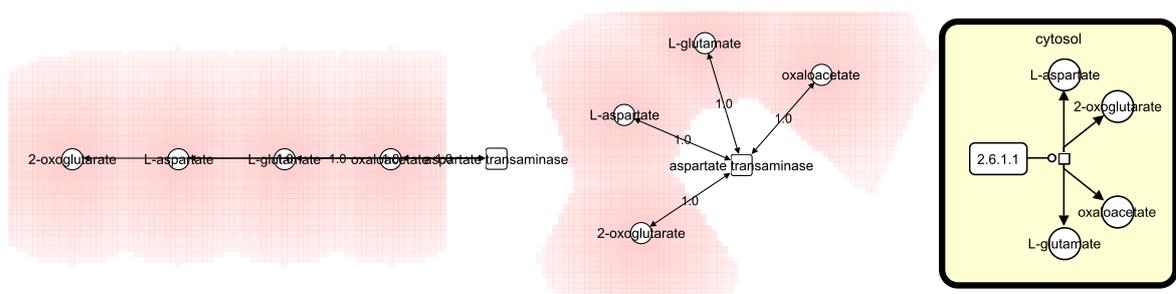


Abbildung 9.2: Aspartat-Abbau des MetaCrop-Repositoryums. Links: Darstellung nach SBML-Import in VANTED, Mitte: Darstellung mit initialem kräftebasierten Layout (*force-directed layout*) in VANTED, rechts: standardisierte Darstellung in der SBGN *Process Description*-Sprache.

In Absprache mit den MetaCrop-Kuratoren und der ersten Version von MetaCrop folgend waren wiederum verschiedene Anpassungen und Erweiterungen der Darstellung notwendig (siehe auch Abbildung 9.2), die im Wesentlichen mit den in Abschnitt 9.1.1 beschriebenen übereinstimmen. Es wurden zusätzlich die Farben der *Compartment*-Knoten auf die in MetaCrop schon verwendeten Farben angepasst und die Namen von Reaktionen entfernt. Für den Prozess wurde wieder Skript-Funktionalität von VANTED genutzt. Das Layout der Karten wurde auf Grund der Größe der Karten (19 Karten enthalten 10 oder weniger Reaktionen und fünf Karten enthalten zwölf bis 21 Reaktionen) nach Diskussion mit den Kuratoren manuell durchgeführt.

9.1.3 Zentralstoffwechsel in Pflanzen

Auf der Basis der einzelnen Stoffwechselweg-Karten der ersten MetaCrop-Version wurde eine Gesamtkarte zusammengestellt, die wichtige Stoffwechselwege des Zentralstoffwechsels in

Pflanzen zeigt (siehe Abbildung 9.3) [30]. Die verschiedenen Stoffwechselwege sind wenn möglich über Metabolite miteinander verknüpft und werden durch unterschiedliche Farben repräsentiert. Die Darstellung ist bis auf die Verwendung des *Annotation*-Knotens zur Benennung der Stoffwechselwege SBGN-konform, der *Annotation*-Knoten wird in der nächsten Version in die *Process Description*-Spezifikation aufgenommen.

Die in Abbildung 9.3 gezeigte SBGN-Karte kann z. B. für eine übersichtliche Darstellung des Zentralstoffwechsels in Pflanzen genutzt werden und dient als Ausgangspunkt für die Integration experimenteller Daten, die dann für weitere Analysen im Kontext des Zentralstoffwechsels in Pflanzen verwendet werden können. Im SBGN-Wettbewerb 2010 belegte diese Karte in der Kategorie *Best SBGN map: breadth, accuracy, aesthetics* den 1. Platz.

9.2 Repräsentation medizinischen Wissens: Entwicklung einer diabetischen Retinopathie

Der Diabetes mellitus ist eine Stoffwechselstörung, die zur Hyperglykämie (erhöhtem Blutzuckerspiegel) führt. Durch Insulinmangel kann Glucose nicht mehr in die Zellen aufgenommen werden und reichert sich im Blut an. Folgeerkrankungen des Diabetes mellitus sind vor allem Gefäßerkrankungen [140].

Die diabetische Retinopathie ist eine Mikroangiopathie der Netzhaut, Veränderungen kleiner Blutgefäße (Kapillaren) in der Netzhaut des Auges führen zu einer Schädigung der Netzhaut [94, 118, 140]. Diese äußert sich zunächst durch Mikroaneurysmen der Kapillaren, als Folge kann es zu Einblutungen in den Glaskörper kommen [140]. Im weiteren Verlauf der Krankheit können Schädigungen in der Umgebung des Punktes schärfsten Sehens (Makula) auftreten, die zu Flüssigkeitsansammlungen und Ablagerungen (Makulaödem) in der Netzhaut führen [118]. Es kann zusätzlich im Bereich der Netzhaut und des Sehnervenkopfes zur Neubildung von Gefäßen kommen (proliferative diabetische Retinopathie), die zu weiteren Einblutungen in den Glaskörper führen können [94, 118]. Als Folge des Makulaödems und der proliferativen diabetischen Retinopathie tritt eine Verminderung der Sehschärfe [94, 118] ein. Die diabetische Retinopathie ist eine Erkrankung, die bis zur Erblindung führen kann [94, 118, 140].

In Abbildung 9.4 sind in einer Übersichtsdarstellung wichtige Schritte in der Entwicklung der diabetischen Retinopathie gezeigt. Die Entwicklung dieser Darstellung wurde von Bernard de Bono (University College London, UK) koordiniert. Zusammenfassend wird der Einfluss überschüssiger Glucose im Blut auf die Durchlässigkeit der Kapillarwände und den Blutfluss gezeigt, in dessen Folge es in der Makula zu einer verminderten Sauerstoffversorgung, zu Ablagerungen und zur Bildung eines Ödems kommt.

Für die Darstellung wurde von den Autoren zunächst eine eigene Notation gewählt. Nach der Veröffentlichung von SBGN entwickelte Bernard de Bono die Idee, wichtige Schritte in der Entwicklung der diabetischen Retinopathie in SBGN darzustellen. Im Rahmen der vorliegenden Arbeit entstand mit SBGN-ED, unterstützt durch zahlreiche Diskussionen mit Bernard

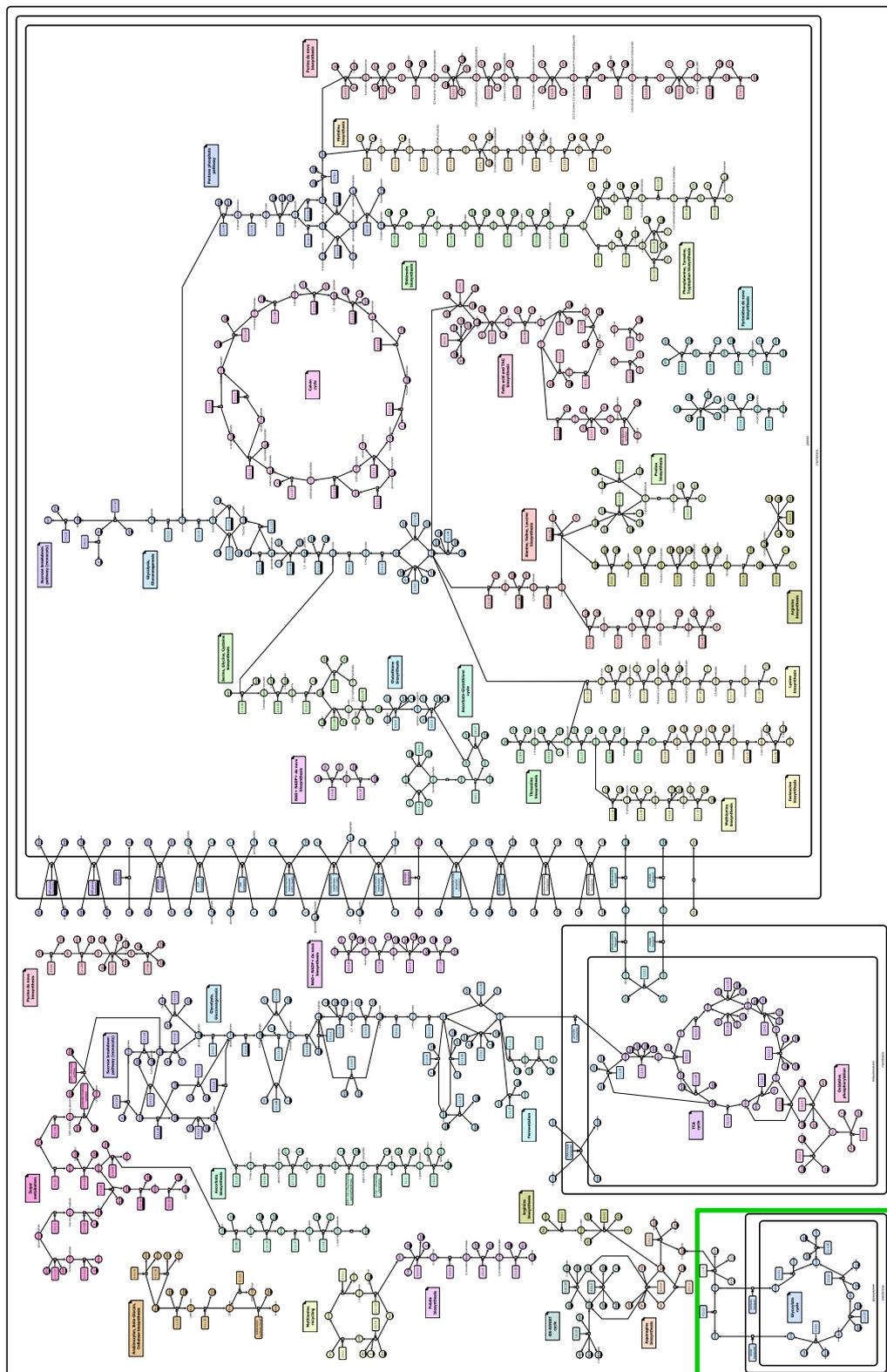


Abbildung 9.3: Zentralstoffwechsel in Pflanzen in einer Übersicht [30]. Die SBGN PD-Karte wurde aus einzelnen Stoffwechselweg-Karten der ersten MetaCrop-Version zusammengestellt. Verschiedene Stoffwechselwege werden durch unterschiedliche Farben repräsentiert, der grün umrandete Bereich zeigt den Glyoxylatzzyklus aus Abbildung 9.1.

de Bono und basierend auf Abbildung 9.4, eine Darstellung in der SBGN *Entity Relationship*-Sprache (siehe Abbildung 9.5) [33]. Der Anwendungsbereich von SBGN wurde dadurch von der Repräsentation biologischen Wissens auf die Repräsentation medizinischen Wissens erweitert.

Um bestimmte Aspekte der Originaldarstellung abbilden zu können, wurden in der SBGN-Darstellung weitreichend Farben für Knoten und den Hintergrund der Karte genutzt. Da SBGN ER nur einen generischen *Entity*-Knoten bereitstellt, wurden vier Farben für die Repräsentation unterschiedlicher *Entity*-Knoten genutzt: blau für Moleküle und Gewebe, pink für Kompartimente, orange für Prozesse und weiß für Phänotypen. Hintergrundfarben wurden zusammen mit expliziten *Entity*-Knoten für Kompartimente genutzt, um eine hierarchische Kompartimentierung darstellen zu können. Diese Form der Darstellung initiierte in der SBGN-Community eine Diskussion, die zu einer Erweiterung der nächsten Version der SBGN *Entity Relationship*-Spezifikation um verschachtelte *Entity*-Knoten führte.

9.3 Repräsentation von Modellen biologischer Prozesse: Das Path2Models-Projekt

Path2Models² ist ein Projekt mit zahlreichen Partnern, Ziel des Projektes ist die systematische Erstellung mathematischer Modelle aus den Stoffwechselweg-Karten der gesamten KEGG-Datenbank und die Ablage der Modelle und deren grafische Repräsentation in der BioModels Database [26].

Traditionell wurden mathematische Modelle auf Basis von Informationen aus der Literatur und auf Basis von experimentellen Daten erstellt. Auf Grund der zunehmenden Größe und Komplexität der Modelle lässt sich dieser Ansatz immer schwieriger umsetzen. Es wurde deshalb damit begonnen, Modelle auf Basis von Stoffwechselweg-Datenbanken zu erstellen. Dabei wird jedoch oftmals von Grund auf begonnen, ein neues Modell zu erstellen, da bereits erstellte Modelle nicht zur Verfügung stehen oder nicht weiter genutzt werden können. Das Path2Models-Projekt versucht deshalb durch die Erstellung mathematischer Modelle in großem Maßstab auf Basis von Stoffwechselwegen der KEGG-Datenbank und die Anwendung standardisierter und unterstützter Dateiformate für den Austausch und die Weiterverwendung der Modelle, den oftmals wiederholten initialen Schritt bei der Erstellung der Modelle zu vereinfachen [20].

Im Path2Models-Projekt wurden drei verschiedene Arten von Modellen erstellt: (1) quantitative, kinetische Modelle von Stoffwechselwegen; (2) qualitative, logische Modelle von Stoffwechsel- und Signalwegen und (3) metabolische Modelle auf Basis des gesamten Genoms. Alle Modelle stehen im SBML-Format zur Verfügung, zusätzlich wurden die kinetischen Modelle als SBGN PD-Karten und die logischen Modelle als SBGN AF-Karten visualisiert. Die bisher im Projekt erstellten mehr als 140.000 Modelle stehen in der BioModels Database in

²<http://www.ebi.ac.uk/biomodels-main/path2models>

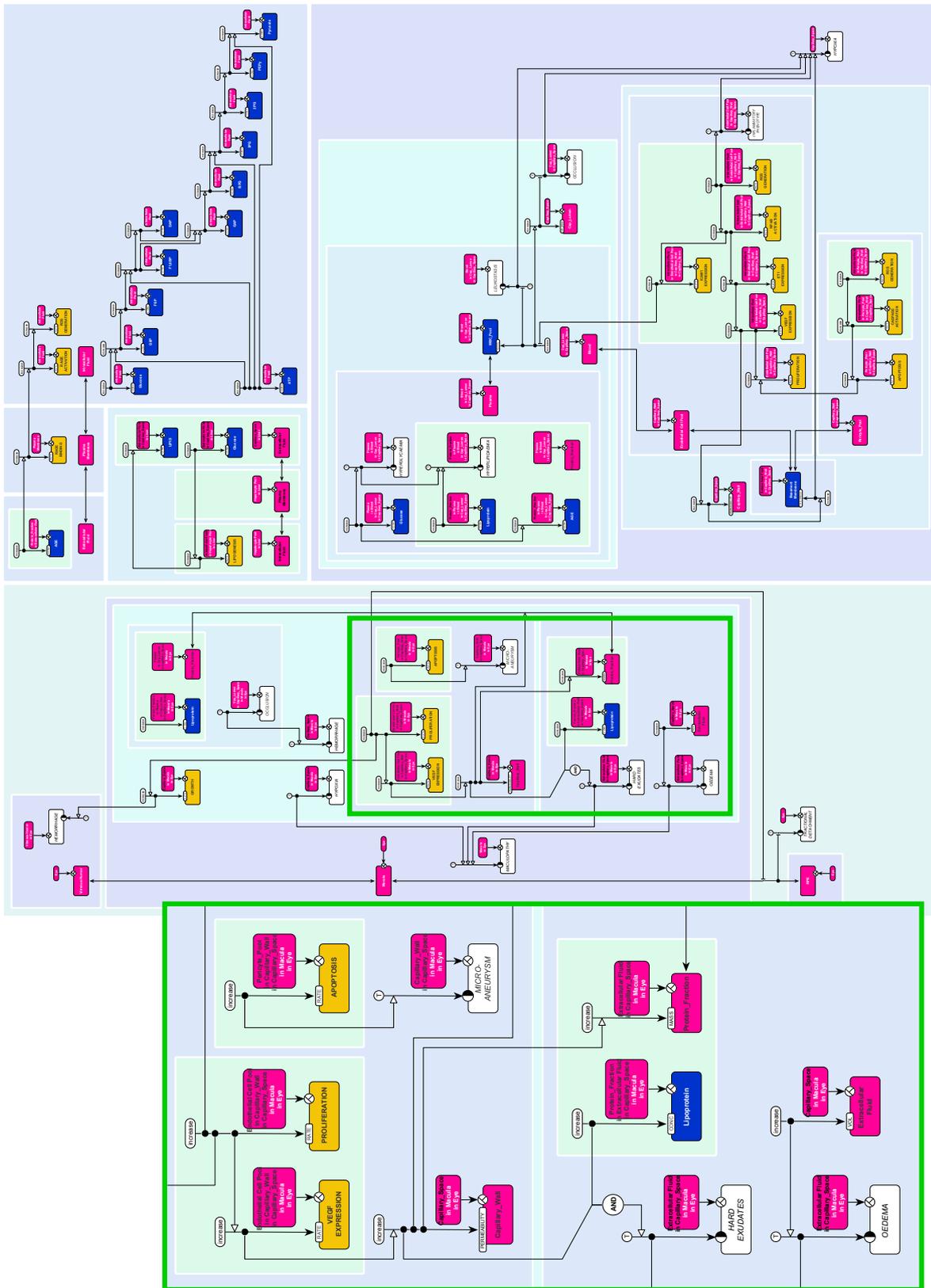


Abbildung 9.5: Darstellung wichtiger Schritte in der Entwicklung der diabetischen Retinopathie in SBGN ER [33] basierend auf der Darstellung in Abbildung 9.4.

einem eigenen Zweig frei zur Verfügung und können als Grundlage für die Entwicklung mathematischer Modelle genutzt werden. Alle in Path2Models enthaltenen Modelle stellen einen Ausgangspunkt für die Modellierung dar, Modelle können für bestimmte biologische Fragestellungen weiter angepasst und durch entsprechende Anfangsbedingungen und Parameter ergänzt werden [20].

Im Rahmen der vorliegenden Arbeit wurden 112.898 kinetische Modelle (erstellt aus 154 Stoffwechselwegen von 1514 Organismen) [20] basierend auf der entsprechenden SBML-Datei als SBGN *Process Description*-Karte visualisiert. In Abbildung 9.6 ist als Beispiel der Zitratzyklus (*Homo sapiens*)³ gezeigt. Die Erstellung der PD-Karten aus den SBML-Dateien und das anschließende automatische Layout (basierend auf der Layout-Methode in Abschnitt 4.2) wurden mit SBGN-ED durchgeführt.

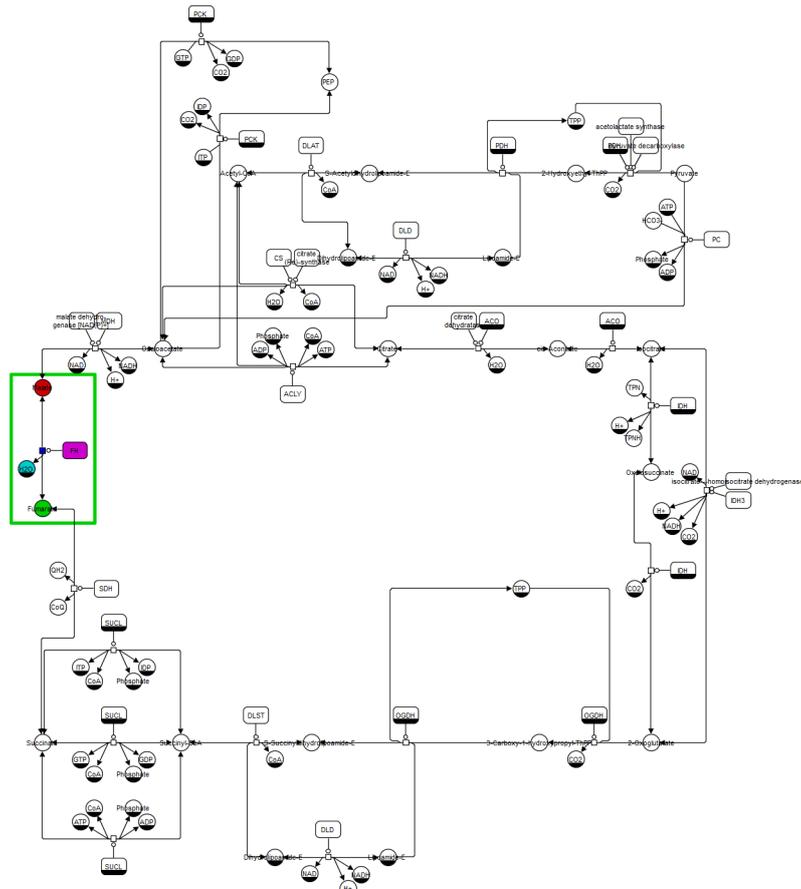
Jedes SBML-Element wurde auf einen entsprechenden SBGN PD-Knoten abgebildet, dazu wurden die SBO-Terme der SBML-Elemente ausgewertet. Für jede Reaktion wurden SBGN PD-Kanten zwischen dem korrespondierenden *Process*-Knoten und allen Reaktionspartnern eingefügt, der Typ der Kanten wurde ebenfalls auf Basis von SBO-Termen in der SBML-Datei bestimmt.

Während der Erstellung der SBML-Modelle aus den von der KEGG-Datenbank bereitgestellten KGML-Dateien wurden die Informationen mit der KEGG-Datenbank verglichen und es wurden gegebenenfalls z. B. Reaktionen oder Reaktionspartner ergänzt. Zusätzlich wurde die SBML Layout-Erweiterung [57] genutzt, um die Positionen der Elemente in der grafischen KEGG-Repräsentation mit in der SBML-Datei ablegen zu können [20]. Diese Positionen wurden dann als initiale Positionen für SBGN-Knoten verwendet.

Simple Chemical-Knoten ohne Positionsinformationen, die mit mehr als einer Kante verbunden waren und *Macromolecule*-Knoten, die mit mehr als einer Kante verbunden waren, wurden vervielfältigt, so dass sie mehrfach in der Karte erscheinen und jeweils nur mit einer Kante verbunden sind. Das Ergebnis dieses ersten Schrittes waren SBGN-konforme Darstellungen mit einem unvollständigen Layout, das in einem zweiten Schritt vervollständigt wurde.

Das automatische Layout wurde basierend auf der Layout-Methode in Abschnitt 4.2 durchgeführt. Diese Methode musste um die Berechnung von initialen Positionsinformationen für Knoten, die nicht im originalen KEGG-Stoffwechselweg enthalten waren, ergänzt werden. Für *Process*-Knoten wurde initiale Positionen abhängig von Reaktionspartnern, die im originalen KEGG-Stoffwechselweg enthalten waren, berechnet. Standen diese Reaktionspartner nicht zur Verfügung, wurden die *Process*-Knoten am oberen Rand der Karte platziert, sonst horizontal oder vertikal ausgerichtet mit den Reaktionspartnern in der Umgebung der Reaktionspartner. *Simple Chemical*-Knoten, die sekundäre Metabolite repräsentieren, wurden gruppiert nach Edukten und Produkten radial um den entsprechenden *Process*-Knoten, der die Reaktion repräsentiert, platziert.

³<http://www.ebi.ac.uk/biomodels-main/BMID00000101169>



```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" xmlns:layout="http://www.sbml.org/sbml/level3/
version1/layout/version1" level="3" version="1" layout:required="false">
<model id="BMID00000101169" name="Citrate cycle" metaid="meta_path_hsa00020"
timeUnits="time" substanceUnits="substance" volumeUnits="volume">
<listOfSpecies>
<species id="Malate" constant="false" initialAmount="1" hasOnlySubstanceUnits="false" name="Malate"
metaid="meta_Malate" boundaryCondition="false" sboTerm="SBO:0000247" compartment="default">
...
</species>
<species id="Fumarate" constant="false" initialAmount="1" hasOnlySubstanceUnits="false" name="Fumarate"
metaid="meta_Fumarate" boundaryCondition="false" sboTerm="SBO:0000247" compartment="default">
...
</species>
<species id="H2O" constant="false" initialAmount="1" hasOnlySubstanceUnits="false" name="H2O"
metaid="meta_H2O" boundaryCondition="false" sboTerm="SBO:0000247" compartment="default">
...
</species>
<species id="FH" constant="false" initialAmount="1" hasOnlySubstanceUnits="false" name="FH"
metaid="meta_FH" boundaryCondition="false" sboTerm="SBO:0000252" compartment="default">
...
</species>
</listOfSpecies>
<listOfReactions>
<reaction id="rnR01082" name="rn:R01082" metaid="meta_rnR01082" reversible="true"
sboTerm="SBO:0000176" fast="false" compartment="default">
<listOfReactants>
<speciesReference id="cpdC00149" constant="false" species="Malate" name="cpd:C00149"
metaid="meta_cpdC00149" sboTerm="SBO:0000010" stoichiometry="1"/>
...
</listOfReactants>
<listOfProducts>
<speciesReference id="cpdC00122_1" constant="false" species="Fumarate" name="cpd:C00122"
metaid="meta_cpdC00122_1" sboTerm="SBO:0000010" stoichiometry="1"/>
<speciesReference id="cpdC00001" constant="false" species="H2O" name="cpd:C00001"
metaid="meta_cpdC00001" sboTerm="SBO:0000010" stoichiometry="1"/>
...
</listOfProducts>
<listOfModifiers>
<modifierSpeciesReference id="mod_rnR01082" species="FH" name="mod_rnR01082"
metaid="meta_mod_rnR01082" sboTerm="SBO:0000460"/>
...
</listOfModifiers>
...
</reaction>
...
</listOfReactions>
...
</model>
</sbml>
    
```

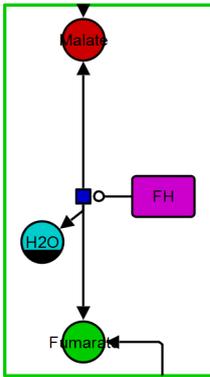


Abbildung 9.6: Zitratzyklus (*Homo sapiens*) aus dem Path2Models-Projekt. Oben: SBGN PD-Karte, unten: Ausschnitt aus dem SBML-Modell (rechts) und Ausschnitt aus der PD-Karte (links), korrespondierende Elemente sind durch gleiche Farben hervorgehoben.

Das Ergebnis der beiden zuvor beschriebenen Schritte waren SBGN PD-Karten mit einem kompakten SBGN-konformen Layout, das dem originalen KEGG-Layout sehr ähnlich ist. Die Karten wurden als SBGN-ML Dateien (siehe auch Kapitel 7) und PNG-Bilddateien⁴ exportiert und in der BioModels Database abgelegt. Es gibt jedoch noch einige offene Probleme, wie das vereinzelt Auftreten langer Knotennamen im Gegensatz zur einheitlichen Größe der Knoten und langer Kanten zwischen Knoten. Die Anzahl langer Kanten könnte durch weitere Vervielfachung von Knoten (der gleiche Knoten, versehen mit einem *Clone Marker*, erscheint an mehreren Positionen in der Karte) verringert werden.

9.4 Weitere Anwendungsbeispiele

Neben den bisher vorgestellten Anwendungsbeispielen, die im Rahmen dieser Arbeit entstanden, lassen sich in der Literatur Beispiele finden, die unabhängig von dieser Arbeit sind und für die SBGN-ED zur Erstellung und Bearbeitung von SBGN-Karten verwendet wurde. Diese weiteren Beispiele sollen hier kurz aufgelistet werden. Sie zeigen, dass die entwickelten Methoden und die Implementierung bereits von weiteren Anwendern genutzt werden.

- JUNKER, Astrid ; HARTMANN, Anja ; SCHREIBER, Falk ; BÄUMLEIN, Helmut: An engineer's view on regulation of seed development [73].

In diesem Artikel wird SBGN genutzt, um wichtige regulatorische Prozesse, die während der Samenentwicklung in der Ackerschmalwand stattfinden, als SBGN-Karten darzustellen. Die SBGN-Karten werden in einem eigenen Internetportal zur Verfügung gestellt.

- JOHNNIE, Joseph G.: A Systems Engineering Framework for Metabolic Engineering Experiments [71].

Diese Arbeit beschreibt ein System für das Metabolic Engineering, das für die Synthese, die Analyse und das Design metabolischer Netzwerke genutzt werden kann und untersucht, wie Abstraktionen aus der Systemtechnik (SysML) zur Ergänzung und Erweiterung von Abstraktionen in der Biologie (SBGN) genutzt werden können.

- JUNKER, Astrid ; JUNKER, Björn H.: Synthetic Gene Networks in Plant Systems [74].

Dieses Buchkapitel gibt einen Überblick über die Nutzung genetischer Netzwerke von Pflanzen in der Synthetischen Biologie und beschreibt im Detail Protokolle für die Anwendung zweier Systeme für die induzierte Genexpression in Kartoffel und Ackerschmalwand.

- MAZEIN, Alexander ; WATTERSON, Steven ; HSIEH, Wei-Yuan ; GRIFFITHS, William J. ; GHAZAL, Peter: A comprehensive machine-readable view of the mammalian cholesterol biosynthesis pathway [109].

Dieser Artikel stellt eine Übersicht über vorhandene Literatur zum Stoffwechselweg der

⁴<http://www.libpng.org/pub/png/>

Cholesterin-Biosynthese bereit. Zusätzlich wurde der Stoffwechselweg der Cholesterin-Biosynthese (Mevalonat-, Kandutch-Russell- und Bloch-Stoffwechselweg) in SBGN visualisiert und steht auch als SBGN-ML Datei zur Verfügung.

9.5 Zusammenfassung

In diesem Kapitel wurden Anwendungsbeispiele dargestellt, die im Rahmen dieser Arbeit unter Verwendung der vorgestellten Methoden und der zugehörigen Implementierung entstanden. Die im MetaCrop-Repositorium enthaltenen Stoffwechselwege wurden als SBGN PD-Karten dargestellt, zusätzlich wurde basierend auf diesen Karten der Zentralstoffwechsel in Pflanzen in einer Übersichtskarte zusammengefasst. Aus dem medizinischen Umfeld wurde die Darstellung wichtiger Schritte bei der Entwicklung einer diabetischen Retinopathie als SBGN ER-Karte umgesetzt. Im Path2Models-Projekt wurden mehr als 112.000 kinetische Modelle basierend auf entsprechenden SBML-Dateien als SBGN PD-Karten unter Anwendung einer automatischen Layout-Methode visualisiert.

Zusätzlich wurden noch weitere Beispiele aus der Literatur aufgelistet, die Methoden dieser Arbeit und die zugehörige Implementierung nutzen, aber unabhängig von dieser Arbeit entstanden.

Kapitel 10

Zusammenfassung und Ausblick

10.1 Zusammenfassung

Visualisierungen sind in der Biologie wichtig für das Verständnis und für die Analyse von Daten. Dabei werden oft Darstellungen biologischer Netzwerke genutzt, die in Form von Graphen mit biologischen Entitäten als Knoten und den Wechselwirkungen zwischen Entitäten als Kanten gezeigt werden. Zusätzlich ist eine Visualisierung experimenteller Daten im Kontext der biologischen Netzwerke möglich.

Eine standardisierte Notation für die Darstellung biologischer Netzwerke war bisher nicht verfügbar, ein Mangel an Standardisierungsansätzen besonders im Umfeld der Systembiologie wurde jedoch als Problem identifiziert [58]. Die Systems Biology Graphical Notation mit den drei Sprachen *Process Description*, *Entity Relationship* und *Activity Flow* schlägt eine standardisierte Darstellung biologischer Prozesse und Netzwerke unter Beachtung unterschiedlicher Aspekte und in verschiedenen Sichten vor.

Das Ziel der vorliegenden Arbeit war die Entwicklung von Methoden und Algorithmen für die Systems Biology Graphical Notation, die für eine standardisierte Darstellung biologischen Wissens genutzt werden und zu einem Erkenntnisgewinn in der Biologie beitragen.

Für ein besseres Verständnis der Arbeit wurden in Kapitel 2 und 3 zunächst wichtige Grundlagen aus der Biologie, Mathematik und Informatik zusammengefasst und es wurde eine Einführung in SBGN gegeben.

In Kapitel 4 wurde eine Methode vorgestellt, die metabolische Stoffwechselweg-Karten aus der KEGG-Datenbank in SBGN PD-Karten übersetzt und dabei zusätzlich einen Layout-Algorithmus anwendet, der das originale, manuell erstellte Layout erhält und den SBGN Layout-Richtlinien folgt. Der Erhalt des manuellen Layouts ist sehr wichtig, da die Qualität eines manuellen Layouts höher als die Qualität eines automatischen Layouts ist [58]. Die entwickelte Methodik zeigt eine Möglichkeit, wie bereits vorhandenes Wissen in eine Darstellung in SBGN überführt werden kann und demonstriert, wie die SBGN Layout-Richtlinien in einem Algorithmus umgesetzt werden können.

Eine Methode für die Umwandlung von *Process Description*-Karten in *Activity Flow*-Karten wurde in Kapitel 5 präsentiert. Es wurden Regeln für die Umwandlung definiert und basierend auf den Regeln vier Templates spezifiziert, die die Anwendung vereinfachen. Die Methode beschreibt einen ersten Ansatz, um in SBGN automatisiert verschiedene Sichten auf ein biologisches System zu erhalten.

Die SBGN Bricks wurden in Kapitel 6 beschrieben, sie ermöglichen eine vereinfachte Anwendung von SBGN zur Darstellung biologischen Wissens. SBGN selbst definiert die SBGN-Elemente, es definiert aber nicht im Detail, wie diese verschiedene biologische Prozesse darstellen können. SBGN Bricks sind Templates in allen drei SBGN-Sprachen zur Repräsentation einer Vielzahl grundlegender biologischer Prozesse, die zu verschiedenen Arten biologischer Netzwerke kombiniert werden können. Eine erste Version eines Bricks-Wörterbuchs ist online für die Nutzung im Allgemeinen oder auch in anderen Software-Anwendungen verfügbar.

Ein Community-basierter Ansatz für den elektronischen Austausch von SBGN-Karten zwischen Software-Anwendungen und zur Ablage von SBGN-Karten in Repositorien wurde in Kapitel 7 erläutert. Das Dateiformat SBGN-ML und die Software-Bibliothek LibSBGN bieten eine einfache Möglichkeit für den Austausch und die Ablage von SBGN-Karten und werden als Referenzimplementierung vom SBGN-Projekt unterstützt.

In Kapitel 8 wurden einige Aspekte der Implementierung von SBGN-ED als VANTED Add-on vorgestellt. Die Software SBGN-ED implementiert die in dieser Arbeit beschriebenen Methoden und Konzepte sowie weitere Methoden zum Erstellen, Bearbeiten und Validieren von SBGN-Karten. Neben Biographer [96] und PathVisio [143] ist SBGN-ED eine der wenigen Anwendungen, die alle drei SBGN-Sprachen und SBGN-ML unterstützen. SBGN-ED ist die einzige Software-Anwendung, die zusätzlich Methoden für die Übersetzung vorhandener Darstellungen in eine Darstellung in SBGN unter Erhalt des Layouts und für die Umwandlung zwischen SBGN-Sprachen sowie eine Unterstützung für die SBGN Bricks anbietet. Da es als VANTED Add-on implementiert ist, können weitere Methoden von VANTED z. B. zur Exploration sowie zur Integration und Analyse experimenteller Daten genutzt werden.

Kapitel 9 stellte Anwendungsbeispiele vor, die im Rahmen der vorliegenden Arbeit erarbeitet wurden. Es wurden Beispiele zur Repräsentation von Stoffwechselprozessen (MetaCrop-Repositorium), zur Repräsentation medizinischen Wissens (Entwicklung einer diabetischen Retinopathie) und zur Repräsentation von Modellen biologischer Prozesse (Path2Models-Projekt) gezeigt, außerdem wurde ein Überblick über weitere Anwendungsbeispiele gegeben.

10.2 Ausblick

Die in der vorliegenden Dissertation vorgestellte Methoden und Algorithmen für die Systems Biology Graphical Notation lassen sich für weiterführende Fragestellungen anpassen bzw. erweitern.

Die Methode zur Übersetzung metabolischer Stoffwechselweg-Karten aus der KEGG-Datenbank in SBGN PD-Karten unter Erhalt des originalen Layouts kann als Ausgangspunkt

zur Entwicklung weiterer Übersetzungsmethoden für Stoffwechselweg-Karten anderer Datenbanken wie z. B. BioCyc/MetaCyc [23] und WikiPathways [82] genutzt werden.

Der Layout-Algorithmus, der während der Übersetzung von Stoffwechselweg-Karten der KEGG-Datenbank genutzt wurde, kann als Basis für die Entwicklung allgemeiner Layout-Algorithmen für SBGN-Karten verwendet werden, die den SBGN Layout-Richtlinien folgen. Weitere Anwendungsfälle für den Algorithmus können auch SBGN-Karten sein, die aus dem SBML- oder dem BioPAX-Format in SBGN übersetzt werden.

Eine Methodik für die flexible Umwandlung von SBGN PD-Karten in SBGN AF-Karten ist ein erster Ansatz für die Umwandlung von SBGN-Karten in die verschiedenen SBGN-Sprachen und ermöglicht neue Anwendungen [126]. Darauf basierend können weitere Methoden entwickelt werden, die eine Umwandlung zwischen anderen SBGN-Sprachen bieten, denkbar sind z. B. Umwandlungen von der *Entity Relationship*-Sprache in die *Activity Flow*-Sprache oder von der *Process Description*-Sprache in die *Entity Relationship*-Sprache, um weitere Sichten auf ein biologisches System zu erhalten.

Die SBGN Bricks sind in einer ersten Version als Bricks-Wörterbuch online verfügbar und können erweitert oder angepasst werden. Die Bricks stehen dort zusätzlich zum Download im SBGN-ML Format bereit und können neben SBGN-ED in weiteren Software-Anwendungen, die SBGN und SBGN-ML unterstützen, zum vereinfachten Erstellen von SBGN-Karten verwendet werden.

Das Dateiformat SBGN-ML bietet eine einfache Möglichkeit zum elektronischen Austausch von SBGN-Karten. Das Format beschränkt sich derzeit auf die Ablage der Geometrie und der Topologie einer SBGN-Karte und wird durch das LibSBGN-Projekt um die Ablage detaillierter Grafikattribute für die Elemente einer SBGN-Karte ergänzt werden. Die zugehörige Software-Bibliothek LibSBGN ist momentan für die Programmiersprachen C++ und Java verfügbar, durch das genutzte Objekt-Modell ist eine Erweiterung auf andere Sprachen möglich.

Neben möglichen Anpassungen und Erweiterungen der Methoden und Algorithmen ist eine kontinuierliche Pflege und Weiterentwicklung der Implementierung wichtig. Gegebenenfalls sind nach der Veröffentlichung neuer Versionen der SBGN-Spezifikationen Anpassungen und/oder Erweiterungen von SBGN-ED erforderlich. Zusätzlich sind weitere Funktionen für eine Verbesserung der Anwenderfreundlichkeit denkbar.

Literaturverzeichnis

- [1] ADORNO, Maddalena ; CORDENONSI, Michelangelo ; MONTAGNER, Marco ; DUPONT, Sirio ; WONG, Christine ; HANN, Byron ; SOLARI, Aldo ; BOBISSE, Sara ; RONDINA, Maria B. ; GUZZARDO, Vincenza u. a.: A Mutant-p53/Smad Complex Opposes p63 to Empower TGF β -Induced Metastasis. In: *Cell* 137 (2009), Nr. 1, S. 87–98 (S. 88)
- [2] ALBERGHINA, Lilia (Hrsg.) ; WESTERHOFF, Hans V. (Hrsg.): *Systems Biology: Definitions and Perspectives*. Berlin Heidelberg, Germany : Springer, 2005 (Topics in Current Genetics) (S. 11)
- [3] ALBERTS, Bruce ; BRAY, Dennis ; HOPKIN, Karen ; JOHNSON, Alexander ; LEWIS, Julian ; RAFF, Martin ; ROBERTS, Keith ; WALTER, Peter: *Lehrbuch der Molekularen Zellbiologie*. Weinheim, Deutschland : Wiley-VCH Verlag GmbH, 1999 (S. 7, 9 und 10)
- [4] ALBRECHT, Mario ; KERREN, Andreas ; KLEIN, Karsten ; KOHLBACHER, Oliver ; MUTZEL, Petra ; PAUL, Wolfgang ; SCHREIBER, Falk ; WYBROW, Michael: On Open Problems in Biological Network Visualization. In: EPPSTEIN, David (Hrsg.) ; GANSNER, Emden R. (Hrsg.): *Graph Drawing 2009* Bd. 5849. Berlin Heidelberg, Germany : Springer, 2010 (Lecture Notes in Computer Science), S. 256–267 (S. 19)
- [5] AOKI-KINOSHITA, Kiyoko F. ; KANEHISA, Minoru: Gene annotation and pathway mapping in KEGG. In: BERGMAN, Nicholas H. (Hrsg.): *Comparative Genomics, Volume 2*. Humana Press, 2007 (Methods in Molecular Biology 396), Kapitel 6, S. 71–91 (S. 40 und 43)
- [6] ASHBURNER, Michael ; BALL, Catherine A. ; BLAKE, Judith A. ; BOTSTEIN, David ; BUTLER, Heather ; CHERRY, J. Michael ; DAVIS, Allan P. ; DOLINSKI, Kara ; DWIGHT, Selina S. ; EPPIG, Janan T. u. a.: Gene Ontology: tool for the unification of biology. In: *Nature Genetics* 25 (2000), Nr. 1, S. 25–29 (S. 71)
- [7] BABUR, Özgün ; DOGRUSÖZ, Ugur ; DEMIR, Emek ; SANDER, Chris: ChiBE: interactive visualization and manipulation of BioPAX pathway models. In: *Bioinformatics* 26 (2010), Nr. 3, S. 429–431 (S. 36)
- [8] BACHMAIER, Christian ; BRANDENBURG, Franz J. ; FORSTER, Michael ; HOLLEIS, Paul ; RAITNER, Marcus: Gravisto: Graph Visualization Toolkit. In: PACH, János

- (Hrsg.): *Graph Drawing 2004* Bd. 3383. Berlin Heidelberg, Germany : Springer, 2005 (Lecture Notes in Computer Science), S. 502–503 (S. 86)
- [9] BACHMAIER, Christian ; BRANDES, Ulrik ; SCHREIBER, Falk: Biological Networks. In: TAMASSIA, Roberto (Hrsg.): *Handbook of Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013 (Discrete Mathematics and Its Applications), Kapitel 20, S. 621–651 (S. 17)
- [10] BADER, Gary D. ; CARY, Michael P. ; SANDER, Chris: Pathguide: a Pathway Resource List. In: *Nucleic Acids Research* 34 (2006), Nr. 1, S. D504–D506 (S. 39)
- [11] BEAZLEY, David M.: SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++. In: *Proceedings of the USENIX Fourth Annual Tcl/Tk Workshop*. Berkeley, CA, USA : USENIX Association, 1996, S. 129–139 (S. 91)
- [12] BECKER, Moritz Y. ; ROJAS, Isabel: A graph layout algorithm for drawing metabolic pathways. In: *Bioinformatics* 17 (2001), Nr. 5, S. 461–467 (S. 18)
- [13] BELTRAME, Luca ; CALURA, Enrica ; POPOVICI, Razvan R. ; RIZZETTO, Lisa ; GUEDEZ, Damariz R. ; DONATO, Michele ; ROMUALDI, Chiara ; DRAGHICI, Sorin ; CAVALIERI, Duccio: The Biological Connection Markup Language: a SBGN-compliant format for visualization, filtering and analysis of biological pathways. In: *Bioinformatics* 27 (2011), Nr. 15, S. 2127–2133 (S. 83)
- [14] BIOPAX WORKGROUP: *BioPAX – Biological Pathways Exchange: Language Level 3, Release Version 1 Documentation*. <http://www.biopax.org/release/biopax-level3-documentation.pdf>, 2010. – [letzter Zugriff 07/2014] (S. 12)
- [15] BORNSTEIN, Benjamin J. ; KEATING, Sarah M. ; JOURAKU, Akiya ; HUCKA, Michael: LibSBML: an API Library for SBML. In: *Bioinformatics* 24 (2008), Nr. 6, S. 880–881 (S. 12 und 81)
- [16] BRANDES, Ulrik ; EIGLSPERGER, Markus ; HERMAN, Ivan ; HIMSOLT, Michael ; MARSHALL, M. S.: GraphML Progress Report – Structural Layer Proposal. In: MUTZEL, Petra (Hrsg.) ; JÜNGER, Michael (Hrsg.) ; LEIPERT, Sebastian (Hrsg.): *Graph Drawing 2001* Bd. 2265. Berlin Heidelberg, Germany : Springer, 2002 (Lecture Notes in Computer Science), S. 501–512 (S. 83 und 90)
- [17] BRANDES, Ulrik ; EIGLSPERGER, Markus ; LERNER, Jürgen: *GraphML Primer*. <http://graphml.graphdrawing.org/primer/graphml-primer.html>, 2004. – [letzter Zugriff 07/2014] (S. 83)
- [18] BRANDES, Ulrik ; EIGLSPERGER, Markus ; LERNER, Jürgen ; PICH, Christian: Graph Markup Language (GraphML). In: TAMASSIA, Roberto (Hrsg.): *Handbook of Graph*

- Drawing and Visualization*. Chapman and Hall/CRC, 2013 (Discrete Mathematics and Its Applications), Kapitel 16, S. 517–541 (S. 83 und 90)
- [19] BRAZMA, Alvis ; KRESTYANINOVA, Maria ; SARKANS, Ugis: Standards for systems biology. In: *Nature Reviews Genetics* 7 (2006), Nr. 8, S. 593–605 (S. 12)
- [20] BÜCHEL, Finja ; RODRIGUEZ, Nicolas ; SWAINSTON, Neil ; WRZODEK, Clemens ; CZAUDERNA, Tobias ; KELLER, Roland ; MITTAG, Florian ; SCHUBERT, Michael ; GLONT, Mihai ; GOLEBIEWSKI, Martin u. a.: Path2Models: large-scale generation of computational models from biochemical pathway maps. In: *BMC Systems Biology* 7 (2013), Nr. 1, S. 116.1–19 (S. 35, 99 und 101)
- [21] BÜCHEL, Finja ; WRZODEK, Clemens ; MITTAG, Florian ; DRÄGER, Andreas ; EICHNER, Johannes ; RODRIGUEZ, Nicolas ; LE NOVÈRE, Nicolas ; ZELL, Andreas: Qualitative translation of relations from BioPAX to SBML qual. In: *Bioinformatics* 28 (2012), Nr. 20, S. 2648–2653 (S. 65)
- [22] CALZONE, Laurence ; FAGES, François ; SOLIMAN, Sylvain: BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. In: *Bioinformatics* 22 (2006), Nr. 14, S. 1805–1807 (S. 36)
- [23] CASPI, Ron ; ALTMAN, Tomer ; BILLINGTON, Richard ; DREHER, Kate ; FOERSTER, Hartmut ; FULCHER, Carol A. ; HOLLAND, Timothy A. ; KESELER, Ingrid M. ; KOTHARI, Anamika ; KUBO, Aya u. a.: The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. In: *Nucleic Acids Research* 42 (2014), Nr. 1, S. D459–D471 (S. 39 und 107)
- [24] CAVALIERI, Duccio ; RIVERO, Damariz ; BELTRAME, Luca ; BUSCHOW, Sonja ; CALURA, Enrica ; RIZZETTO, Lisa ; GESSANI, Sandra ; GAUZZI, Maria ; REITH, Walter ; BAUR, Andreas u. a.: DC-ATLAS: a systems biology resource to dissect receptor specific signal transduction in dendritic cells. In: *Immunome Research* 6 (2010), Nr. 1, S. 10.1–12 (S. 35)
- [25] CHANDRAN, Deepak ; BERGMANN, Frank T. ; SAURO, Herbert M.: *Athena: Modular CAM/CAD Software for Synthetic Biology*. <http://arxiv.org/abs/0902.2598>, 2009. – [letzter Zugriff 07/2014] (S. 36)
- [26] CHELLIAH, Vijayalakshmi ; LAIBE, Camille ; LE NOVÈRE, Nicolas: BioModels Database: A Repository of Mathematical Models of Biological Processes. In: SCHNEIDER, Maria V. (Hrsg.): *In Silico Systems Biology*. Humana Press, 2013 (Methods in Molecular Biology 1021), Kapitel 10, S. 189–199 (S. 35 und 99)
- [27] COOK, Daniel L. ; FARLEY, Joel F. ; TAPSCOTT, Stephen J.: A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. In: *Genome Biology* 2 (2001), Nr. 4, S. research0012.1–10 (S. 37)

- [28] COURTOT, Mélanie ; JUTY, Nick ; KNÜPFER, Christian ; WALTEMATH, Dagmar ; ZHUKOVA, Anna ; DRÄGER, Andreas ; DUMONTIER, Michel ; FINNEY, Andrew ; GOLEBIEWSKI, Martin ; HASTINGS, Janna u. a.: Controlled vocabularies and semantics in systems biology. In: *Molecular Systems Biology* 7 (2011), Nr. 1, S. 543.1–12 (S. 13)
- [29] CROFT, David ; MUNDO, Antonio F. ; HAW, Robin ; MILACIC, Marija ; WEISER, Joel ; WU, Guanming ; CAUDY, Michael ; GARAPATI, Phani ; GILLESPIE, Marc ; KAMDAR, Maulik R. u. a.: The Reactome pathway knowledgebase. In: *Nucleic Acids Research* 42 (2014), Nr. 1, S. D472–D477 (S. 35 und 39)
- [30] CZAUDERNA, Tobias ; GRAFAHREND-BELAU, Eva ; HARTMANN, Anja ; JUNKER, Astrid ; KLUKAS, Christian ; SCHREIBER, Falk: *SBGN-PD map of Central Metabolism in Plants*. http://www.sbgn.org/Competition/Competition_2010/, 2010. – [letzter Zugriff 07/2014] (S. 96 und 97)
- [31] CZAUDERNA, Tobias ; KLUKAS, Christian ; SCHREIBER, Falk: Editing, validating and translating of SBGN maps. In: *Bioinformatics* 26 (2010), Nr. 18, S. 2340–2341 (S. 36, 81 und 85)
- [32] CZAUDERNA, Tobias ; WYBROW, Michael ; MARRIOTT, Kim ; SCHREIBER, Falk: Conversion of KEGG metabolic pathways to SBGN maps including automatic layout. In: *BMC Bioinformatics* 14 (2013), Nr. 1, S. 250.1–17 (S. 50)
- [33] DE BONO, Bernard ; GRENON, Pierre ; CZAUDERNA, Tobias ; SCHREIBER, Falk ; FENECH, Matthew E. ; LUTHERT, Philip J. ; FOSS, Alexander: *Development of Diabetic Retinopathy*. http://www.sbgn.org/Competition/Competition_2010/, 2010. – [letzter Zugriff 07/2014] (S. 99 und 100)
- [34] DECKARD, Anastasia ; BERGMANN, Frank T. ; SAURO, Herbert M.: Supporting the SBML layout extension. In: *Bioinformatics* 22 (2006), Nr. 23, S. 2966–2967 (S. 36, 81 und 90)
- [35] DEMIR, Emek ; BABUR, Özgün ; RODCHENKOV, Igor ; AKSOY, Bülent A. ; FUKUDA, Ken I. ; GROSS, Benjamin ; SÜMER, Onur S. ; BADER, Gary D. ; SANDER, Chris: Using Biological Pathway Data with Paxtools. In: *PLoS Computational Biology* 9 (2013), Nr. 9, S. e1003194.1–5 (S. 12 und 65)
- [36] DEMIR, Emek ; CARY, Michael P. ; PALEY, Suzanne ; FUKUDA, Ken I. ; LEMER, Christian ; VASTRIK, Imre ; WU, Guanming ; D’EUSTACHIO, Peter ; SCHAEFER, Carl ; LUCIANO, Joanne u. a.: The BioPAX community standard for pathway data sharing. In: *Nature Biotechnology* 28 (2010), Nr. 9, S. 935–942 (S. 12 und 13)
- [37] DI BATTISTA, Giuseppe ; EADES, Peter ; TAMASSIA, Roberto ; TOLLIS, Ioannis G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. New Jersey, USA : Prentice Hall, 1999 (S. 15 und 16)

- [38] DIESTEL, Reinhard: *Graphentheorie*. Berlin Heidelberg, Germany : Springer, 2000 (S. 32)
- [39] DILEK, Alptug ; BELVIRANLI, Mehmet E. ; DOGRUSÖZ, Ugur: VISIBIOweb: visualization and layout services for BioPAX pathway models. In: *Nucleic Acids Research* 38 (2010), Nr. Suppl 2, S. W150–W154 (S. 36)
- [40] DOGRUSÖZ, Ugur ; GIRAL, Erhan ; CETINTAS, Ahmet ; CIVRIL, Ali ; DEMIR, Emek: A layout algorithm for undirected compound graphs. In: *Information Sciences* 179 (2009), Nr. 7, S. 980–994 (S. 18)
- [41] DRÄGER, Andreas ; RODRIGUEZ, Nicolas ; DUMOUSSEAU, Marine ; DÖRR, Alexander ; WRZODEK, Clemens ; LE NOVÈRE, Nicolas ; ZELL, Andreas ; HUCKA, Michael: JSBML: a flexible Java library for working with SBML. In: *Bioinformatics* 27 (2011), Nr. 15, S. 2167–2168 (S. 12 und 81)
- [42] DROSTE, Peter ; WIECHERT, Wolfgang ; NÖH, Katharina: Semi-automatic drawing of metabolic networks. In: *Information Visualization* 11 (2012), Nr. 3, S. 171–187 (S. 36)
- [43] DWYER, Tim ; KOREN, Yehuda ; MARRIOTT, Kim: IPSep-CoLa: an incremental procedure for separation constraint layout of graphs. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 5, S. 821–828 (S. 17, 19 und 47)
- [44] DWYER, Tim ; MARRIOTT, Kim ; WYBROW, Michael: Dunnart: A Constraint-Based Network Diagram Authoring Tool. In: TOLLIS, Ioannis G. (Hrsg.) ; PATRIGNANI, Maurizio (Hrsg.): *Graph Drawing 2008* Bd. 5417. Berlin Heidelberg, Germany : Springer, 2009 (Lecture Notes in Computer Science), S. 420–431 (S. 36)
- [45] EADES, Peter ; LAI, Wei ; MISUE, Kazuo ; SUGIYAMA, Kozo: Preserving the mental map of a diagram. In: SANTO, Harold P. (Hrsg.): *Proceedings of the 1st International Conference on Computational Graphics and Visualization Techniques (Compugraphics) 1991* Bd. 1, 1991, S. 24–33 (S. 19)
- [46] FAGES, François ; SOLIMAN, Sylvain: From Reaction Models to Influence Graphs and Back: A Theorem. In: FISHER, Jasmin (Hrsg.): *Formal Methods in Systems Biology* Bd. 5054. Berlin Heidelberg, Germany : Springer, 2008 (Lecture Notes in Computer Science), S. 90–102 (S. 65)
- [47] FIELDING, Roy T.: *Architectural Styles and the Design of Network-based Software Architectures*. Irvine, CA, USA, University of California, Diss., 2000 (S. 91)
- [48] FIELDING, Roy T. ; TAYLOR, Richard N.: Principled Design of the Modern Web Architecture. In: GHEZZI, Carlo (Hrsg.) ; JAZAYERI, Mehdi (Hrsg.) ; WOLF, Alexander L. (Hrsg.): *Proceedings of the 22nd International Conference on Software Engineering*. New York, NY, USA : ACM, 2000, S. 407–416 (S. 91)

- [49] FIELDING, Roy T. ; TAYLOR, Richard N.: Principled Design of the Modern Web Architecture. In: *ACM Transactions on Internet Technology* 2 (2002), Nr. 2, S. 115–150 (S. [91](#))
- [50] FREEMAN, Tom C. ; RAZA, Sobia ; THEOCHARIDIS, Athanasios ; GHAZAL, Peter: The mEPN scheme: an intuitive and flexible graphical system for rendering biological pathways. In: *BMC Systems Biology* 4 (2010), Nr. 1, S. 65.1–13 (S. [37](#))
- [51] FRIEDRICH, Carsten ; SCHREIBER, Falk: Visualisation and navigation methods for typed protein-protein interaction networks. In: *Applied Bioinformatics* 2 (2003), Nr. Suppl 3, S. 19–24 (S. [18](#))
- [52] FUJITA, Kazuhiro A. ; OSTASZEWSKI, Marek ; MATSUOKA, Yukiko ; GHOSH, Samik ; GLAAB, Enrico ; TREFOIS, Christophe ; CRESPO, Isaac ; PERUMAL, Thanneer M. ; JURKOWSKI, Wiktor ; ANTONY, Paul M. A. u. a.: Integrating Pathways of Parkinson’s Disease in a Molecular Interaction Map. In: *Molecular Neurobiology* 49 (2014), Nr. 1, S. 88–102 (S. [35](#))
- [53] FUNAHASHI, Akira ; MATSUOKA, Yukiko ; JOURAKU, Akiya ; MOROHASHI, Mineo ; KIKUCHI, Norihiro ; KITANO, Hiroaki: CellDesigner 3.5: A Versatile Modeling Tool for Biochemical Networks. In: *Proceedings of the IEEE* 96 (2008), Nr. 8, S. 1254–1265 (S. [36](#))
- [54] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA : Addison-Wesley, 1995 (Addison-Wesley Professional Computing Series) (S. [86](#))
- [55] GANSNER, Emden R. ; KOREN, Yehuda ; NORTH, Stephen: Graph Drawing by Stress Majorization. In: PACH, János (Hrsg.): *Graph Drawing 2004* Bd. 3383. Berlin Heidelberg, Germany : Springer, 2005 (Lecture Notes in Computer Science), S. 239–250 (S. [17](#))
- [56] GAREY, Michael R. ; JOHNSON, David S.: Crossing Number is NP-Complete. In: *SIAM Journal on Algebraic Discrete Methods* 4 (1983), Nr. 3, S. 312–316 (S. [44](#))
- [57] GAUGES, Ralph ; ROST, Ursula ; SAHLE, Sven ; WEGNER, Katja: A model diagram layout extension for SBML. In: *Bioinformatics* 22 (2006), Nr. 15, S. 1879–1885 (S. [52](#), [83](#) und [101](#))
- [58] GEHLENBORG, Nils ; O’DONOGHUE, Seán I. ; BALIGA, Nitin S. ; GOESMANN, Alexander ; HIBBS, Matthew A. ; KITANO, Hiroaki ; KOHLBACHER, Oliver ; NEUWEGER, Heiko ; SCHNEIDER, Reinhard ; TENENBAUM, Dan ; GAVIN, Anne-Claude: Visualization of omics data for systems biology. In: *Nature Methods* 7 (2010), Nr. 3 Suppl, S. S56–S68 (S. [1](#) und [105](#))

- [59] GENÇ, Burkay ; DOGRUSÖZ, Ugur: A layout algorithm for signaling pathways. In: *Information Sciences* 176 (2006), Nr. 2, S. 135–149 (S. 18)
- [60] GONÇALVES, Emanuel ; VAN IERSEL, Martijn P. ; SAEZ-RODRIGUEZ, Julio: CySBGN: A Cytoscape plug-in to integrate SBGN maps. In: *BMC Bioinformatics* 14 (2013), Nr. 1, S. 17.1–7 (S. 36 und 81)
- [61] GRAFAHREND-BELAU, Eva ; WEISE, Stephan ; KOSCHÜTZKI, Dirk ; SCHOLZ, Uwe ; JUNKER, Björn H. ; SCHREIBER, Falk: MetaCrop: a detailed database of crop plant metabolism. In: *Nucleic Acids Research* 36 (2008), Nr. 1, S. D954–D958 (S. 90, 93 und 94)
- [62] HESSE, Wolfgang: Ontologie(n). In: *Informatik Spektrum* 25 (2002), Nr. 6, S. 477–480 (S. 13)
- [63] HIMSOLT, Michael: Graphlet: design and implementation of a graph editor. In: *Software: Practice and Experience* 30 (2000), Nr. 11, S. 1303–1324 (S. 90)
- [64] HOOPS, Stefan ; SAHLE, Sven ; GAUGES, Ralph ; LEE, Christine ; PAHLE, Jürgen ; SIMUS, Natalia ; SINGHAL, Mudita ; XU, Liang ; MENDES, Pedro ; KUMMER, Ursula: COPASI – a COMplex PATHway SIMulator. In: *Bioinformatics* 22 (2006), Nr. 24, S. 3067–3074 (S. 36)
- [65] HOWER, Walter ; GRAF, Winfried H.: A bibliographical survey of constraint-based approaches to CAD, graphics, layout, visualization, and related topics. In: *Knowledge-Based Systems* 9 (1996), Nr. 7, S. 449–464 (S. 17)
- [66] HUCKA, Michael ; BERGMANN, Frank T. ; HOOPS, Stefan ; KEATING, Sarah ; SAHLE, Sven ; SCHAFF, James C. ; SMITH, Lucian ; WILKINSON, Darren: *The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core*. <http://sbml.org/Documents/Specifications>, 2010. – [letzter Zugriff 07/2014] (S. 12)
- [67] HUCKA, Michael ; FINNEY, Andrew ; SAURO, Herbert M. ; BOLOURI, Hamid ; DOYLE, John C. ; KITANO, Hiroaki ; AND THE REST OF THE SBML FORUM: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. In: *Bioinformatics* 19 (2003), Nr. 4, S. 524–531 (S. 12)
- [68] IGARASHI, Takeo ; MATSUOKA, Satoshi ; KAWACHIYA, Sachiko ; TANAKA, Hidehiko: Interactive Beautification: A Technique for Rapid Geometric Design. In: ROBERTSON, George G. (Hrsg.) ; SCHMANDT, Chris (Hrsg.): *ACM Symposium on User Interface Software and Technology*. New York, NY, USA : ACM, 1997, S. 105–114 (S. 17)
- [69] JANSSON, Andreas ; JIRSTRAND, Mats: Biochemical modeling with Systems Biology Graphical Notation. In: *Drug Discovery Today* 15 (2010), Nr. 9–10, S. 365–370 (S. 36)

- [70] JOHNNIE, Joseph G. ; AUSTIN, Mark ; SRIRAM, Ganesh ; CONWAY, Matt ; MISRA, Ashish: Systems Engineering and Metabolic Engineering: A Side-by-Side Comparison. In: *Procedia Computer Science* 8 (2012), S. 226–231 (S. 37)
- [71] JOHNNIE, Joseph G.: *A Systems Engineering Framework for Metabolic Engineering Experiments*. College Park, MD, USA, University of Maryland, Diplomarbeit, 2011 (S. 103)
- [72] JUNGNICHEL, Dieter: *Graphs, Networks and Algorithms*. Berlin Heidelberg, Germany : Springer, 2008 (Algorithms and Computation in Mathematics) (S. 14)
- [73] JUNKER, Astrid ; HARTMANN, Anja ; SCHREIBER, Falk ; BÄUMLEIN, Helmut: An engineer’s view on regulation of seed development. In: *Trends in Plant Science* 15 (2010), Nr. 6, S. 303–307 (S. 35 und 103)
- [74] JUNKER, Astrid ; JUNKER, Björn H.: Synthetic Gene Networks in Plant Systems. In: WEBER, Wilfried (Hrsg.) ; FUSSENEGGER, Martin (Hrsg.): *Synthetic Gene Networks: Methods and Protocols*. Humana Press, 2012 (Methods in Molecular Biology 813), Kapitel 21, S. 343–358 (S. 103)
- [75] JUNKER, Astrid ; ROHN, Hendrik ; CZAUDERNA, Tobias ; KLUKAS, Christian ; HARTMANN, Anja ; SCHREIBER, Falk: Creating interactive, web-based and data-enriched maps with the Systems Biology Graphical Notation. In: *Nature Protocols* 7 (2012), S. 579–593 (S. 90, 92 und 94)
- [76] JUNKER, Astrid ; SOROKIN, Anatoly ; CZAUDERNA, Tobias ; SCHREIBER, Falk ; MAZEIN, Alexander: Wiring diagrams in biology: towards the standardized representation of biological information. In: *Trends in Biotechnology* 30 (2012), Nr. 11, S. 555–557 (S. 74)
- [77] JUNKER, Björn H. ; KLUKAS, Christian ; SCHREIBER, Falk: VANTED: a system for advanced data analysis and visualization in the context of biological networks. In: *BMC Bioinformatics* 7 (2006), S. 109.1–13 (S. 85)
- [78] JUNKER, Björn H. (Hrsg.) ; SCHREIBER, Falk (Hrsg.): *Analysis of Biological Networks*. Hoboken, NJ, USA : John Wiley & Sons, 2008 (Wiley Series on Bioinformatics: Computational Techniques and Engineering) (S. 9, 10, 14 und 15)
- [79] KAMADA, Tomihisa ; KAWAI, Satoru: An Algorithm for Drawing General Undirected Graphs. In: *Information Processing Letters* 31 (1989), Nr. 1, S. 7–15 (S. 17)
- [80] KANEHISA, Minoru ; GOTO, Susumu ; SATO, Yoko ; KAWASHIMA, Masayuki ; FURUMICHI, Miho ; TANABE, Mao: Data, information, knowledge and principle: back to metabolism in KEGG. In: *Nucleic Acids Research* 42 (2014), Nr. 1, S. D199–D205 (S. 39)

- [81] KARP, Peter D. ; PALEY, Suzanne: Automated Drawing of Metabolic Pathways. In: LIM, Hwa A. (Hrsg.) ; CANTOR, Charles R. (Hrsg.): *Proceedings of the Third International Conference on Bioinformatics and Genome Research*. Singapore : World Scientific, 1995, S. 225–238 (S. 18)
- [82] KELDER, Thomas ; VAN IERSEL, Martijn P. ; HANSPERS, Kristina ; KUTMON, Martina ; CONKLIN, Bruce R. ; EVELO, Chris T. ; PICO, Alexander R.: WikiPathways: building research communities on biological pathways. In: *Nucleic Acids Research* 40 (2012), Nr. 1, S. D1301–D1307 (S. 107)
- [83] KÉPÈS, François (Hrsg.): *Biological Networks*. Singapore : World Scientific, 2007 (Complex Systems and Interdisciplinary Science, Volume 3) (S. 9 und 10)
- [84] KHOLODENKO, Boris N. ; KIYATKIN, Anatoly ; BRUGGEMAN, Frank J. ; SONTAG, Eduardo ; WESTERHOFF, Hans V. ; HOEK, Jan B.: Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. In: *Proceedings of the National Academy of Sciences* 99 (2002), Nr. 20, S. 12841–12846 (S. 64)
- [85] KITANO, Hiroaki (Hrsg.): *Foundations of Systems Biology*. Cambridge, MA, USA und London, UK : The MIT Press, 2001 (S. 11 und 12)
- [86] KITANO, Hiroaki: Systems Biology: A Brief Overview. In: *Science* 295 (2002), Nr. 5560, S. 1662–1664 (S. 1, 11 und 12)
- [87] KITANO, Hiroaki: A graphical notation for biochemical networks. In: *BIOSILICO* 1 (2003), Nr. 5, S. 169–176 (S. 22 und 37)
- [88] KITANO, Hiroaki ; FUNAHASHI, Akira ; MATSUOKA, Yukiko ; ODA, Kanae: Using process diagrams for the graphical representation of biological networks. In: *Nature Biotechnology* 23 (2005), Nr. 8, S. 961–966 (S. 21, 22 und 37)
- [89] KLIPP, Edda ; HERWIG, Ralf ; KOWALD, Axel ; WIERLING, Christoph ; LEHRACH, Hans: *Systems Biology in Practice: Concepts, Implementation and Application*. Weinheim, Germany : Wiley-VCH Verlag GmbH & Co. KGaA, 2005 (S. 11)
- [90] KLIPP, Edda ; LIEBERMEISTER, Wolfram ; HELBIG, Anselm ; KOWALD, Axel ; SCHABER, Jörg: Systems biology standards – the community speaks. In: *Nature Biotechnology* 25 (2007), Nr. 4, S. 390–391 (S. 21)
- [91] KOHN, Kurt W. ; ALADJEM, Mirit I. ; WEINSTEIN, John N. ; POMMIER, Yves: Molecular Interaction Maps of Bioregulatory Networks: A General Rubric for Systems Biology. In: *Molecular Biology of the Cell* 17 (2006), Nr. 1, S. 1–13 (S. 22 und 37)
- [92] KOJIMA, Kaname ; NAGASAKI, Masao ; MIYANO, Satoru: An efficient biological pathway layout algorithm combining grid-layout and spring embedder for complicated cellular location information. In: *BMC Bioinformatics* 11 (2010), Nr. 1, S. 335.1–15 (S. 19)

- [93] KOLCH, Walter: Meaningful relationships: the regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. In: *Biochemical Journal* 351 (2000), Nr. 2, S. 289–305 (S. 64)
- [94] KOLLIAS, Aris N. ; ULBIG, Michael W.: Diabetische Retinopathie: Frühzeitige Diagnostik und effiziente Therapie. In: *Deutsches Ärzteblatt* 107 (2010), Nr. 5, S. 75–84 (S. 96)
- [95] KOLPAKOV, Fedor ; PUZANOV, Mikhail ; KOSHUKOV, Alexander: BioUML: Visual modeling, automated code generation and simulation of biological systems. In: KOLCHANOV, Nikolay (Hrsg.) ; HOFESTÄDT, Ralf (Hrsg.): *Proceedings of the Fifth International Conference on Bioinformatics of Genome Regulation and Structure* Bd. 3. Novosibirsk, Russia : Institute of Cytology and Genetics SB RAS, 2006, S. 281–284 (S. 36)
- [96] KRAUSE, Falko ; SCHULZ, Marvin ; RIPKENS, Ben ; FLÖTTMANN, Max ; KRANTZ, Marcus ; KLIPP, Edda ; HANDORF, Thomas: Biographer: web-based editing and rendering of SBGN compliant biochemical networks. In: *Bioinformatics* 29 (2013), Nr. 11, S. 1467–1468 (S. 36, 92 und 106)
- [97] KREBS, Hans A. ; JOHNSON, William A.: The role of citric acid in intermediate metabolism in animal tissues. In: *Enzymologia* 4 (1937), S. 148–156. – Nachdruck in: *FEBS Letters* 117 (1980), Nr. S1, S. K2–K10 (S. 22)
- [98] KURLANDER, David ; FEINER, Steven: Inferring constraints from multiple snapshots. In: *ACM Transactions on Graphics* 12 (1993), Nr. 4, S. 277–304 (S. 17)
- [99] LE NOVÈRE, Nicolas ; COURTOT, Mélanie ; LAIBE, Camille: Adding Semantics in Kinetics Models of Biochemical Pathways. In: KETTNER, Carsten (Hrsg.) ; HICKS, Martin G. (Hrsg.): *Proceedings of the 2nd International ESCEC Symposium on Experimental Standard Conditions on Enzyme Characterizations*. Frankfurt am Main, Germany : Beilstein-Institut, 2007, S. 137–153 (S. 13)
- [100] LE NOVÈRE, Nicolas ; HUCKA, Michael ; MI, Huaiyu ; MOODIE, Stuart L. ; SCHREIBER, Falk ; SOROKIN, Anatoly ; DEMIR, Emek ; WEGNER, Katja ; ALADJEM, Mirit I. ; WIMALARATNE, Sarala M. u. a.: The Systems Biology Graphical Notation. In: *Nature Biotechnology* 27 (2009), Nr. 8, S. 735–741 (S. 1, 13, 21, 22, 23, 37 und 39)
- [101] LE NOVÈRE, Nicolas ; MOODIE, Stuart L. ; SOROKIN, Anatoly ; SCHREIBER, Falk ; MI, Huaiyu: *Systems Biology Graphical Notation: Entity Relationship language Level 1*. <http://dx.doi.org/10.1038/npre.2011.5902.1>, 2011. – [letzter Zugriff 07/2014] (S. 23 und 28)
- [102] LEHNINGER, Albert L. ; NELSON, David L. ; COX, Michael M.: *Prinzipien der Biochemie*. Heidelberg Berlin, Germany und Oxford, UK : Spektrum Akademischer Verlag, 1994 (S. 61, 79, 142 und 143)

- [103] LI, Weijiang ; KURATA, Hiroyuki: A grid layout algorithm for automatic drawing of biochemical networks. In: *Bioinformatics* 21 (2005), Nr. 9, S. 2036–2042 (S. 18)
- [104] LIKIĆ, Vladimir A. ; MCCONVILLE, Malcolm J. ; LITHGOW, Trevor ; BACIC, Antony: Systems Biology: The Next Frontier for Bioinformatics. In: *Advances in Bioinformatics* 2010 (2010), S. 1–10. – Artikel-ID 268925 (S. 12)
- [105] LUNA, Augustin ; KARAC, Evrim I. ; SUNSHINE, Margot ; CHANG, Lucas ; NUSSINOV, Ruth ; ALADJEM, Mirit I. ; KOHN, Kurt W.: A formal MIM specification and tools for the common exchange of MIM diagrams: an XML-Based format, an API, and a validation method. In: *BMC Bioinformatics* 12 (2011), Nr. 1, S. 167.1–13 (S. 22 und 37)
- [106] LYNCH, Kevin: *The Image of the City*. Cambridge, MA, USA : The MIT Press, 1960 (S. 19)
- [107] MÄDER, Ulrike ; SCHMEISKY, Arne G. ; FLÓREZ, Lope A. ; STÜLKE, Jörg: SubtiWiki – a comprehensive community resource for the model organism *Bacillus subtilis*. In: *Nucleic Acids Research* 40 (2012), Nr. 1, S. D1278–D1287 (S. 35)
- [108] MALLIKARJUN, Venkatesh ; CLARKE, David J. ; CAMPBELL, Colin J.: Cellular redox potential and the biomolecular electrochemical series: A systems hypothesis. In: *Free Radical Biology and Medicine* 53 (2012), Nr. 2, S. 280–288 (S. 37)
- [109] MAZEIN, Alexander ; WATTERSON, Steven ; HSIEH, Wei-Yuan ; GRIFFITHS, William J. ; GHAZAL, Peter: A comprehensive machine-readable view of the mammalian cholesterol biosynthesis pathway. In: *Biochemical Pharmacology* 86 (2013), Nr. 1, S. 56–66 (S. 103)
- [110] METSKER, Steven J.: *Design Patterns Java™ Workbook*. Boston, MA, USA : Addison-Wesley, 2002 (Software Patterns Series) (S. 86)
- [111] MI, Huaiyu ; MURUGANUJAN, Anushya ; THOMAS, Paul D.: PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. In: *Nucleic Acids Research* 41 (2013), Nr. 1, S. D377–D386 (S. 35 und 39)
- [112] MI, Huaiyu ; SCHREIBER, Falk ; LE NOVÈRE, Nicolas ; MOODIE, Stuart L. ; SOROKIN, Anatoly: *Systems Biology Graphical Notation: Activity Flow language Level 1*. <http://dx.doi.org/10.1038/npre.2009.3724.1>, 2009. – [letzter Zugriff 07/2014] (S. 23, 30 und 33)
- [113] MINTERT, Stefan (Hrsg.): *XML & Co. – Die W3C-Spezifikation für Dokumenten- und Datenarchitektur*. München, Deutschland : Addison-Wesley, 2002 (S. 12, 77 und 78)

- [114] MISUE, Kazuo ; EADES, Peter ; LAI, Wei ; SUGIYAMA, Kozo: Layout Adjustment and the Mental Map. In: *Journal of Visual Languages and Computing* 6 (1995), Nr. 2, S. 183–210 (S. 19)
- [115] MIZUNO, Satoshi ; IJIMA, Risa ; OGISHIMA, Soichi ; KIKUCHI, Masataka ; MATSUOKA, Yukiko ; GHOSH, Samik ; MIYAMOTO, Tadashi ; MIYASHITA, Akinori ; KUWANO, Ryozo ; TANAKA, Hiroshi: AlzPathway: a comprehensive map of signaling pathways of Alzheimer’s disease. In: *BMC Systems Biology* 6 (2012), Nr. 1, S. 52.1–10 (S. 35)
- [116] MOODIE, Stuart L. ; LE NOVÈRE, Nicolas ; DEMIR, Emek ; MI, Huaiyu ; VILLÉGER, Alice C.: *Systems Biology Graphical Notation: Process Description language Level 1*. <http://dx.doi.org/10.1038/npre.2011.3721.4>, 2010. – [letzter Zugriff 07/2014] (S. 23, 26, 43, 61, 64, 65, 79, 142 und 143)
- [117] MOODIE, Stuart L. ; SOROKIN, Anatoly ; GORYANIN, Igor ; GHAZAL, Peter: A Graphical Notation to describe the Logical Interactions of Biological Pathways. In: *Journal of Integrative Bioinformatics* 3 (2006), Nr. 2, S. 36.1–11 (S. 22 und 37)
- [118] NENTWICH, Martin M. ; ULBIG, Michael W.: Diabetische Retinopathie. In: *Der Diabetologe* 6 (2010), Nr. 6, S. 491–502 (S. 96)
- [119] OLIVIER, Brett G. ; SNOEP, Jacky L.: Web-based kinetic modelling using JWS Online. In: *Bioinformatics* 20 (2004), Nr. 13, S. 2143–2144 (S. 36)
- [120] PALSSON, Bernhard O.: *Systems Biology: Properties of Reconstructed Networks*. New York, NY, USA : Cambridge University Press, 2006 (S. 11)
- [121] PENNISI, Elizabeth: How Will Big Pictures Emerge From a Sea of Biological Data? In: *Science* 309 (2005), Nr. 5731, S. 94 (S. 1)
- [122] PIRSON, Isabelle ; FORTEMAISON, Nathalie ; JACOBS, Christine ; DREMIER, Sarah ; DUMONT, Jacques E. ; MAENHAUT, Carine: The visual display of regulatory information and networks. In: *Trends in Cell Biology* 10 (2000), Nr. 10, S. 404–408 (S. 37)
- [123] PÜSCHEL, Ute (Hrsg.) ; KUMMER, Gertrud (Hrsg.): *Wissensspeicher Biologie*. Berlin, DDR : Volk und Wissen Volkseigener Verlag, 1979 (S. 7 und 9)
- [124] REYES-PALOMARES, Armando ; MONTAÑEZ, Raul ; REAL-CHICHARRO, Alejandro ; CHNIBER, Othmane ; KERZAZI, Amine ; NAVAS-DELGADO, Ismael ; MEDINA, Miguel A. ; ALDANA-MONTES, José F. ; SÁNCHEZ-JIMÉNEZ, Francisca: Systems biology metabolic modeling assistant: an ontology-based tool for the integration of metabolic data in kinetic modeling. In: *Bioinformatics* 25 (2009), Nr. 6, S. 834–835 (S. 36)
- [125] ROHN, Hendrik ; JUNKER, Astrid ; HARTMANN, Anja ; GRAFAHREND-BELAU, Eva ; TREUTLER, Hendrik ; KLAPPERSTÜCK, Matthias ; CZAUDERNA, Tobias ; KLUKAS,

- Christian ; SCHREIBER, Falk: VANTED v2: a framework for systems biology applications. In: *BMC Systems Biology* 6 (2012), Nr. 1, S. 139.1–13 (S. 85)
- [126] ROUGNY, Adrien ; FROIDEVAUX, Christine ; YAMAMOTO, Yoshitaka ; INOUE, Katsumi: Translating the SBGN-AF language into logic to analyze signalling networks. In: INOUE, Katsumi (Hrsg.) ; SAKAMA, Chiaki (Hrsg.): *Proceedings of the 1st International Workshop on Learning and Non Monotonic Reasoning (LNMR)*, 2013, S. 43–54. – <http://arxiv.org/abs/1311.4639> [letzter Zugriff 07/2014] (S. 66 und 107)
- [127] SARAIYA, Purvi ; NORTH, Chris ; DUCA, Karen: Visualizing Biological Pathways: Requirements Analysis, Systems Evaluation and Research Agenda. In: *Information Visualization* 4 (2005), Nr. 3, S. 191–205 (S. 1 und 39)
- [128] SCHREIBER, Falk: High Quality Visualization of Biochemical Pathways in BioPath. In: *In Silico Biology* 2 (2002), Nr. 2, S. 59–73 (S. 17 und 18)
- [129] SCHREIBER, Falk ; COLMSEE, Christian ; CZAUDERNA, Tobias ; GRAFAHREND-BELAU, Eva ; HARTMANN, Anja ; JUNKER, Astrid ; JUNKER, Björn H. ; KLAPPERSTÜCK, Matthias ; SCHOLZ, Uwe ; WEISE, Stephan: MetaCrop 2.0: managing and exploring information about crop plant metabolism. In: *Nucleic Acids Research* 40 (2012), Nr. 1, S. D1173–D1177 (S. 35, 90, 93 und 95)
- [130] SCHREIBER, Falk ; DWYER, Tim ; MARRIOTT, Kim ; WYBROW, Michael: A generic algorithm for layout of biological networks. In: *BMC Bioinformatics* 10 (2009), Nr. 1, S. 375.1–12 (S. 17 und 19)
- [131] SCHUBERT, Matthias: *Mathematik für Informatiker*. Wiesbaden, Deutschland : Vieweg + Teubner, 2009 (S. 13)
- [132] SEGER, Rony ; KREBS, Edwin G.: The MAPK signaling cascade. In: *The FASEB Journal* 9 (1995), Nr. 9, S. 726–735 (S. 64)
- [133] SHAUL, Yoav D. ; SEGER, Rony: The MEK/ERK cascade: From signaling specificity to diverse functions. In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1773 (2007), Nr. 8, S. 1213–1226 (S. 64)
- [134] SOROKIN, Anatoly ; PALIY, Kirill ; SELKOV, Alexey ; DEMIN, Oleg V. ; DRONOV, Serge ; GHAZAL, Peter ; GORYANIN, Igor: The Pathway Editor: A tool for managing complex biological networks. In: *IBM Journal of Research and Development* 50 (2006), Nr. 6, S. 561–573 (S. 36)
- [135] SREENIVASIAH, Pradeep K. ; RANI, Shilpa ; CAYETANO, Joseph ; ARUL, Novino ; KIM, Do H.: IPAVS: Integrated Pathway Resources, Analysis and Visualization System. In: *Nucleic Acids Research* 40 (2012), Nr. 1, S. D803–D808 (S. 35)

- [136] SUGIYAMA, Kozo: *Graph Drawing and Applications for Software and Knowledge Engineers*. Singapore : World Scientific, 2002 (Series on Software Engineering and Knowledge Engineering, Volume 11) (S. 15)
- [137] SYMONS, Stephan ; NIESELT, Kay: MGv: a generic graph viewer for comparative omics data. In: *Bioinformatics* 27 (2011), Nr. 16, S. 2248–2255 (S. 36 und 51)
- [138] THE GENE ONTOLOGY CONSORTIUM: The Gene Ontology in 2010: extensions and refinements. In: *Nucleic Acids Research* 38 (2010), Nr. 1, S. D331–D335 (S. 71)
- [139] THEUS, Martin ; URBANEK, Simon: *Interactive Graphics for Data Analysis - Principles and Examples*. Chapman and Hall/CRC, 2009 (Computer Science and Data Analysis Series) (S. 90)
- [140] THEWS, Gerhard ; MUTSCHLER, Ernst ; VAUPEL, Peter: *Anatomie, Physiologie, Pathophysiologie des Menschen*. Wissenschaftliche Verlagsgesellschaft Stuttgart, 1991 (S. 96)
- [141] TITTMANN, Peter: *Graphentheorie: Eine anwendungsorientierte Einführung*. München, Deutschland und Wien, Österreich : Fachbuchverlag Leipzig im Carl Hanser Verlag, 2003 (Mathematik – Studienhilfen) (S. 14)
- [142] TOLMAN, Edward C.: Cognitive maps in rats and men. In: *The Psychological Review* 55 (1948), Nr. 4, S. 189–208 (S. 19)
- [143] VAN IERSEL, Martijn P. ; KELDER, Thomas ; PICO, Alexander ; HANSPERS, Kristina ; COORT, Susan ; CONKLIN, Bruce ; EVELO, Chris: Presenting and exploring biological pathways with PathVisio. In: *BMC Bioinformatics* 9 (2008), Nr. 1, S. 399.1–9 (S. 36, 81, 92 und 106)
- [144] VAN IERSEL, Martijn P. ; VILLÉGER, Alice C. ; CZAUDERNA, Tobias ; BOYD, Sarah E. ; BERGMANN, Frank T. ; LUNA, Augustin ; DEMIR, Emek ; SOROKIN, Anatoly ; DOGRUSÖZ, Ugur ; MATSUOKA, Yukiko u. a.: Software support for SBGN maps: SBGN-ML and LibSBGN. In: *Bioinformatics* 28 (2012), Nr. 15, S. 2016–2021 (S. 83)
- [145] VILLÉGER, Alice C. ; PETTIFER, Stephen R. ; KELL, Douglas B.: Arcadia: a visualization tool for metabolic pathways. In: *Bioinformatics* 26 (2010), Nr. 11, S. 1470–1471 (S. 36 und 90)
- [146] VOET, Donald ; VOET, Judith G. ; PRATT, Charlotte W.: *Lehrbuch der Biochemie*. Weinheim, Deutschland : Wiley-VCH Verlag GmbH & Co. KGaA, 2002 (S. 7 und 9)
- [147] VOGT, Torsten ; CZAUDERNA, Tobias ; SCHREIBER, Falk: Translation of SBGN maps: Process Description to Activity Flow. In: *BMC Systems Biology* 7 (2013), Nr. 1, S. 115.1–19 (S. 66)

- [148] WEGNER, Katja ; KNABE, Johannes ; ROBINSON, Mark ; EGRI-NAGY, Attila ; SCHILSTRA, Maria ; NEHANIV, Chrystopher: The NetBuilder' project: development of a tool for constructing, simulating, evolving, and analysing complex regulatory networks. In: *BMC Systems Biology* 1 (2007), Nr. Suppl 1, S. P72.1–2 (S. 36)
- [149] WEGNER, Katja ; KUMMER, Ursula: A new dynamical layout algorithm for complex biochemical reaction networks. In: *BMC Bioinformatics* 6 (2005), Nr. 1, S. 212.1–12 (S. 18)
- [150] WIKIPEDIA: *Tricarboxylic acid cycle (also known as the citric acid cycle) and some preceding steps*. http://en.wikipedia.org/wiki/File:Citric_acid_cycle_with_aconitate_2.svg, 2008. – [letzter Zugriff 07/2014] (S. 22)
- [151] WRZODEK, Clemens ; BÜCHEL, Finja ; RUFF, Manuel ; DRÄGER, Andreas ; ZELL, Andreas: Precise generation of systems biology models from KEGG pathways. In: *BMC Systems Biology* 7 (2013), Nr. 1, S. 15.1–12 (S. 36 und 51)
- [152] WYBROW, Michael ; MARRIOTT, Kim ; STUCKEY, Peter J.: Orthogonal connector routing. In: EPPSTEIN, David (Hrsg.) ; GANSNER, Emden R. (Hrsg.): *Graph Drawing 2009* Bd. 5849. Berlin Heidelberg, Germany : Springer, 2010 (Lecture Notes in Computer Science), S. 219–231 (S. 19 und 48)

Anhang A

Beispiele für die Übersetzung von KEGG Stoffwechselweg-Karten in SBGN PD-Karten

In Abschnitt 4.3 wurde bereits der Pentose-Phosphat-Stoffwechselweg als Beispiel für die Übersetzung von KEGG Stoffwechselweg-Karten in SBGN PD-Karten dargestellt. Hier sollen nun als weitere Beispiel die Glykolyse / Glukoneogenese¹ (Abbildung A.1 und A.2), der Zitratzyklus² (Abbildung A.3 und A.4) und der Valin, Leucin und Isoleucin-Abbau³ (Abbildung A.5 und A.6) gezeigt werden.

¹<http://www.kegg.jp/kegg/pathway/map/map00010.html>

²<http://www.kegg.jp/kegg/pathway/map/map00020.html>

³<http://www.kegg.jp/kegg/pathway/map/map00280.html>

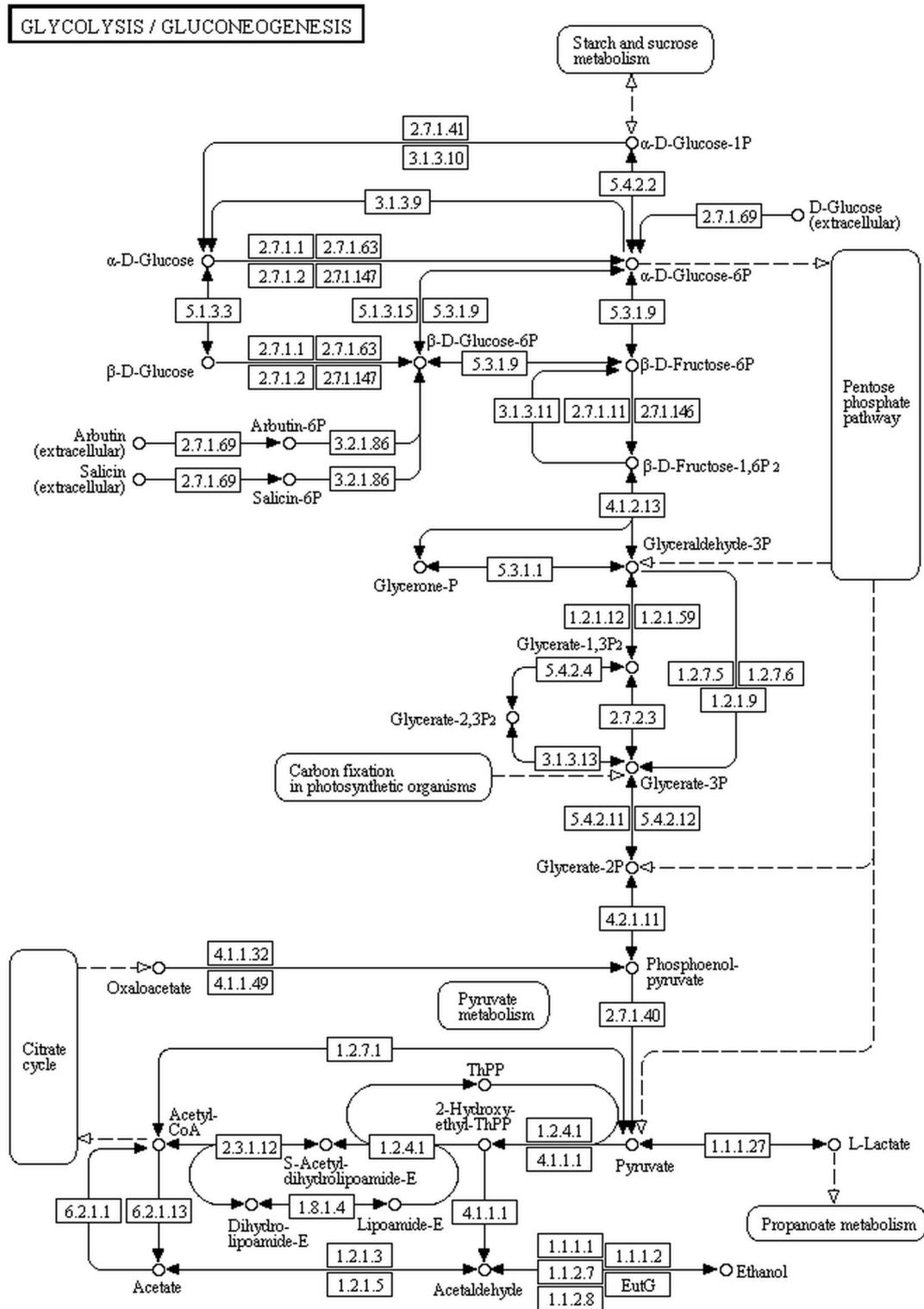


Abbildung A.1: Referenzbild der Glykolyse / Glukoneogenese aus der KEGG-Datenbank.

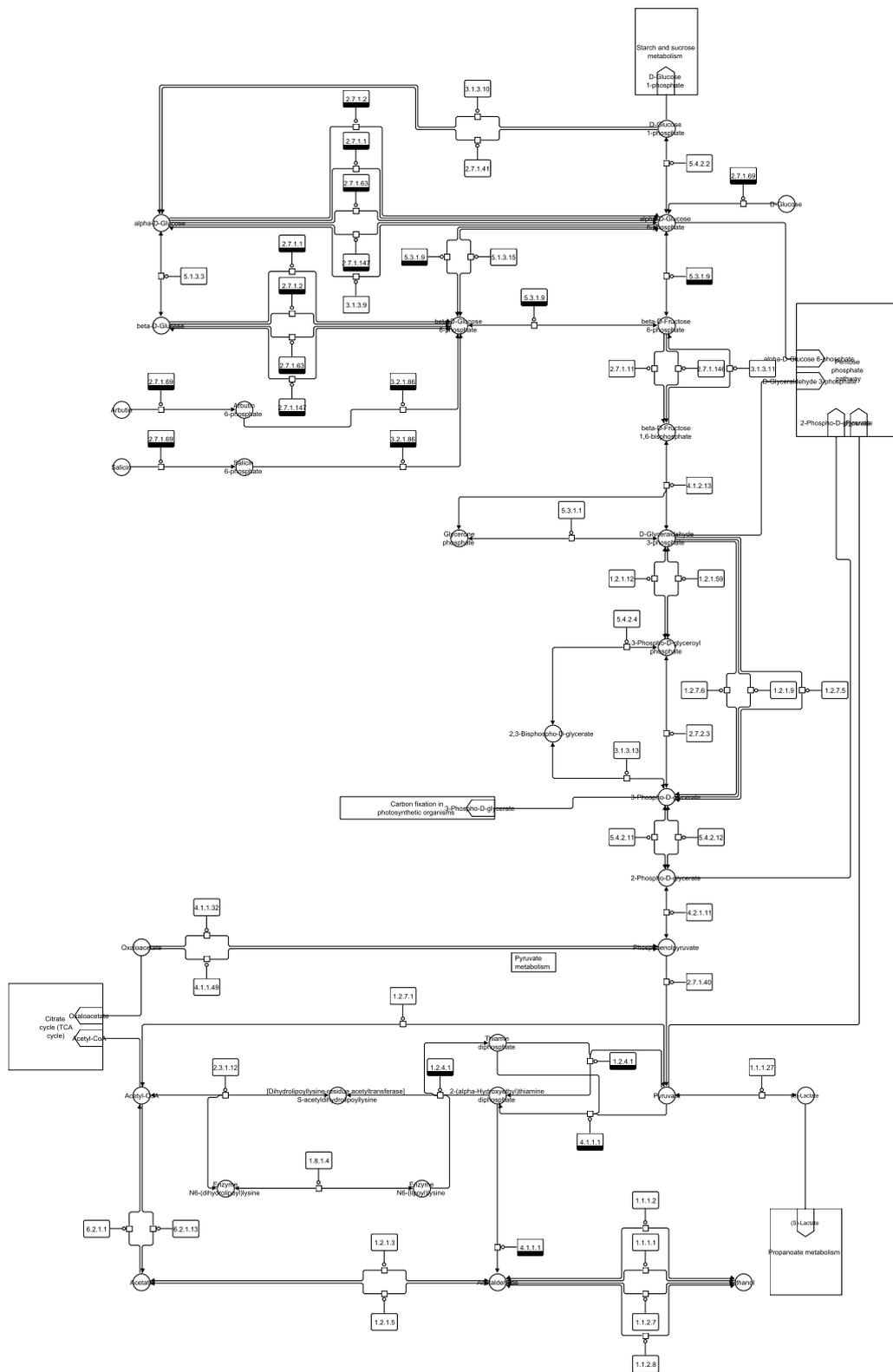


Abbildung A.2: Die Glykolyse / Glukoneogenese aus der KEGG-Datenbank wurde basierend auf der KGML-Datei in eine SBGN PD-Karte übersetzt, anschließend wurde der beschriebene Layout-Algorithmus angewendet (siehe Kapitel 4).

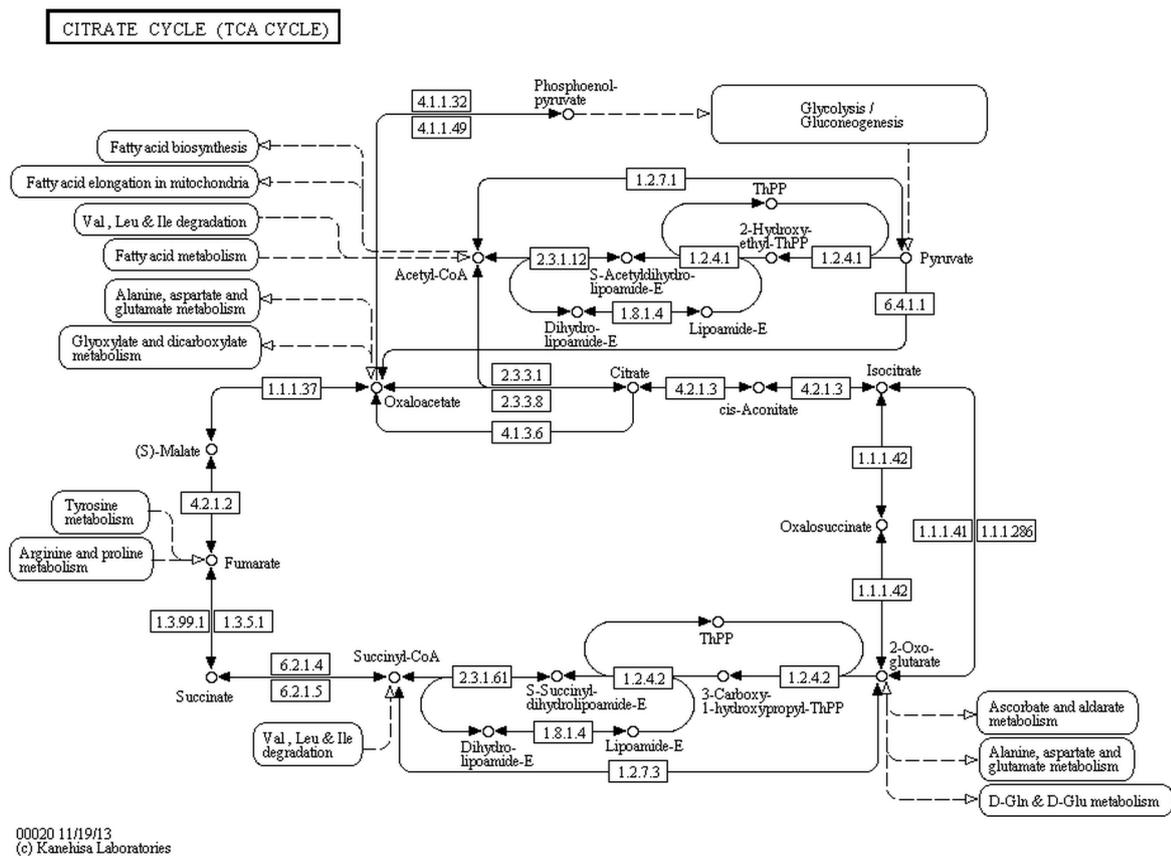


Abbildung A.3: Referenzbild des Zitratzyklus aus der KEGG-Datenbank.

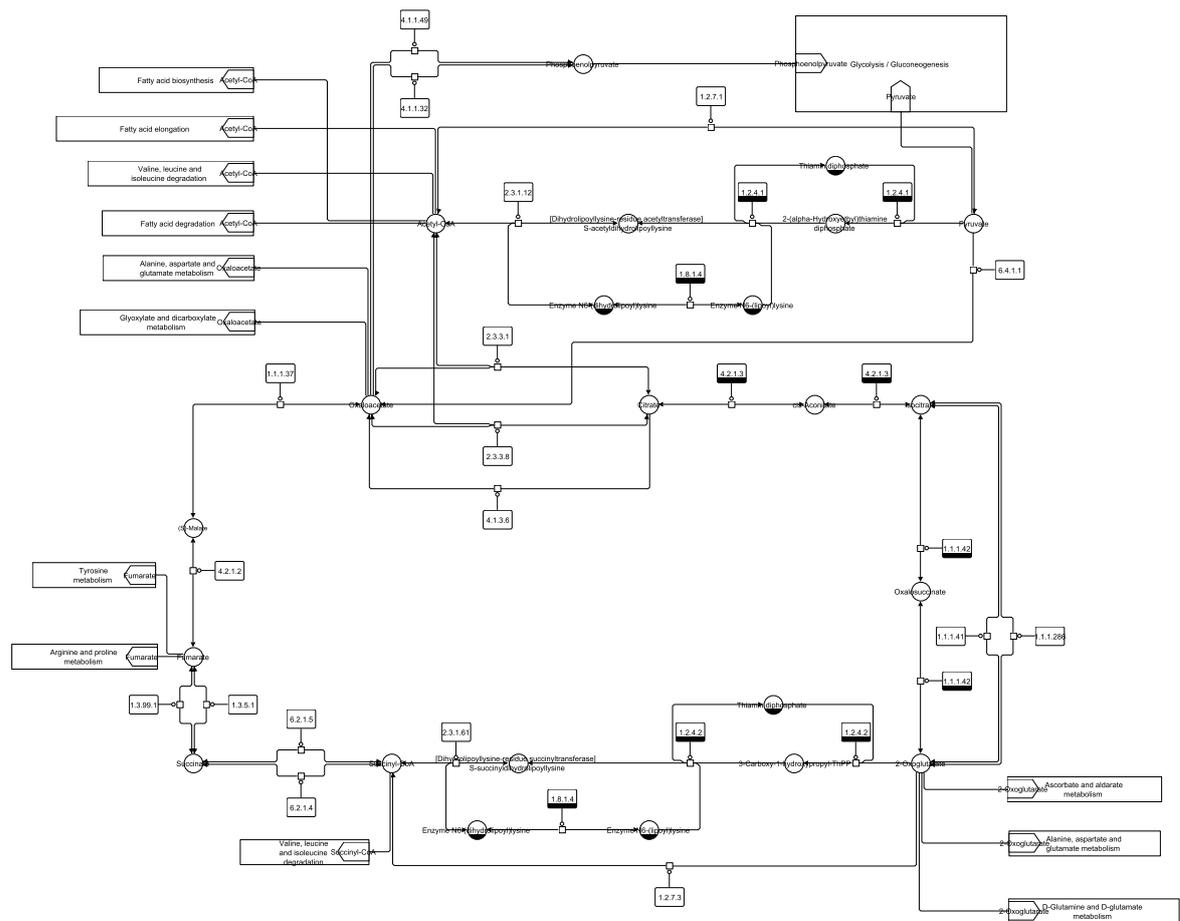
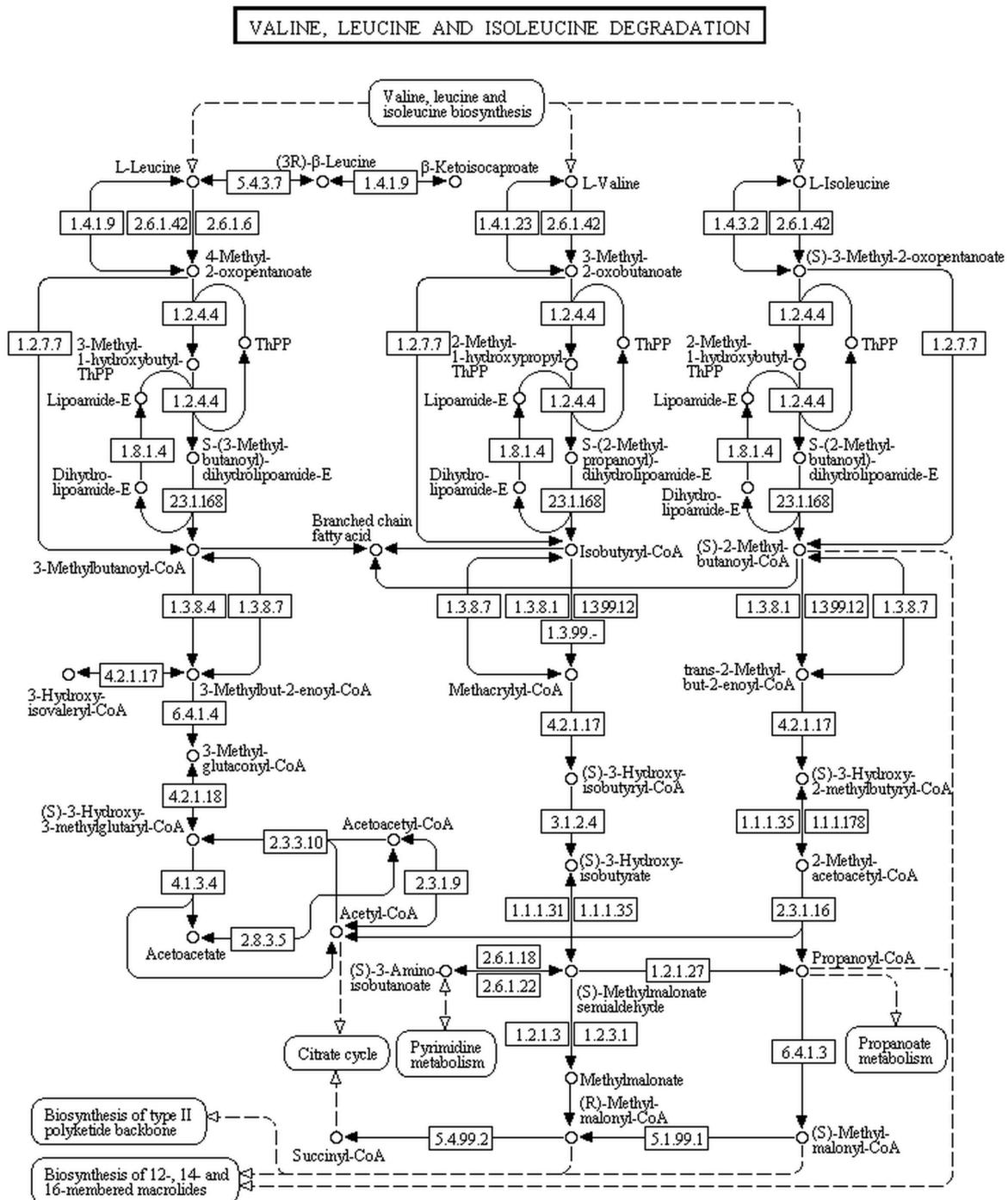


Abbildung A.4: Der Zitratzyklus aus der KEGG-Datenbank wurde basierend auf der KGML-Datei in eine SBGN PD-Karte übersetzt, anschließend wurde der beschriebene Layout-Algorithmus angewendet (siehe Kapitel 4).



00280 9/12/12
(c) Kanehisa Laboratories

Abbildung A.5: Referenzbild des Valin, Leucin und Isoleucin-Abbaus aus der KEGG-Datenbank.

Anhang B

Abbildungsvorschriften für die Umwandlung von SBGN *Process Description*-Karten in SBGN *Activity Flow*-Karten

Als Ergänzung zu den Umwandlungsregeln in den Tabellen 5.1 bis 5.6 sind auf den folgenden Seiten Abbildungsvorschriften für die Umwandlung von SBGN *Process Description*-Karten in SBGN *Activity Flow*-Karten zusammengefasst, dazu werden eine SBGN PD-Karte als Graph $G = (V, E)$ mit Knoten V und Kanten E und eine SBGN AF-Karte als Graph $G^* = (V^*, E^*)$ mit Knoten V^* und Kanten E^* betrachtet.

Regeln $c_1 - c_8$ für die Umwandlung gemeinsamer Elemente der *Process Description*- und *Activity Flow*-Sprache (siehe auch Tabelle 5.1)

$$\begin{aligned}
 c_1 : V_{CP} & \rightarrow V_{CP}^* \\
 c_2 : V_{PH} & \rightarrow V_{PH}^* \\
 c_3 : V_{PA} & \rightarrow V_{PE}^* \\
 c_4 : V_{SU} \times (\mathcal{P}(V_{TE}) \setminus \emptyset) & \rightarrow V_{SU}^* \times (\mathcal{P}(V_{TE}^*) \setminus \emptyset) \\
 c_5 : V_{TA} & \rightarrow V_{TA}^* \\
 c_6 : V_{LO} & \rightarrow V_{LO}^* \\
 c_7 : E_{LO} & \rightarrow E_{LO}^* \\
 c_8 : E_{EQ} & \rightarrow E_{EQ}^*
 \end{aligned}$$

- V_{CP} Menge der *Compartment*-Knoten der PD-Karte
- V_{CP}^* Menge der *Compartment*-Knoten der AF-Karte
- V_{PH} Menge der *Phenotype*-Knoten der PD-Karte
- V_{PH}^* Menge der *Phenotype*-Knoten der AF-Karte
- V_{PA} Menge der *Perturbing Agent*-Knoten der PD-Karte
- V_{PE}^* Menge der *Perturbation*-Knoten der AF-Karte
- V_{SU} Menge der *Submap*-Knoten der PD-Karte
- V_{SU}^* Menge der *Submap*-Knoten der AF-Karte
- V_{TE} Menge der *Terminal*-Knoten der PD-Karte
- V_{TE}^* Menge der *Terminal*-Knoten der AF-Karte
- V_{TA} Menge der *Tag*-Knoten der PD-Karte
- V_{TA}^* Menge der *Tag*-Knoten der AF-Karte
- V_{LO} Menge der *Logical Operator*-Knoten der PD-Karte
- V_{LO}^* Menge der *Logical Operator*-Knoten der AF-Karte
- E_{LO} Menge der *Logic Arc*-Kanten der PD-Karte
- E_{LO}^* Menge der *Logic Arc*-Kanten der AF-Karte
- E_{EQ} Menge der *Equivalence Arc*-Kanten der PD-Karte
- E_{EQ}^* Menge der *Equivalence Arc*-Kanten der AF-Karte

Regeln $n_1 - n_6$ für die Umwandlung von *Entity Pool Node*-Knoten (siehe auch Tabelle 5.2)

$$V_{EPN} = V_{SC} \cup V_{MSC} \cup V_{MA} \cup V_{MMA} \cup V_{NU} \cup V_{MNU} \cup \\ V_{CO} \cup V_{MCO} \cup V_{UN}$$

$$n_1 - n_5 : V_{EPN} \rightarrow V_{BI}^*$$

$$n_6 : V_{SO} \rightarrow \emptyset$$

- V_{SC} Menge der *Simple Chemical*-Knoten der PD-Karte
- V_{MSC} Menge der *Multimer Simple Chemical*-Knoten der PD-Karte
- V_{MA} Menge der *Macromolecule*-Knoten der PD-Karte
- V_{MMA} Menge der *Multimer Macromolecule*-Knoten der PD-Karte
- V_{NU} Menge der *Nucleic Acid Feature*-Knoten der PD-Karte
- V_{MNU} Menge der *Multimer Nucleic Acid Feature*-Knoten der PD-Karte
- V_{CO} Menge der *Complex*-Knoten der PD-Karte
- V_{MCO} Menge der *Multimer Complex*-Knoten der PD-Karte
- V_{UN} Menge der *Unspecified Entity*-Knoten der PD-Karte
- V_{SO} Menge der *Source*- und *Sink*-Knoten der PD-Karte
- V_{BI}^* Menge der *Biological Activity*-Knoten der AF-Karte

Regeln a₁ – a₂ für die Umwandlung von Knoten mit gleichem Knotennamen aber unterschiedlichen *Auxiliary Units* (siehe auch Tabelle 5.3)

$$\begin{aligned}
 T_{AU} &= \{(v_1, \dots, v_k) \mid \text{Knotenname}(v_i) = \text{Knotenname}(v_j) \text{ für } i, j = 1, \dots, k\} \\
 a_1 : T_{AU} &\rightarrow V_{BI}^* \quad \text{mit} \quad |V_{BI}^*| = |T_{AU}| \\
 a_2 : T_{BI}^* &= \{(v_1^*, \dots, v_k^*) \mid \text{Knotenname}(v_i^*) \neq \text{Knotenname}(v_j^*) \text{ für } i, j = 1, \dots, k\} \\
 T_{AU} &\rightarrow T_{BI}^* \quad \text{mit} \quad |T_{BI}^*| = |T_{AU}|
 \end{aligned}$$

T_{AU} Menge von Tupeln, jedes Tupel enthält *Entity Pool Node*-Knoten der PD-Karte mit identischen Knotennamen

V_{BI}^* Menge der *Biological Activity*-Knoten der AF-Karte

T_{BI}^* Menge von Tupeln, jedes Tupel enthält *Biological Activity*-Knoten der AF-Karte mit unterschiedlichen Knotennamen

Regeln co₁ – co₂ für die Umwandlung von *Complex*-Knoten (siehe auch Tabelle 5.4)

Die Regeln co_1 – co_2 werden implizit in Regel n_4 angewendet.

Regeln cl₁ – cl₃ für die Umwandlung von Knoten mit einem *Clone Marker* (siehe auch Tabelle 5.5)

$$\begin{aligned}
 T_{CL} &= \{(v_1, \dots, v_k) \mid v_i = v_j \text{ für } i, j = 1, \dots, k\} \\
 cl_1 : T_{CL} &\rightarrow V_{BI}^* \quad \text{mit} \quad |V_{BI}^*| = |T_{CL}| \\
 cl_2 : T_{BI}^* &= \{(v_1^*, \dots, v_k^*) \mid v_i^* \neq v_j^* \text{ für } i, j = 1, \dots, k\} \\
 T_{CL} &\rightarrow T_{BI}^* \quad \text{mit} \quad |T_{BI}^*| = |T_{CL}| \\
 cl_3 : T_{CL} &\rightarrow \emptyset
 \end{aligned}$$

T_{CL} Menge von Tupeln, jedes Tupel enthält identische Knoten der PD-Karte (Knoten der PD-Karte mit einem *Clone Marker*)

V_{BI}^* Menge der *Biological Activity*-Knoten der AF-Karte

T_{BI}^* Menge von Tupeln, jedes Tupel enthält unterschiedliche *Biological Activity*-Knoten der AF-Karte

Regeln $p_1 - p_6$ für die Umwandlung von Kanten (siehe auch Tabelle 5.6)

$$p_1 : T = \{(v_A, e_{CO,1}, v_{PN,1}, e_{PR,2}, v_B) \mid e_{CO,1} = (v_A, v_{PN,1}), e_{PR,2} = (v_{PN,1}, v_B)\}$$

$$T^* = \{(v_A^*, e_{MA,1}^*, v_B^*) \mid e_{MA,1}^* = (v_A^*, v_B^*)\}$$

$$T \rightarrow T^*$$

$$p_2 : T = \{(v_A, e_{PR,1}, v_{PN,1}, e_{PR,2}, v_B) \mid e_{PR,1} = (v_{PN,1}, v_A), e_{PR,2} = (v_{PN,1}, v_B)\}$$

$$T^* = \{(v_A^*, e_{MA,1}^*, e_{MA,2}^*, v_B^*) \mid e_{MA,1}^* = (v_A^*, v_B^*), e_{MA,2}^* = (v_B^*, v_A^*)\}$$

$$T \rightarrow T^*$$

$$p_3 : T = \{(v_D, e_{MA,3}, v_{PN,1}, e_{PR,2}, v_B) \mid e_{MA,3} = (v_D, v_{PN,1}), e_{PR,2} = (v_{PN,1}, v_B)\}$$

$$T^* = \{(v_D^*, e_{MA,1}^*, v_B^*) \mid e_{MA,1}^* = (v_D^*, v_B^*)\}$$

$$T \rightarrow T^*$$

$$p_4 : T = \{(v_A, e_{CO,1}, v_D, e_{MA,3}, v_{PN,1}) \mid e_{CO,1} = (v_A, v_{PN,1}), e_{MA,3} = (v_D, v_{PN,1})\}$$

$$T^* = \{(v_A^*, e_{MA,1}^*, v_D^*) \mid e_{MA,1}^* = (v_D^*, v_A^*)\}$$

$$T \rightarrow T^*$$

$$p_5 : T = \{(v_A, e_{CO,1}, v_D, e_{MA,3}, v_{PN,1}, e_{PR,2}, v_B, e_{CO,4}, v_E, e_{MA,6}, v_{PN,2}, e_{PR,5}, v_C) \mid$$

$$e_{CO,1} = (v_A, v_{PN,1}), e_{MA,3} = (v_D, v_{PN,1}), e_{PR,2} = (v_{PN,1}, v_B),$$

$$e_{CO,4} = (v_B, v_{PN,2}), e_{MA,6} = (v_E, v_{PN,2}), e_{PR,5} = (v_{PN,2}, v_C)\}$$

$$T^* = \{(v_D^*, e_{MA,1}^*, v_E^*) \mid e_{MA,1}^* = (v_D^*, v_E^*)\}$$

$$T \rightarrow T^*$$

$$p_6 : T = \{(v_A, e_{CO,1}, v_D, e_{CS,3}, v_{PN,1}, e_{PR,2}, v_B) \mid$$

$$e_{CO,1} = (v_A, v_{PN,1}), e_{CS,3} = (v_D, v_{PN,1}), e_{PR,2} = (v_{PN,1}, v_B)\}$$

$$T^* = \{(v_A^*, e_{LO,1}^*, v_D^*, e_{LO,2}^*, v_{AND}^*, e_{PO,3}^*, v_B^*) \mid$$

$$e_{LO,1}^* = (v_A^*, v_{AND}^*), e_{LO,2}^* = (v_D^*, v_{AND}^*), e_{PO,3}^* = (v_{AND}^*, v_B^*)\}$$

$$T \rightarrow T^*$$

T Tupel von Knoten und Kanten der PD-Karte

T^* Tupel von Knoten und Kanten der AF-Karte

$$\begin{aligned}
 v_A, v_B, v_C, v_D, v_E &\in V_{EPN} \\
 v_{PN,1}, v_{PN,2} &\in V_{PN} \\
 e_{CO,1}, e_{CO,4} &\in E_{CO} \\
 e_{PR,1}, e_{PR,2}, e_{PR,5} &\in E_{PR} \\
 e_{MA,3}, e_{MA,6} &\in E_{MA} \\
 e_{CS,3} \in E_{CA} \vee e_{CS,3} &\in E_{ST} \\
 v_A^*, v_B^*, v_D^*, v_E^* &\in V_{BI}^* \\
 v_{AND}^* &\in V_{LO}^* \\
 e_{MA,1}^*, e_{MA,2}^* &\in E_{MA}^* \\
 e_{PO,3}^* &\in E_{PO}^* \\
 e_{LO,1}^*, e_{LO,2}^* &\in E_{LO}^*
 \end{aligned}$$

$$\begin{aligned}
 V_{EPN} &= V_{SC} \cup V_{MSC} \cup V_{MA} \cup V_{MMA} \cup V_{NU} \cup V_{MNU} \cup \\
 &\quad V_{CO} \cup V_{MCO} \cup V_{UN} \cup V_{PA} \\
 V_{PN} &= V_{PR} \cup V_{OM} \cup V_{UC} \cup V_{AS} \cup V_{DI} \\
 E_{MA} &= E_{MO} \cup E_{ST} \cup E_{CA} \cup E_{IN} \cup E_{NE} \\
 E_{MA}^* &= E_{PO}^* \cup E_{NG}^* \cup E_{UN}^* \cup E_{NE}^*
 \end{aligned}$$

V_{SC}	Menge der <i>Simple Chemical</i> -Knoten der PD-Karte
V_{MSC}	Menge der <i>Multimer Simple Chemical</i> -Knoten der PD-Karte
V_{MA}	Menge der <i>Macromolecule</i> -Knoten der PD-Karte
V_{MMA}	Menge der <i>Multimer Macromolecule</i> -Knoten der PD-Karte
V_{NU}	Menge der <i>Nucleic Acid Feature</i> -Knoten der PD-Karte
V_{MNU}	Menge der <i>Multimer Nucleic Acid Feature</i> -Knoten der PD-Karte
V_{CO}	Menge der <i>Complex</i> -Knoten der PD-Karte
V_{MCO}	Menge der <i>Multimer Complex</i> -Knoten der PD-Karte
V_{UN}	Menge der <i>Unspecified Entity</i> -Knoten der PD-Karte
V_{PA}	Menge der <i>Perturbing Agent</i> -Knoten der PD-Karte
V_{PR}	Menge der <i>Process</i> -Knoten der PD-Karte
V_{OM}	Menge der <i>Omitted Process</i> -Knoten der PD-Karte
V_{UC}	Menge der <i>Uncertain Process</i> -Knoten der PD-Karte
V_{AS}	Menge der <i>Association</i> -Knoten der PD-Karte
V_{DI}	Menge der <i>Dissociation</i> -Knoten der PD-Karte
V_{BI}^*	Menge der <i>Biological Activity</i> -Knoten der AF-Karte
V_{LO}^*	Menge der <i>Logical Operator</i> -Knoten der AF-Karte

E_{CO}	Menge der <i>Consumption</i> -Kanten der PD-Karte
E_{PR}	Menge der <i>Production</i> -Kanten der PD-Karte
E_{MO}	Menge der <i>Modulation</i> -Kanten der PD-Karte
E_{ST}	Menge der <i>Stimulation</i> -Kanten der PD-Karte
E_{CA}	Menge der <i>Catalysis</i> -Kanten der PD-Karte
E_{IN}	Menge der <i>Inhibition</i> -Kanten der PD-Karte
E_{NE}	Menge der <i>Necessary Stimulation</i> -Kanten der PD-Karte
E_{PO}^*	Menge der <i>Positive Influence</i> -Kanten der AF-Karte
E_{NG}^*	Menge der <i>Negative Influence</i> -Kanten der AF-Karte
E_{UN}^*	Menge der <i>Unknown Influence</i> -Kanten der AF-Karte
E_{NE}^*	Menge der <i>Necessary Stimulation</i> -Kanten der AF-Karte
E_{LO}^*	Menge der <i>Logic Arc</i> -Kanten der AF-Karte

Anhang C

Beispiele für die Umwandlung von SBGN *Process Description*-Karten in SBGN *Activity Flow*-Karten

In Abschnitt 5.4 wurden bereits die Umwandlungen der PD-Karte eines Signalweges (MAPK-Kaskade) in zwei AF-Karten als Beispiele gezeigt. Hier sollen nun als weitere Beispiele die Umwandlungen der PD-Karte eines Stoffwechselweges (Glykolyse) in AF-Karten unter Anwendung der vier Templates (siehe auch Abschnitt 5.2) dargestellt werden (Abbildung C.1 und C.2).

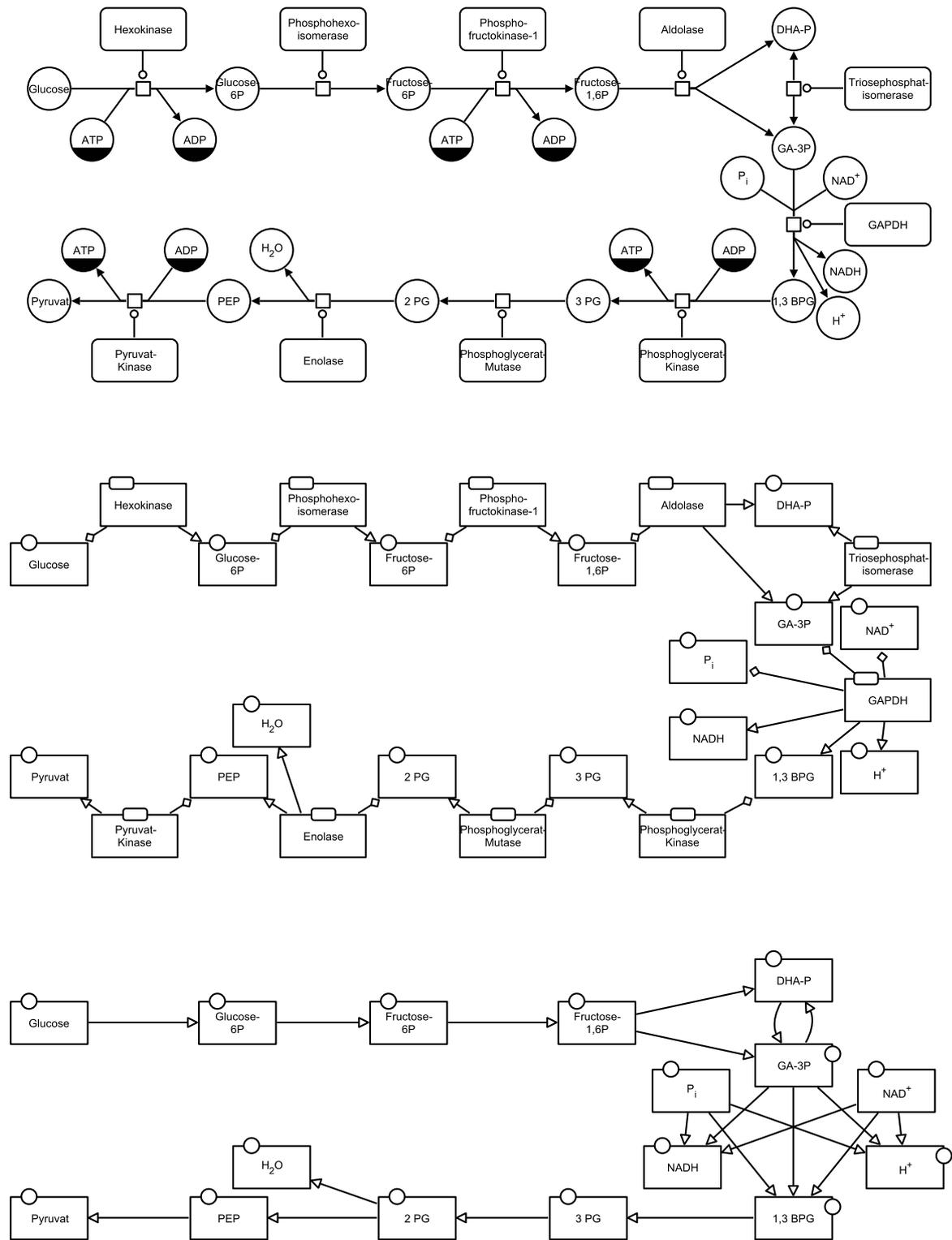


Abbildung C.1: Umwandlung einer PD-Karte in eine AF-Karte am Beispiel der Glykolyse [102], übernommen und angepasst aus [116]. Oben: die originale PD-Karte, Mitte: die übersetzte AF-Karte unter Anwendung des Templates „Enzym-Aktivitäten“, unten: die übersetzte AF-Karte unter Anwendung des Templates „Metabolit-Aktivitäten“.

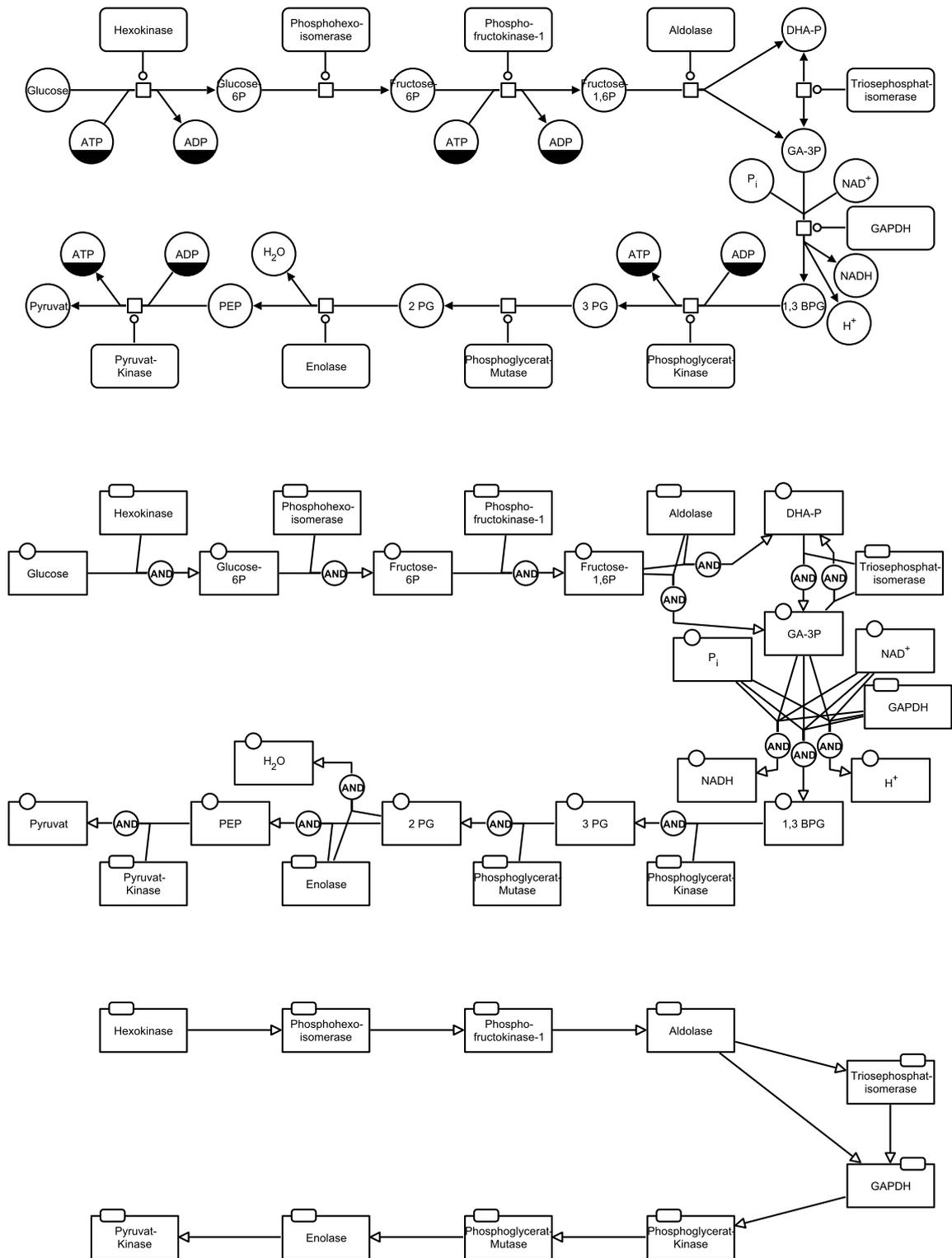


Abbildung C.2: Umwandlung einer PD-Karte in eine AF-Karte am Beispiel der Glykolyse [102], übernommen und angepasst aus [116]. Oben: die originale PD-Karte, Mitte: die übersetzte AF-Karte unter Anwendung des Templates „Enzym- und Metabolit-Aktivitäten“, unten: die übersetzte AF-Karte unter Anwendung des Templates „Mittelbare Enzym-Aktivitäten“.

Anhang D

Abbildungen der SBGN Bricks

In Tabelle 6.3 sind bereits Abbildungen der SBGN Bricks für Genregulationsnetzwerke gezeigt. Eine Übersicht über alle SBGN Bricks ist in den folgenden Tabellen zu finden. Es sind die SBGN Bricks für Stoffwechselnetzwerke (Tabelle D.1), Signalnetzwerke (Tabelle D.2), Protein-Protein-Interaktionsnetzwerke (Tabelle D.3), und Genregulationsnetzwerke (Tabelle D.4) sowie sonstige SBGN Bricks (Tabellen D.5 und D.6) dargestellt.

Prozess	Brick in PD	Brick in ER	Brick in AF
Katalyse GO:0008152 GO:0044238 GO:0003824		—	
irreversibel 1 Edukt 1 Produkt		—	
irreversibel 2 Edukte 1 Produkt		—	
irreversibel 1 Edukt 2 Produkte		—	—
irreversibel 2 Edukte 2 Produkte		—	—
reversibel 1 Edukt 1 Produkt		—	—
reversibel 2 Edukte 1 Produkt		—	—
reversibel 1 Edukt 2 Produkte		—	—
reversibel 2 Edukte 2 Produkte		—	—

Tabelle D.1: SBGN Bricks für Stoffwechselnetzwerke.

Prozess	Brick in PD	Brick in ER	Brick in AF
Protein- Phosphorylierung GO:0006468 GO:0003824			

Tabelle D.2: SBGN Bricks für Signalnetzwerke.

Prozess	Brick in PD	Brick in ER	Brick in AF
Komplexbildung GO:0006461			
Komplexdissoziation GO:0043241			
Oligomerisierung Homodimerisierung GO:0035786			

Tabelle D.3: SBGN Bricks für Protein-Protein-Interaktionsnetzwerke.

Prozess	Brick in PD	Brick in ER	Brick in AF
Transkription mit Bindung GO:0006351 GO:0006355 GO:0009299 GO:0065004			
Transkription ohne Bindung GO:0006351 GO:0006355 GO:0009299			
Translation GO:0006412			

Tabelle D.4: SBGN Bricks für Genregulationsnetzwerke (siehe auch Abschnitt 6.1.1).

	Brick in PD	Brick in ER	Brick in AF
Funktionelle Beziehung GO:0008150	-	-	

Tabelle D.5: Sonstige SBGN Bricks.

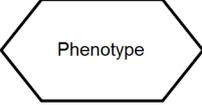
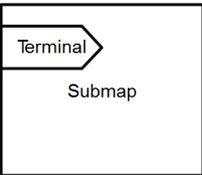
Prozess	Brick in PD	Brick in ER	Brick in AF
Transport GO:0006810			
passiv (Diffusion)			

Tabelle D.6: Sonstige SBN Bricks (fortgesetzt von Tabelle D.5).

Anhang E

Übersicht über SBGN-ML

Als Ergänzung zum Beispiel in Abbildung 7.1 ist in den folgenden Tabellen in Anlehnung an die SBGN-Referenzkarten (siehe Abbildungen 3.2, 3.3 und 3.4) eine Übersicht über SBGN-ML Elemente zu finden. In Tabelle E.1 sind allgemeine SBGN-ML Elemente gezeigt, die für mehr als eine der drei SBGN-Sprachen genutzt werden, Tabelle E.2 zeigt spezifische SBGN-ML Elemente für die *Process Description*-Sprache, Tabelle E.3 zeigt spezifische SBGN-ML Elemente für die *Entity Relationship*-Sprache und Tabelle E.4 zeigt spezifische SBGN-ML Elemente für die *Activity Flow*-Sprache.

SBGN	SBGN-ML
	<pre><glyph id="glyph1" class="phenotype"> <label text="Phenotype"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph2" class="and"> <bbox x="60.0" y="20.0" h="40.0" w="40.0"/> <port id="glyph2.1" x="50.0" y="40.0"/> <port id="glyph2.2" x="110.0" y="40.0"/> </glyph></pre>
	<pre><glyph id="glyph3" class="or"> <bbox x="60.0" y="20.0" h="40.0" w="40.0"/> <port id="glyph3.1" x="50.0" y="40.0"/> <port id="glyph3.2" x="110.0" y="40.0"/> </glyph></pre>
	<pre><glyph id="glyph4" class="not"> <bbox x="60.0" y="20.0" h="40.0" w="40.0"/> <port id="glyph4.1" x="50.0" y="40.0"/> <port id="glyph4.2" x="110.0" y="40.0"/> </glyph></pre>
	<pre><glyph id="glyph5" class="delay"> <bbox x="60.0" y="20.0" h="40.0" w="40.0"/> <port id="glyph5.1" x="50.0" y="40.0"/> <port id="glyph5.2" x="110.0" y="40.0"/> </glyph></pre>
	<pre><glyph id="glyph6" class="compartment"> <label text="Compartment"> <bbox x="36.5" y="28.0" h="16.0" w="87.0"/> </label> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph7" class="submap"> <label text="Submap"/> <bbox x="10.0" y="20.0" h="120.0" w="140.0"/> <glyph id="glyph8" orientation="right" class="terminal"> <label text="Terminal"/> <bbox x="10.0" y="35.0" h="30.0" w="70.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph9" orientation="right" class="tag"> <label text="Tag"/> <bbox x="30.0" y="20.0" h="60.0" w="100.0"/> </glyph></pre>

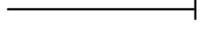
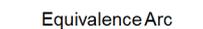
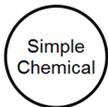
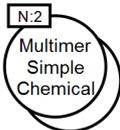
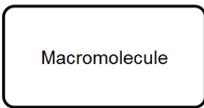
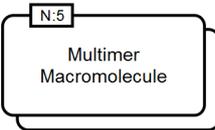
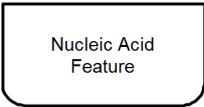
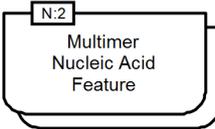
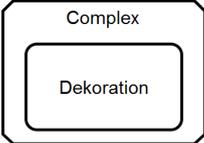
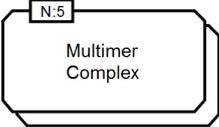
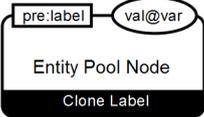
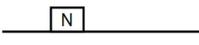
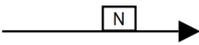
 Modulation	<pre data-bbox="555 237 1086 427"><arc id="arc1" class="modulation" source="glyph10" target="glyph11"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>
 Stimulation	<pre data-bbox="555 434 1086 624"><arc id="arc2" class="stimulation" source="glyph12" target="glyph13"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>
 Inhibition	<pre data-bbox="555 631 1086 822"><arc id="arc3" class="inhibition" source="glyph14" target="glyph15"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>
 Necessary Stimulation	<pre data-bbox="555 826 1171 1016"><arc id="arc4" class="necessary stimulation" source="glyph16" target="glyph17"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>
 Logic Arc	<pre data-bbox="555 1021 1086 1211"><arc id="arc5" class="logic arc" source="glyph18" target="glyph19"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>
 Equivalence Arc	<pre data-bbox="555 1216 1086 1382"><arc id="arc6" class="equivalence arc" source="glyph20" target="glyph21"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc></pre>

Tabelle E.1: Allgemeine SBGN-ML Elemente, IDs, Namen, Positionen und Größen sind beispielhaft gewählt.

SBGN	SBGN-ML
	<pre><glyph id="glyph1" class="unspecified entity"> <label text="Unspecified&#xA;Entity"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph2" class="simple chemical"> <label text="Simple&#xA;Chemical"/> <bbox x="45.0" y="20.0" h="70.0" w="70.0"/> </glyph></pre>
	<pre><glyph id="glyph3" class="simple chemical multimer"> <label text="Multimer&#xA;Simple&#xA;Chemical"/> <bbox x="45.0" y="55.0" h="70.0" w="70.0"/> <glyph id="glyph3a" class="unit of information"> <label text="N:2"/> <bbox x="47.0" y="46.5" h="17.0" w="31.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph4" class="macromolecule"> <label text="Macromolecule"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph5" class="macromolecule multimer"> <label text="Multimer&#xA;Macromolecule"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> <glyph id="glyph5a" class="unit of information"> <label text="N:5"/> <bbox x="29.5" y="11.5" h="17.0" w="31.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph6" class="nucleic acid feature"> <label text="Nucleic Acid&#xA;Feature"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph7" class="nucleic acid feature multimer"> <label text="Multimer&#xA;Nucleic Acid&#xA;Feature"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> <glyph id="glyph7a" class="unit of information"> <label text="N:2"/> <bbox x="29.5" y="11.5" h="17.0" w="31.0"/> </glyph> </glyph></pre>

	<pre> <glyph id="glyph8" class="complex"> <label text="Complex"> <bbox x="50.0" y="22.0" h="16.0" w="60.0"/> </label> <bbox x="10.0" y="20.0" h="100.0" w="140.0"/> <glyph id="glyph9" class="macromolecule"> <label text="Dekoration"/> <bbox x="26.0" y="50.0" h="60.0" w="108.0"/> </glyph> </glyph> </pre>
	<pre> <glyph id="glyph10" class="complex multimer"> <label text="Multimer&#xA;Complex"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> <glyph id="glyph10a" class="unit of information"> <label text="N:5"/> <bbox x="29.5" y="11.5" h="17.0" w="31.0"/> </glyph> </glyph> </pre>
	<pre> <glyph id="glyph11" class="macromolecule"> <label text="Entity Pool Node"/> <clone> <label text="Clone Label"> <bbox x="47.5" y="73.0" h="15.0" w="65.0"/> </label> </clone> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> <glyph id="glyph11a" class="unit of information"> <label text="pre:label"/> <bbox x="17.0" y="11.5" h="17.0" w="56.0"/> </glyph> <glyph id="glyph11b" class="state variable"> <state variable="var" value="val"/> <bbox x="85.0" y="6.5" h="27.0" w="60.0"/> </glyph> </glyph> </pre>
	<pre> <glyph id="glyph12" class="perturbing agent"> <label text="Perturbing&#xA;Agent"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph> </pre>
	<pre> <glyph id="glyph13" class="source and sink"> <bbox x="55.0" y="20.0" h="50.0" w="50.0"/> </glyph> </pre>

 Process	<pre> <glyph id="glyph14" class="process"> <bbox x="68.0" y="20.0" h="24.0" w="24.0"/> <port id="glyph14.1" x="58.0" y="32.0"/> <port id="glyph14.2" x="102.0" y="32.0"/> </glyph> </pre>
 Omitted Process	<pre> <glyph id="glyph15" class="omitted process"> <bbox x="68.0" y="20.0" h="24.0" w="24.0"/> <port id="glyph15.1" x="58.0" y="32.0"/> <port id="glyph15.2" x="102.0" y="32.0"/> </glyph> </pre>
 Uncertain Process	<pre> <glyph id="glyph16" class="uncertain process"> <bbox x="68.0" y="20.0" h="24.0" w="24.0"/> <port id="glyph16.1" x="58.0" y="32.0"/> <port id="glyph16.2" x="102.0" y="32.0"/> </glyph> </pre>
 Association	<pre> <glyph id="glyph17" class="association"> <bbox x="68.0" y="20.0" h="24.0" w="24.0"/> <port id="glyph17.1" x="58.0" y="32.0"/> <port id="glyph17.2" x="102.0" y="32.0"/> </glyph> </pre>
 Dissociation	<pre> <glyph id="glyph18" class="dissociation"> <bbox x="68.0" y="20.0" h="24.0" w="24.0"/> <port id="glyph18.1" x="58.0" y="32.0"/> <port id="glyph18.2" x="102.0" y="32.0"/> </glyph> </pre>
 Consumption	<pre> <arc id="arc1" class="consumption" source="glyph19" target="glyph20.1"> <glyph id="glyph21" class="cardinality"> <label text="N"/> <bbox y="1.0" x="42.0" h="18.5" w="24.5"/> </glyph> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>
 Production	<pre> <arc id="arc2" class="production" source="glyph22.2" target="glyph23"> <glyph id="glyph24" class="cardinality"> <label text="N"/> <bbox y="1.0" x="78.0" h="18.5" w="24.5"/> </glyph> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>

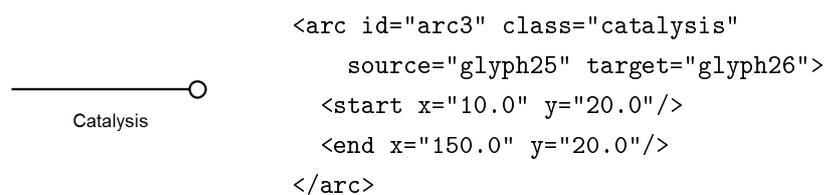
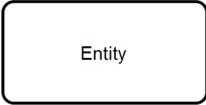
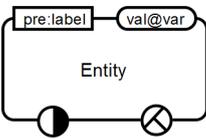
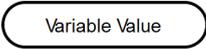
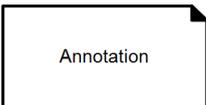
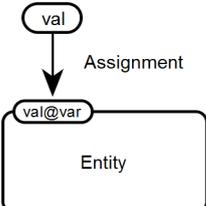


Tabelle E.2: Spezifische SBGN-ML Elemente für die *Process Description*-Sprache, IDs, Namen, Positionen und Größen sind beispielhaft gewählt.

SBGN	SBGN-ML
	<pre><glyph id="glyph1" class="entity"> <label text="Entity"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph2" class="entity"> <label text="Entity"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> <glyph id="glyph2a" class="unit of information"> <label text="pre:label"/> <bbox x="17.0" y="11.5" h="17.0" w="56.0"/> </glyph> <glyph id="glyph2b" class="state variable"> <state variable="var" value="val"/> <bbox x="87.0" y="11.5" h="17.0" w="56.0"/> </glyph> <glyph id="glyph2c" class="existence"> <bbox x="34.0" y="77.5" h="25.0" w="25.0"/> </glyph> <glyph id="glyph2d" class="location"> <bbox x="104.0" y="77.5" h="25.0" w="25.0"/> </glyph> </glyph></pre>
	<pre><arc id="arc1" class="interaction" source="glyph3" target="glyph4"> <glyph id="arc1.1" class="outcome"> <bbox x="70.0" y="80.0" h="20.0" w="20.0"/> </glyph> <start x="80.0" y="20.0"/> <end x="80.0" y="160.0"/> </arc></pre>
	<pre><glyph id="glyph5" class="variable value"> <label text="Variable Value"/> <bbox x="10.0" y="20.0" h="30.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph6" class="annotation"> <label text="Annotation"/> <bbox x="10.0" y="20.0" h="70.0" w="140.0"/> </glyph></pre>
	<pre><arc id="arc2" class="assignment" source="glyph7" target="glyph8"> <start x="45.0" y="20.0"/> <end x="45.0" y="70.0"/> </arc></pre>

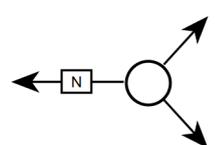
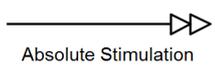
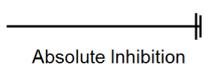
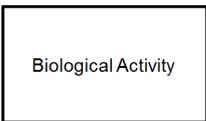
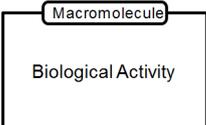
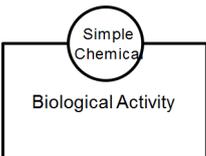
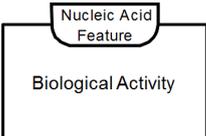
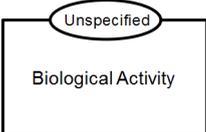
	<pre> <arcgroup class="interaction"> <glyph id="glyph9" class="interaction"> <bbox x="92.5" y="50.0" h="30.0" w="30.0"/> </glyph> <arc id="arc3" class="interaction" source="glyph9" target="glyph10"> <start x="121.0" y="50.0"/> <end x="150.25" y="17.5"/> </arc> <arc id="arc4" class="interaction" source="glyph9" target="glyph11"> <start x="121.0" y="80.0"/> <end x="150.25" y="112.5"/> </arc> <arc id="arc5" class="interaction" source="glyph9" target="glyph12"> <glyph id="glyph13" class="cardinality"> <label text="N"/> <bbox x="45.0" y="54.0" h="17.5" w="22.5"/> </glyph> <start x="92.5" y="65.0"/> <end x="10.0" y="65.0"/> </arc> </arcgroup> </pre>
 <p>Absolute Stimulation</p>	<pre> <arc id="arc6" class="absolute stimulation" source="glyph14" target="glyph15"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>
 <p>Absolute Inhibition</p>	<pre> <arc id="arc7" class="absolute inhibition" source="glyph16" target="glyph17"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>

Tabelle E.3: Spezifische SBGN-ML Elemente für die *Entity Relationship*-Sprache, IDs, Namen, Positionen und Größen sind beispielhaft gewählt.

SBGN	SBGN-ML
	<pre><glyph id="glyph1" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> </glyph></pre>
	<pre><glyph id="glyph2" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> <glyph id="glyph2a" class="unit of information"> <label text="Macromolecule"/> <entity name="macromolecule"/> <bbox x="36.5" y="11.5" h="17.0" w="87.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph3" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> <glyph id="glyph3a" class="unit of information"> <label text="Simple&#xA;Chemical"/> <entity name="simple chemical"/> <bbox x="53.5" y="13.5" h="53.0" w="53.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph4" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> <glyph id="glyph4a" class="unit of information"> <label text="Nucleic Acid&#xA;Feature"/> <entity name="nucleic acid feature"/> <bbox x="42.5" y="4.5" h="31.0" w="75.0"/> </glyph> </glyph></pre>
	<pre><glyph id="glyph5" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> <glyph id="glyph5a" class="unit of information"> <label text="Unspecified"/> <entity name="unspecified entity"/> <bbox x="41.0" y="5.5" h="29.0" w="78.0"/> </glyph> </glyph></pre>

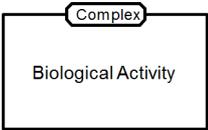
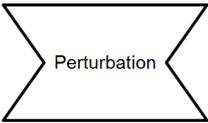
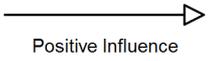
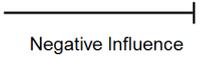
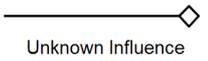
	<pre> <glyph id="glyph6" class="biological activity"> <label text="Biological Activity"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> <glyph id="glyph6a" class="unit of information"> <label text="Complex"/> <entity name="complex"/> <bbox x="52.0" y="11.5" h="17.0" w="56.0"/> </glyph> </glyph> </pre>
	<pre> <glyph id="glyph7" class="perturbation"> <label text="Perturbation"/> <bbox x="10.0" y="20.0" h="80.0" w="140.0"/> </glyph> </pre>
	<pre> <arc id="arc1" class="positive influence" source="glyph8" target="glyph9"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>
	<pre> <arc id="arc2" class="negative influence" source="glyph10" target="glyph11"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>
	<pre> <arc id="arc3" class="unknown influence" source="glyph12" target="glyph13"> <start x="10.0" y="20.0"/> <end x="150.0" y="20.0"/> </arc> </pre>

Tabelle E.4: Spezifische SBGN-ML Elemente für die *Activity Flow*-Sprache, IDs, Namen, Positionen und Größen sind beispielhaft gewählt.

Erklärung

Ich erkläre an Eides statt, dass ich die Arbeit selbstständig und ohne fremde Hilfe verfasst, keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ich erkläre weiterhin, dass ich keine vergeblichen Promotionsversuche unternommen habe und die Dissertation in der gegenwärtigen oder einer anderen Fassung keiner anderen Fakultät vorgelegen hat.

Tobias Czauderna

Lebenslauf

Persönliche Daten

Name Tobias Czauderna
Geburtsdatum und -ort 08.12.1977 in Magdeburg
Staatsangehörigkeit deutsch

Schulische Ausbildung

bis 06/1996 Werner-von-Siemens Gymnasium Magdeburg
Abschluss mit Abitur, Gesamtnote *1,7*

Zivildienst

10/1996–10/1997 Walter-Friedrich-Krankenhaus Magdeburg

Studium

10/1997–01/2003 Studium der Elektrotechnik
Otto-von-Guericke-Universität Magdeburg
Abschluss mit Diplom, Gesamtnote *gut*
03/2002–08/2002 Studienaufenthalt
University of South Australia Adelaide, Australien
10/2010–01/2015 Promotionsstudium
Martin-Luther-Universität Halle-Wittenberg

Beruflicher Werdegang

02/1999–12/1999 studentische Hilfskraft
Fraunhofer-Institut für Fabrikbetrieb und -automatisierung Magdeburg
01/2000–06/2001 studentische Hilfskraft
Otto-von-Guericke-Universität Magdeburg
01/2002–02/2002 studentische Hilfskraft
Otto-von-Guericke-Universität Magdeburg
02/2003–07/2014 wissenschaftlicher Mitarbeiter
Leibniz-Institut für Pflanzengenetik und Kulturpflanzenforschung (IPK)
Stadt Seeland OT Gatersleben
seit 08/2014 wissenschaftlicher Mitarbeiter
Monash University Melbourne, Australien

