Bond Graph Methodology

Wolfgang Borutzky

# Bond Graph Methodology

Development and Analysis of Multidisciplinary Dynamic System Models



Prof. Dr. Ing. Wolfgang Borutzky Bonn-Rhein-Sieg University of Applied Sciences 53754 Sankt Augustin Germany

ISBN 978-1-84882-881-0 e-ISBN 978-1-84882-882-7 DOI 10.1007/978-1-84882-882-7 Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009939277

© Springer-Verlag London Limited 2010

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The original edition of this book was published by SCS Publishing House as *Bond Graphs:* A *Methodology for Modelling Multidisciplinary Dynamic Systems.* © SCS Publishing House, 2004

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: eStudioCalamar, Figueres/Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my wife Heidrun and to my son Lars

### Preface

Nowadays, engineering systems are of ever-increasing complexity and must be considered as *multidisciplinary* systems composed of interacting subsystems or system components from different engineering disciplines. Thus, an integration of various engineering disciplines, e.g, mechanical, electrical and control engineering in a *concurrent* design approach is required. With regard to the systematic development and analysis of system models, *interdisciplinary* computer aided methodologies are becoming more and more important.

A graphical description formalism particularly suited for multidisciplinary systems are *bond graphs* devised by Professor Henry Paynter in as early as 1959 at the Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts, USA and in use since then all over the world.

This monograph is devoted exclusively to the bond graph methodology. It gives a comprehensive, in-depth, state-of-the-art presentation including recent results scattered over research articles and dissertations and research contributions by the author to a number of topics.

The book systematically covers the fundamentals of developing bond graphs and deriving mathematical models from them, the recent developments in methodology, symbolic and numerical processing of mathematical models derived from bond graphs. Additionally it discusses modern modelling languages, the paradigm of object-oriented modelling, modern software that can be used for building and for processing of bond graph models, and provides a chapter with small case studies illustrating various applications of the methodology.

In favour of presenting topics in some reasonable depth and to keep the size of the book manageable, the book refrains from scratching the surface of too many topics. For this reason, some topics, e.g., chemical reactions or links to qualitative reasoning, are not addressed. The compilation of the material in this book and its presentation has been motivated by the author's individual experiences in research and teaching for more than two decades and has been inspired by his personal interaction with many leading personalities in this area.

This monograph addresses students, lecturers, researchers, and practicing engineers in industry who want to learn more about the potential and the state-of-the-art design of this powerful interdisciplinary graphical modelling methodology and who want to see how it can help them better understand physical processes in multienergy domain engineering systems in order to develop proper models in their respective engineering field. As bond graph modelling is based on the intuitive consideration of energy exchange between system components from various energy domains, the methodology is particularly suited for modelling and design tasks in mechatronics.

Bond Graph Methodology – Development and Analysis of Multidisciplinary Models addresses the fundamentals as well as advanced topics. It has been designed to serve readers interested in what bond graph modelling is about, readers with expertise in related areas who want to see how bond graph modelling can help them in their projects as well as members of the international community of bond graph modellers. The book can be used as a supplementary text in master's programme courses on modelling, simulation and control, as well as a guide for self-study and as a reference.

The progress made in bond graph modelling is due to many people around the world. Without their research, this monograph would not have been possible.

#### Acknowledgements

I would like to take this opportunity to thank the following colleagues of mine for their review of chapters of the manuscript. Their valuable feedback, comments and suggestions are gratefully acknowledged.

Vjekoslav Damić	University of Dubrovnik, Croatia
Geneviève Dauphin-Tanguy	École Centrale de Lille, France
José Granda	California State University Sacramento, CA, USA
Sergio Junco	Universidad Nacional de Rosario, Argentina
Loucas Louca	University of Cyprus, Cyprus
Wilfrid Marquis-Favre	Institut National des Sciences Appliquées de Lyon,
	France
David Murray-Smith	University of Glasgow, Scotland, UK
Arun Samantaray	Indian Institute of Technology, Kharagpur, India
Jean Thoma	University of Waterloo, Ontario, Canada

Furthermore, I wish to thank the team of Springer-Verlag at London, UK, for supporting this project. In particular, I would like to thank Claire Protherough for her kind help.

Last but not least, I wish to express my sincere thanks and appreciation to my wife for her support, her tolerance and her patience with me.

Sankt Augustin May 2009 Wolfgang Borutzky

### **About This Book**

This book on Bond Graph Methodology is organised into 13 chapters. Its focus is mainly on concepts, modelling approaches, techniques and software tools that support the process of bond graph based physical systems modelling.

By covering a number of advanced topics, e.g., models of variable structure, multibody systems and open thermodynamic systems, the book aims at demonstrating the true interdisciplinary potential of bond graph methodology. The discussion of concepts makes use of many examples that, for pedagogical reasons, have been kept fairly small and easy to survey. A glossary supports the use of the terminology.

The introduction briefly recalls essential features of block diagrams, signal flow graphs, network representations and the paradigm of object-oriented modelling. The intention is to begin with well known graphical model representations and modelling techniques and to outline the context in which bond graph modelling is embedded.

Chapter 2 provides the fundamentals of bond graph based physical systems modelling, allowing the reader to systematically construct a so-called non-causal bond graph from a system schematic system representation. Non-causal bond graphs reflect the structure of the system to be modelled. In the conceptual phase of developing a hierarchical (top-down, or bottom-up) model, functional relations between physical quantities, given either by equations or by look-up tables are not important in the beginning. Consequently, at a conceptual model development stage, noncausal bond graphs are used to represent physical effects and relations between them in a *qualitative* manner. At this early stage of model development, a mathematical model derived from a non-causal bond graph could take only the form of a set of differential-algebraic equations including many redundant algebraic equations due to the interconnection of system component models.

With regard to the systematic derivation of a mathematical model from a bond graph, the concept of *computational causality* is explained in Chapter 3. The so-called *Standard Causality Assignment Procedure* (SCAP), introduced by Karnopp and Rosenberg, is presented and the choice of state space variables is addressed.

After an introduction of the basic notion of a *causal path*, it is shown how an ordered set of equations can be derived manually from causal bond graphs of moderate size by following causal paths back and forth. In the simplest case, the equations obtained can be turned into a set of explicit state space equations by eliminating auxiliary algebraic variables.

Beyond the most simple case of an explicit state space form, several types of causal paths in a bond graph and their effect on the form of the mathematical model are considered in Chapter 4. It turns out that a mathematical model to be derived from a causal bond graph, in general, is a set of differential-algebraic Equations (DAE system). Therefore, important notions from the theory of DAE systems are provided, in particular the notions of the *index of nilpotency* and the *differential index* of a DAE system.

From a modelling point of view, the generation of a DAE system can be avoided by inserting small energy storage elements into the bond graph. This way, a model is slightly modified. This approach, quite common before numerically robust solvers for DAE systems of index < 2 came up, is briefly discussed. Moreover, two alternatives to the standard causality assignment procedure, viz., the *method of relaxed causalities*, introduced by Joseph and Martens, and Lagrange causalities (Karnopp 1977) are considered.

In Chapter 5, some aspects of solving DAE systems numerically by means of the standard multistep Backward Differentiation Formula (BDF) are discussed. Some modelling and simulation software packages can perform symbolic manipulation of the equations of the DAE system *before* it is passed on to a numerical solver. In this context, reduction of the differential index of a DAE system based on Pantelides' algorithm is considered. Moreover, it is demonstrated how tearing of algebraic constraints can be supported by adding controlled sources to a bond graph.

Chapter 6 shows that a bond graph can be considered a *core* model representation from which not only equations for simulation can be derived, but also transfer functions, information about structural properties, e.g., *structural* controllability and *structural* observability as well as equations of the inverse system, parameter sensitivities and forms of state equations for robustness study. Also, bond graphs can support model-based fault detection and isolation.

As is well known, engineering components (e.g., diodes, thyristors, hydraulic check valves, clutches) or physical effects (e.g., stick-slip friction or the stop of a piston at a limiting position) give rise to the modelling abstraction of an instantaneous discontinuous change of state. With regard to the overall system dynamics, this abstraction is appropriate and can help avoid numerical problems due to steep gradients. For these reasons, it is common practice to use this abstraction. Chapter 7 considers several approaches to a combination of this abstraction with bond graph modelling. The latter methodology is based on conservation laws from physics and requires that state transitions take place in a nonzero time period. It appears that so far, no standard bond graph approach to hybrid modelling (time continuous models including the description of discrete events) has formed.

Chapters 8, 9, and 10 on multibody systems, on lumped parameter bond graph model approximations of distributed parameter models, and on open thermodynamic systems show the potential of the interdisciplinary bond graph methodology. *Multibond graphs* as introduced by Breedveld allow for a concise representation of models of multibody systems with bodies assumed to be rigid. In Chapter 8, multibond graph modelling of multibody systems as proposed by Bos is presented. That is, translational motion of bodies assumed to be rigid is referred to an inertial frame, whereas the rotation of a body is described with reference to a body fixed frame. Moreover, bond graph modelling of multibody systems can also be used for the joint coordinate method that is well known in mechanical engineering.

If some bodies of a multibody system are to be considered flexible and consequently are to be represented by a distributed parameter model, then the latter can be approximated by a lumped parameter bond graph model based either on modal analysis (Karnopp and Margolis) or on a finite element approach (Pelegay, Doblare and Buil).

Chapter 10 presents fundamentals of pseudo bond graph as well as true bond graph modelling of open thermodynamic systems. Moreover, some effects in hydraulic systems are addressed that have given rise to an ad hoc representation in some bond graph related research reported in the literature.

Chapter 11 on automated modelling discusses modern modelling languages, the paradigm of object-oriented modelling (OOM), modern software that can be used for systematic development and processing of bond graph models.

As bond graph models have been developed concurrently in various places all over the world, Chapter 11 also addresses the issue of how this engineering knowledge can be shared. Inspired by the success of XML in many computer science related areas, a bond graph markup language, BGML, is proposed that can support the exchange and the reuse of bond graph models.

Regarding software for bond graph modelling, some features of three state-ofthe-art integrated modelling and simulation environments have been considered and illustrated by some screen shots. However, it must be pointed out that the presentation of software is not meant to be product oriented. The aim, rather, is to outline different approaches towards a software support of bond graph modelling.

Finally, in Chapter 12, small elaborated case studies illustrate various applications of the bond graph methodology.

# Contents

1
4
5
8
9
14
17
17
17
18
20
21
23
29
31
35
42
49
56

		2.5.6	Ideal Energy Sources and Sinks	. 59
		2.5.7	Sensors	. 62
	2.6	Pseudo	Bond Graphs	. 63
	2.7	System	atic Construction of Bond Graphs	. 66
		2.7.1	Construction of Bond graphs for Mechanical Subsystems	
			(Translation and Fixed-axis Rotation)	. 66
		2.7.2	Construction of Bond Graphs for Non-mechanical	
			Subsystems	. 70
		2.7.3	Simplification of Some Bond Graph Structures	. 81
	2.8	Some F	Remarks on the Choice of Orientations in Bond Graphs	. 82
	2.9	Conclu	sion	. 85
	Refer	ences		. 87
3	Deriv	ation of	Mathematical Models from Bond Graphs	. 89
-	3.1	On the	Form of a Mathematical Model	89
	3.2	The Co	ncept of Computational Causality	92
		3.2.1	The Notion of Computational Causality	. 92
		3.2.2	Representation of Computational Causalities in Bond	
		- · ·	Graphs	. 93
		3.2.3	Activated Bonds	. 94
		3.2.4	Rules for Causality Assignment at the Ports of Bond	
			Graph Elements	. 96
	3.3	Sequen	tial Assignment of Computational Causalities	. 101
	3.4	On the	Choice of State Variables	. 104
	3.5	System	atic Derivation of Equations from a Bond Graph	. 109
		3.5.1	Procedure for the Manual Derivation of Equations from	
			a Causal Bond Graph	. 112
		3.5.2	Application of the Procedure to Some Examples	. 112
	3.6	Indepen	ndent State Variables	. 119
	3.7	Determ	ination of the Number of Independent State Variables	. 123
	3.8	Conclu	sion	. 126
	Refer	ences		128
4	Caus	al Bond	Graphs and Forms of Mathematical Models	. 129
	4.1	Causal	Paths Between Resistive Ports	. 129
	4.2	Some F	Fundamentals from the Theory of Differential-Algebraic	
		System	IS	. 134
	4.3	Insertin	g Energy Stores into Causal Paths Between Resistive Ports.	. 139
	4.4	Causal	Paths Between Storage Ports of the Same Type	. 142
	4.5	Closed	Causal Paths	. 145
	4.6	Bond C	Graphs with Causal Paths from Different Classes	. 150
	4.7	Causal	Loops of Unity Loop Gain	. 153
	4.8	Algebra	aic Loops due to Internal Modulation	. 158
	4.9	The Me	ethod of Relaxed Causalities	. 162
	4.10	Lagran	ge Causalities	. 166

		4.10.1	Identification of Generalised Coordinates in a Bond Graph	167
		4.10.2	Determination of Generalised Forces from a Bond Graph	167
		4.10.3	Derivation of Lagrange Equations from a Bond Graph	168
	4.11	Conclu	sion	171
	Refe	rences		174
5	Com	puting N	Iathematical Models Derived from Bond Graphs	177
	5.1	Numeri	ical Solution of Differential-Algebraic Systems	177
		5.1.1	The Backward Differentiation Formula	177
		5.1.2	Problems with the Numerical Solution of DAEs by	
			Means of the BDF	179
	5.2	Reduct	ion of the Index of a Differential-Algebraic System	182
	5.3	Reduct	ion of Hamiltonian Equations of Motion with Constraints .	189
	5.4	Tearing	g of Algebraic Constraints	195
		5.4.1	Causal Paths Between Resistive Ports	198
		5.4.2	Causal Conflicts at Junctions	203
		5.4.3	Causal Paths Between Storage Ports of the Same Type	205
		5.4.4	Causal Loops	209
	5.5	The Per	rturbation Index of Differential-Algebraic Equations	215
	5.6	Conclu	sion	217
	Refer	rences		220
6	Anal	ysis of C	ausal Bond Graph Models	223
	6.1	Equation	ons Determining the Steady-state of a Dynamic System	224
	6.2	Transfe	r Functions	227
		6.2.1	Transfer Functions from the State Space Model	228
		6.2.2	Transfer Functions from a Signal Flow Graph	229
		6.2.3	Transfer Functions Directly from a Causal Bond Graph	231
	6.3	Equation	ons of the Inverse System	235
	6.4	Structu	ral Controllability and Observability	240
		6.4.1	Structural Controllability	240
		6.4.2 D	Structural Observability	244
	6.5	Parame	ter Sensitivities	249
		6.5.1	Incremental Models of Linear Bond Graph Elements	250
		6.5.2	Derivation of Parameter Frequency Domain Sensitivities	254
			from an Incremental Bond Graph	254
	6.6	State E	quations for Robustness Study	261
		6.6.1	Incremental Models of Linear Bond Graph Elements	262
		(())	Revisited	262
		6.6.2	Derivation of the Canonical Form of State Equations	265
		(())	The Stee lead Leteneous and Graph	203
		0.0.5	I ne Standard Interconnection Form	269
	67	0.0.4 Diam	Unitime of the Uncertainty Bond Graph Approach	2/1
	0./	Bicausa	al Bond Graphs	270
		0./.1	Bicausai Bond Graphs for Parameter Estimation	279

		6.7.2	Bicausal Bond Graphs for System Inversion	280
		6.7.3	Bicausal Bond Graphs for State Estimation	282
	6.8	Bond (	Graph Model-based Fault Detection and Isolation	282
		6.8.1	Analytical Redundancy Relations	283
		6.8.2	Structural Fault Signature Matrices	286
		6.8.3	Fault Isolation	287
		6.8.4	Residual Sinks in Bond Graph Model-based Fault	
			Detection	288
	6.9	Reduct	tion of Model Complexity	292
		6.9.1	Model Partitioning	292
		6.9.2	Model Reduction	293
		6.9.3	Structural Model Simplification	294
	6.10	Conclu	usion	297
	Refer	ences .		299
7	Mode	els of Va	ariable Structure	305
	7.1	Bond (	Graph Models with Fixed Causalities	306
		7.1.1	Extending Element Characteristics	306
		7.1.2	Switching Between System Modes by means of	
			Modulation	308
		7.1.3	Switched Power Junctions	313
		7.1.4	Switching Off Degrees of Freedom by Sinks of Invariant	
			Causality	317
	7.2	Variab	le Causality Bond Graphs	331
		7.2.1	Ideal Switches as Another Basic Bond Graph Element	331
		7.2.2	Controlled Junctions – Hybrid Bond Graphs	337
	7.3	A Con	nbined Petri Net – Bond Graph Representation	340
	7.4	Conclu	usion	346
	Refer	ences.		349
8	Mult	ibody S	ystems	353
	8.1	Brief S	Survey of Bond Graph Modelling of Multibody Systems	353
	8.2	Multib	oond Graphs	354
		8.2.1	Multibonds and Arrays of Bond Graph Elements	355
		8.2.2	Multiport Energy Storage Elements	356
		8.2.3	Multiport Transformers and Gyrators	358
		8.2.4	Rotation of a Rigid Body in Space Described by a	
			Multiport Gyrator	359
		8.2.5	Multiport Resistors	364
		8.2.6	Splitting a Multibond	365
	8.3	Bond (	Graph Modelling of the 3D Motion of Multibody Systems	366
		8.3.1	Multibond Graph of a Freely Moving Rigid Body	366
		8.3.2	Connecting Instances of the Rigid Body Model	369
		8.3.3	Multibond Graph Model of a Revolute Joint	370
		8.3.4	Multibond Graph Model of a Prismatic Joint	370
			-	

		8.3.5	Multibond Graph of a Three Degrees of Freedom Robot 372
		8.3.6	Causalities in Multibond Graphs 372
	8.4	The Joi	nt Coordinate Method 374
		8.4.1	Formulation of a Reduced Set of Equations of Motion 376
		8.4.2	Reduction of the Equations of Motion: Transformation
			of I Stores in the Bond Graph
		8.4.3	Deriving the Reduced Form of Equations of Motion
			from the Bond Graph
		8.4.4	Application of the Procedure to a Planar Pendulum 380
	8.5	Softwa	re for Modelling and Simulation of Multibody Systems 385
	8.6	Conclu	sion
	Refer	ences	
9	Bond	Graph	Approximation of Distributed Parameter Models 391
	9.1	Approx	imation of a One-dimensional Distributed Parameter
		Model	by an Oscillator Chain 392
	9.2	Brief S	urvey of Bond Graph Approximations of Distributed
		Parame	ter Models
	9.3	Modal	Analysis
		9.3.1	The Bernoulli-Euler Beam
		9.3.2	A Modal Bond Graph Model of the Bernoulli-Euler Beam 397
		9.3.3	State Space Approximation
		9.3.4	Features of the Generic Modal Beam Bond Graph Model . 399
		9.3.5	Further Aspects of the Generic Modal Beam Bond
			Graph Model
		9.3.6	Flexible Mechanical Structures 405
	9.4	Finite <b>F</b>	Element Method
		9.4.1	Classical Finite Element Method Revisited
		9.4.2	Bond Graph Representation of Finite Element Models 412
	9.5	Conclu	sion
	Refer	ences	
10	Bond	Graph	Modelling of Open Thermodynamic Systems
	10.1	Modell	ing Thermodynamic Systems by Pseudo Bond Graphs 426
		10.1.1	Pseudo Bond Graph of a Heated Stirred Tank
		10.1.2	Pseudo Bond Graph of a Variable Pneumatic Control
			Volume
		10.1.3	Pseudo Bond Graph of a Compressible Fluid Flow
			Through an Orifice
		10.1.4	Pseudo Bond Graph of a Pneumatic Bridge Circuit 436
	10.2	True Bo	ond Graph Models of Thermodynamic Systems
		10.2.1	True Bond Graph of a Variable Pneumatic Control Volume 438
		10.2.2	True Bond Graph of a Pneumatic Outlet Orifice
		10.2.3	Further True Bond Graph Approaches to the Modelling
			of Thermodynamic Systems
			· · · · · · · · · · · · · · · · · · ·

		10.2.4	Bond Graph of a Double Acting Hydraulic Cylinder	. 450
		10.2.5	Flow Forces in Hydraulic Spool Valves	. 456
	10.3	Conclu	sion	. 464
	Refer	ences		. 467
11	Auto	motod M	[ada]];na	160
11	Auto	Continu	loue function Languages	. 409
	11.1	Object	Oriented Modelling Languages	. 470
	11.2	11 2 1	Connection of Submodels Assording to the Dhysical	. 470
		11.2.1	Structure of the System	470
		1122	Algebraic Loops	480
		11.2.2	Algebraic Dependencies Between State Variables	480
	113	Bond G	angebraic Dependencies Detween State Variables	482
	11.5	Describ	and Grants in SIDOPS	484
	11.4	Describ	aing Bond Graphs in Modelica	490
	11.5	11 5 1	Bond Graph Power Ports and Their Interconnection	490
		11.5.1	Basic Bond Graph Elements	495
		11.5.3	Computational Causality	. 498
		11.5.4	Hierarchical Bond Graphs	. 500
	11.6	Softwar	re for Bond Graph Modelling	. 505
		11.6.1	ENPORT <sup>TM</sup>	506
		11.6.2	TUTSIM <sup>TM</sup>	508
		11.6.3	Bond Graph Preprocessors	511
		11.6.4	Bond Graph Toolboxes	514
		11.6.5	Integrated Modelling and Simulation Environments	. 515
		11.6.6	Transformation Between Different Model Description	
			Forms	. 530
	11.7	Exchan	ge and Reuse of Bond Graph Models	. 535
		11.7.1	Useful XML Features for the Description and Processing	
			of Bond Graph Models	. 536
		11.7.2	Information that an Exchange Format for Bond Graphs	
			Should Capture	. 537
		11.7.3	A Schema for an XML Based Description of Combined	
			Bond Graph and Block Diagram Models	. 538
		11.7.4	Pseudo Bond Graphs in BGML	. 546
		11.7.5	Controlled Junctions in BGML	. 546
		11.7.6	Supporting the Exchange and Reuse of Submodels	. 547
		11.7.7	Iransforming the BGML Description of a Bond Graph	<b>E 40</b>
		11 7 0	Model into a Target Language	. 549
	11.0	11./.8	AIVIL Based Bond Graph Component Model Libraries	
	11.8 D.C.	Conclu	sion	. 554
	Kefer	ences		

12	Appli	cations	
	12.1	Inverted Pendulum	561
	12.2	Shunt Motor	566
	12.3	A Machine with an Unbalanced Rotor	570
	12.4	An Electronic Balance with Displacement Compensation	575
	12.5	A Piezoelectric Seismometer	
	12.6	Engagement of a Clutch	
	12.7	Dry Friction in a Suspension Strut of a Car	593
	12.8	A Buck Converter	601
	12.9	A Two Degrees of Freedom Rotary Joint Manipulator	610
	12.10	Fluid Level Control in a Three Tank System	
	12.11	Fault Detection in a Hydraulic Two Tank System	
	12.12	Heated Stirred Tank	629
	12.13	A Counterflow Heat Exchanger	
	12.14	Conclusion	
	Refere	ences	639
13	Overa	Ill Conclusion and Outlook	643
	Refere	ences	648
Glo	ossary		651
Ind	ex		657

# Abbreviations

Three dimensional
Analytical Redundancy Relation
Backward Differentiation Formula
Block Diagram
Bond Graph
Bond Graph Markup Language
Constitutive Relation
Continuous System Simulation Language
Differential-Algebraic Equation
Document Type Definition
Degree Of Freedom
Eulerian Junction Structure
Fault Detection and Isolation
Finite Element Method
Fast Fourier Transform
Finite State Machine
Graphical User Interface
Hybrid Bond Graph
Input-Output
Initial Value Problem
Linear Fractional Transformation form
Local Truncation Error
Linear Time Invariant system
Multibody System
Multiple Input Multiple Output system
Nondeterministic Polynomial
Ordinary Differential Equation
Object-Oriented Modelling
Object-Oriented Programming
Partial Differential Equation
Proportional, Integral and Derivative

PRV	Pressure Relief Valve
SCAP	Sequential Causality Assignment Procedure
XML	eXtensible Markup Language

## Chapter 1 Introduction

#### 1.1 A Historical Survey of Bond Graph Modelling

When designing a new dynamic system or analysing an existing one, it is common for designers and engineers to use graphical representations of their models in order to communicate with others, to exchange ideas, to express modelling assumptions and to exchange their models for reuse. This is not surprising, as graphical representations are far more suited to human perception than oral or textual ones. For instance, a schematic of a closed loop controlled system is clearly more easily understood than a verbal description given over the telephone. In engineering disciplines, e.g., network representations, block diagrams, or linear graphs have a long tradition. In addition, (domain specific) iconic diagrams have become popular. If graphical representations adhere to formal rules, not only do they avoid misunderstandings between human beings, but they also allow for an automatic transformation into an executable program by means of appropriate software programs.

Among several graphical representation means used in different application areas, bond graphs are a description formalism best suited for modelling physical processes and multidisciplinary dynamic engineering systems including effects or components from different energy domains, viz., the mechanical, the electrical, the thermal, and the hydraulic domain. Many technical systems, often termed *mechatronic* systems, integrate components from different disciplines and exploit interacting effects, e.g., sensors and electronically controlled actuators.

Bond graphs, to be formally introduced in the next chapter, were devised by Professor Henry Paynter<sup>1</sup> at the Massachusetts Institute of Technology (MIT) in 1959. His former Ph.D. students, Professor D. Karnopp and Professor D. Margolis (University of California at Davis), and Professor R. Rosenberg (Michigan State University, East Lansing, Michigan), elaborated the concept into a methodology for physical systems modelling that nowadays is used in academia and industry by many people around the world. In retrospect, Paynter wrote in 1992 ([35] p. 13):

<sup>1 1923-2002</sup> 

So it was that on April 24, 1959, as the writer was about to give a seminar lecture at Case Institute (now Case-Western) on "Interconnected Engineering Systems", he awakened earlier that morning with the 0,1-junctions somehow finally planted in his head! Thus on that date the  $BG^2$  system was complete and constituted a formal discipline.

The first published books of these pioneers include Paynter's historic lecture notes titled *Analysis and Design of Engineering Systems*, dating back to the year 1961 [34], the book titled *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics* by Karnopp and Rosenberg [22] published in 1968, as well as the first edition of the textbook *System Dynamics – A Unified Approach* [23]. This book has become a widely recognised standard. A second edition was published in 1990 and a third edition in 2000, both co-authored by D. Margolis. In 2006, the three authors published an even more mature fourth edition that reflects their experience over decades in teaching courses at universities and in industry. This textbook now is titled *System Dynamics - Modeling and Simulation of Mechatronic Systems* [25].

Early promoters of the new modelling technique were Professor J. Thoma (Professor Emeritus at the University of Waterloo, Ontario, Canada), Professor J. J. van Dixhoorn (University of Twente, Enschede, Netherlands who passed away in 2001), P. Dransfield, Professor at Monash University, Melbourne, Australia and S. Scavarda, Professor at Institut National des Sciences Appliquées de Lyon (INSA), France who passed away in 2008. These gentlemen significantly contributed to the promotion and dissemination of the bond graph modelling technique in Europe, Japan, India and China. Van Dixhoorn founded a Technical Committee on Bond Graph Modelling (TC 16) as part of the International Association for Mathematics and Computers in Simulation (IMACS) that was chaired by J. Thoma for many years.

Right from its beginning, the bond graph methodology was supported by the famous ENPORT<sup>TM</sup> simulation program developed by R. Rosenberg. Its probably best known version has been ENPORT-4<sup>TM</sup>. At that time, bond graphs were entered in alpha-numerical form in a so-called line code. Entries in that line code separated by commas denoted the type of a bond graph vertex followed by the numbers of the edges attached to that node. In his Ph.D. thesis [20], Professor J. Granda (California State University at Sacramento) developed a bond graph preprocessor that transformed the line code into equations for input into widely used simulation programs such as ACSL<sup>TM3</sup>. In a further obvious step, he replaced the line code entry by a graphical editor.

Nowadays, several facts demonstrate the worldwide acceptance and the success of bond graph modelling methodology. During the first years of bond graph methodology development, almost everyone concerned with the new technique knew almost all publications and moreover, knew many members of the small but worldwide community of bond graph modellers personally. Today, the number of bond

<sup>&</sup>lt;sup>2</sup> BG means Bond Graph

<sup>&</sup>lt;sup>3</sup> ACSL, acsIX, and PowerBlock are registered trademarks of The AEgis Technologies Group, Inc., 631 Discovery Drive, Huntsville, AL 35806, USA, http://www.acsIx.com

graph related publications has grown tremendously so that a comprehensive survey is almost impossible. In 1977, V. Gebben [19] published the first *Bond Graph Bibliography* with the aim of recording the enormous increase in the number of bond graph related publications. Updates followed in 1985 [6] and 1991 [8], both published in the renowned *Journal of the Franklin Institute*. Around 1996, Professor F. Cellier, now with the Swiss Federal Institute of Technology, Switzerland, took on the tremendous burden of compiling references to bond graph related publications in a Bond Graph Compendium available on the World Wide Web [12]. Even this comprehensive compendium needs another update. For instance, in the year 2000 alone, several textbooks on bond graph modelling were published [3, 13, 24, 30, 41]. Some more recently published books on bond graph modelling are [25, 31, 37, 43].

In the past decades, bond graph researchers contributed many special sessions on bond graph modelling to international conferences. The author organised such special sessions as part of the ESM 1993, Lyon, France, of the 1994 and the 2003 Mathmod conference in Vienna, Austria, as part of the ESS 1997 in Passau, Germany and contributed to bond graph sessions organised by other members of the community, e.g., to CESA 1996 in Lille, France, and the CIFA 2008 in Bucarest, Romania. As general chairman, the author of this book organised the 2006 European Conference on Modelling and Simulation (ECMS 2006) held near Bonn, Germany [5]. This conference featured a well received track with three sessions devoted to bond graph modelling. Professor Cellier delivered a keynote speech by addressing his current research activities in bond graph modelling. During the last decade, many more conferences with papers, sessions, or even tracks addressing bond graph modelling took place. Space in this introduction does not allow for all of them to be listed.

Besides publications in international conferences, bond graph researchers have contributed to special journal issues, e.g., the 1999 special issue of *Simulation Practice and Theory* edited by J. U. Thoma and H. J. Halin [42], the 2002 special issue of the *Proceedings of the Institute of Mechanical Engineers* edited by P. Gawthrop and S. Scavarda [17], the 2006 special issue of the journal *Mathematical & Computer Modelling of Dynamical Systems* edited by I. Troch, W. Borutzky and P. Gawthrop [44], or the 2009 special issue of the journal *Simulation Modelling Practice and Theory* edited by the author [4]. The latter special issue also includes an introduction of bond graph modelling by the author of this book. In 2007, Gawthrop and Bevan published a tutorial introduction into bond graph modelling for control engineers in the IEEE Control Systems Magazine [16].

Last but not least, the biannual *International Conference on Bond Graph Modelling* (ICBGM) as part of the Western Multiconference (WMC) of the Society for Modelling and Simulation International (SCS) has to be mentioned.

In addition to research, bond graph based physical systems modelling has been considered more and more to be a important fundamental topic in engineering education and has become a regular part in the syllabi of many engineering programmes. In addition to the 2000 textbooks already mentioned above, other textbooks on bond graph modelling have been published in different languages during the past decades, e.g. [2, 7, 14, 18, 23, 32, 40, 46]. This list, which is not meant to

be comprehensive, shows that there has been a growing awareness and acceptance of the bond graph modelling technique worldwide during the past decades. In his contribution to the March, 1995 special issue on simulation in engineering of the journal SIMULATION, Professor F. Cellier points out why bond graphs are "the right choice for educating students in modelling continuous-time physical systems" [11].

Not only in research and education, but also in industry, bond graph modelling has become a useful approach in the everyday business of many engineers in small consulting firms as well as in big companies, especially the automobile industry, in aerospace and in automation. In the context of mechatronic system design, the appropriateness of a bond graph approach is particularly evident. On the other hand, it must be mentioned that bond graph modelling is one of a number of modelling methodologies that are equally well suited for a given design task. People in academia and industry do have their preferences. Consequently, bond graph modelling is not appreciated to the same extent in different places. Although bond graph modelling has spread from MIT to many places all over the world, there are engineering departments even in the USA where this technique is still not used.

In some countries, bond graph modellers founded national associations, e.g., the *Bondgraafclub* in the Netherlands, or the *Club de Bondgraphistes* with members in France and in Belgium.

In 2000, the first edition of this monograph was published as a first comprehensive presentation of bond graph methodology in the German language. The author of this monograph became aware of bond graphs in 1979 by a survey article by A. Schöne [38]. It was the starting point of an ever-continuing enthusiasm for this methodology that I share with friends in many places around the world.

#### 1.2 Some General Aspects of Modelling Dynamic Systems

Building a model is an iterative procedure. It starts with the identification of essential features and of inherent mechanisms of a dynamic system to be designed. In a step by step refinement of the understanding of a dynamic physical system, different forms of representation are used. They are of graphical nature, especially during the conceptual phase as our eyes can easily perceive different information in parallel. Graphical representations are not only easy to grasp, they are also best suited for communication between individuals. This becomes evident if one considers the difficulty of transmitting all the information contained in a schematic of an electronic or hydraulic circuit by telephone, or if one receives the description of a dynamic system in a simulation language on many pages without any comments. The importance of graphical representations is not only essential in the modelling phase, but also for visualisation of simulation results. With the ever increasing computing power, not only is the graphical representation of numerical results as curves required, there is also a demand for 3D animation of system motion in a realistic fashion. With the development of languages such as Virtual Reality Modelling Language (VRML) and appropriate freely available visualisation software tools, 3D presentations may be exchanged via the internet. Thus, graphical representations and visualisation play a vital role during the design of a dynamic system and the design of its control [36].

Graphical and textual model representations of dynamic systems are always associated with a certain view of a system, its properties and its inherent mechanisms. They reflect abstractions and modelling assumptions. The starting point of the modelling process always involve certain questions. Thus, features and system properties are assessed and are either taken into account or deliberately neglected. These considerations and decisions as well as the designer's experience have an impact on the choice of the graphical description. Answers to the question as to what purpose a model of a system to be designed shall serve, may give an indication to an appropriate choice of a description formalism from a repertoire of possible means. Often, the answer to the question concerning the most suitable form of representation is not unique. Several description formalisms may equally serve the requirements, or depending on the design task, a combination of some of them may be appropriate, e.g., a bond graph representation of a system and a block diagram of its control. Moreover, a graphical representation may be transformed into another one provided both are equivalent. For instance, a network could be transformed into a bond graph which in turn could be transformed into a block diagram. Consequently, a program for control systems could be used to simulate a system described as a network. There is a similar situation in computer science. For the software implementation of a solution to a given problem, several programming languages may be appropriate, whereby each one has its own flavour. On the other hand, software programs for automatic translation from one programming language to another are available.

#### 1.3 Object-Oriented Physical Systems Modelling

The ever increasing performance of computers and simulation software has enabled one to model and to simulate problems of more and more complexity. While the simulation of a problem required hours of computation time in former times on computers called minicomputers, the same problem can be solved within minutes or less on a personal computer. The ever increasing complexity of problems to be analysed has had an essential impact on modelling methodologies and the software tools supporting them.

In order to cope with the increasing complexity of the system to be analysed, it is obviously necessary to pursue a hierarchical approach and to build libraries with reusable submodels as is known, for example, from the design of large integrated circuits. Moreover, for the development of large models, the need for a graphical representation becomes apparent. It is true that continuous system simulation languages (CSSLs), e.g., ACSL<sup>®4</sup> (*Advanced Continuous System Simulation Language*), offer macro features to build modularly structured models. However, instead of

<sup>&</sup>lt;sup>4</sup> ACSL, acsIX, and PowerBlock are registered trademarks of The AEgis Technologies Group, Inc., 631 Discovery Drive, Huntsville, AL 35806, USA, http://www.acsIx.com

writing many thousands of lines of code in a more or less well structured model description language, it is more reliable to hierarchically develop a graphical model using model libraries and to have the result transformed automatically into simulation code. To that end, an easy to use graphical user interface (GUI) alone is not sufficient. What is needed is an appropriate model description language underlying the graphical model representation and its automatic transformation into a modelling language. However, as languages based on the CSSL standard established in 1967 [39] are more simulation languages than modelling languages, they show shortcomings in supporting the development of hierarchical modular structured models. Such a modelling language was described as early as 1979 in Elmqvist's dissertation [15].

On the other hand, in computer science, software projects of ever increasing size have led to the paradigm of Object-Oriented Programming (OOP). In their widely recognised 1991 book, Object-Oriented Modelling and Design, J. Rumbaugh and his co-authors present object-oriented modelling as a methodology for the design of large complex software systems that include the analysis of the problem, the design and the software implementation. They use the term Object Modelling Technique (OMT) and point out in the preface of their book that object orientation means more than merely a kind of programming. The attractiveness and the success of this fundamental concept has also had an impact on the terminology in physical systems modelling and the manner models of large systems are developed and described. Since about 1990, with the event of modelling languages such as Omola [1], the term object-oriented physical systems modelling has come into use. In contrast to classical control, describing a system's behaviour by functional input-output blocks and consequently focusing on the computational structure, object-oriented modelling emphasises the view of a model composed of non-causal submodels connected accordingly to the *physical structure* of the system. The new approach at that time was promoted by Anderson and Mattson [28] and F. Cellier [10]. In his dissertation [33], Otter used the modelling language Dymola<sup>®5</sup> [15], developed by Emqvist in as early as 1978, for an object-oriented approach to mechatronic systems modelling. Without discussing details of the language Dymola or its successor Modelica<sup>®6</sup> [29], the characteristics of an object-oriented physical systems modelling approach are presented in the following.

#### Characteristics of Object-Oriented Physical Systems Modelling

According to principles of object-oriented programming in software engineering, object-oriented physical systems modelling may be characterised by the following features.

• Objects

In object-oriented physical systems modelling, models of components of engineering systems as well as models of physical processes are considered to be

<sup>&</sup>lt;sup>5</sup> Dymola<sup>®</sup> is a registered trademark of Dynasim AB, Ideon Science Park, SE-223 70 Lund, Sweden, http://www.dymola.com

<sup>&</sup>lt;sup>6</sup> Modelica<sup>®</sup> is a registered trademark of the Modelica Association, http://www.modelica.org

#### 1.3 Object-Oriented Physical Systems Modelling

*objects*. Since basic models are comprised of inherent parameters, e.g., length, mass, moment of inertia, resistance, or data provided from the outside world through so-called *interfaces*, and constitutive equations, there is a correspondence to objects in the sense of object-oriented programming (OOP). Physical system models may be viewed as an aggregation of data and *methods* operating on them. In OOP, the term *method* means a function or a procedure that can process data of a defined type. For instance, the coordinates of a point, a length, and a function that returns the area of a circle around that point may be aggregated into a *class* called *circle*. A circle around a given point of given radius may be named c1. Then, c1 is a particular object of the class *circle*. Likewise, the voltage across and the current through a resistor, parameters, and the constitutive (non-linear) relation between voltage and current may be considered a class *resistor* with given values for the parameters in the constitutive relation called, for instance, R5, corresponds to a particular resistor in a circuit.

• Model Hierarchy

As it is well known, a model of a system may be composed of lower level submodels which in turn may contain submodels as well. That is, physical-system models are *hierarchical* in nature (in the sense of a membership relation).

Model Classes and Instantiation

As explained above, physical system models and submodels of components can be viewed as particular objects of a certain class. In object-oriented modelling, the term *instantiation* is adopted from OOP. A model or submodel is called an *instance* of a model class. Models or submodels are *instantiated* from generic models or *model classes* in which more general properties common to the members of a model class are described. The members of a model class have got the same structure and exhibit the same general dynamic behaviour. For example, an instance of the model class *diode* is obtained by giving particular values to its parameters. The resulting instance corresponds to a particular diode in a circuit.

Inheritance

If a submodel class is instantiated into a particular submodel, its properties are inherited by the submodel. That is, the particular submodel declaration is a special version of the more general submodel declaration. In Section 11.5.2, an incomplete model class is introduced that only captures the energetic property of a 1-port energy storage element in the sense that this type of element stores a physical quantity such as electrical charge. In that class, a relation between the state and the rate variable is not given. By adding this information, a particular class can be derived. The more special model class inherits all properties of the superclass from which it is derived. Consequently, it describes a particular type of an energy store. Again, by specifying parameters, a particular object or instance of this class is obtained associated with a particular energy store in the system. Another example is an incomplete model class *passiveOnePort* that accounts for the passivity of a 1-port element. A special class *diode* may be obtained from this incomplete superclass inherits the passivity property of a 1-port from the su-

perclass. Using inheritance in the declaration of library model classes has the advantage that the potential of errors in the code is reduced. All features of the superclass are copied. Only features that characterise the more special class need to be added.

• Encapsulation

Knowledge contained in a model is encapsulated. Only a well defined part of it can be accessed in a well defined manner via interfaces to the outside world of an object. This means that the internal definition of a submodel is not affected by the connection of the submodel to other submodels. The part of a submodel describing its interfaces is separated from the part in which the behaviour is described by means of *non-causal* mathematical equations. The latter do not need to be known when submodels are connected in order to build a hierarchical model. In generalised networks for instance, an interface of a submodel is a pin. It is described by two power variables called *across* and *through* variables.

• Polymorphism

In a submodel definition, the description of its interfaces to the outside world is separated from the internal description of its dynamic behaviour. In this internal description, multiple possible cases may be taken into account. Consequently, depending on current conditions, the same submodel may exhibit different behaviour.

Connection of Submodels

Submodel interfaces are connected accordingly to component interconnections in the real physical system, also called the *physical structure* of the system. In contrary to block diagrams, this means that equations of a submodel must be *non-causal*. The interconnection of submodels may require solving them for certain variables. In block diagrams, it is fixed a priori whether an interface variable of a block is an input or an output variable. This must be taken into account when connecting blocks.

Of course, it has been a tradition in physical systems modelling to build hierarchical models and to connect submodels according to the physical structure long before object-oriented modelling was introduced. In Chapter 11, aspects of object-oriented physical systems modelling will be picked up again when looking at bond graph modelling from an object-oriented modelling point of view.

#### 1.4 Traditional Graphical Model Representations

There are many forms of graphical model representation in use in different engineering disciplines, e.g., free body schematics in mechanical engineering, circuit representations in electrical engineering and in hydraulics, linear graphs as well as block diagrams and signal flow diagrams in control theory. All of these forms are well known and have a long tradition. Therefore, only some aspects of block diagrams and signal flow graphs on the one hand, and network representations on the other hand, will be outlined. The aim is to show the context in which bond graph methodology is embedded. In the following, aspects are discussed from a methodology point of view. In Chapter 11, links to corresponding software tools are considered.

#### 1.4.1 Block Diagrams

Block diagrams have the following characteristics:

- They support the abstraction of unilateral signal flow through a system. Information flow is considered not to be bound to the transfer of energy or the transport of mass. A block in an oriented chain of blocks does not have a direct impact on its predecessor. Feedback is taken into account separately by signal feedback loops.
- The interface variables of a signal block are a priori discriminated into input and output signals, independent of the actual use of a signal block in a block diagram model. This must be taken into account when signal blocks are connected.
- Signal blocks represent functional relations between input and output signals. It is neither required nor ensured that relations comply with first principles of physics. Blocks may represent any linear, or nonlinear algebraic, or time dependent relation.
- Block diagrams display which variables must be known in order to compute oth-٠ ers. They represent the structure of the mathematical model, or as van Dixhoorn [45] has termed it, the *computational structure*. They do not reflect the *physi*cal structure of a system. The reason is that feedback is represented by separate feedback loops. Signal blocks cannot be connected like corresponding system components. For instance, if two electrical devices are connected, then the voltages are set to be equal and at the same time, currents are added to zero. In block diagrams, however, a connection between two blocks represents only one signal. As will be pointed out later, in bond graphs, each edge is associated with two conjugate power variables. Consequently, connected bond elements always have a feedback to each other. If submodels in a block diagram are modified by neglecting effects or by taking into account additional ones, then small changes may have a considerable impact on the computational structure and thus on the structure of the block diagram. This disadvantage does not appear in networks or in bond graph graphs.
- As block diagrams represent signal flows and functional relations independent
  of the physical meaning of variables, they can be used in different engineering
  disciplines. They are used particularly in control since control systems are often
  designed in such a manner that there is no feedback between components. The
  computational structure then corresponds to the physical structure.
- Finally, block diagrams support a hierarchical decomposition into functional blocks.



Fig. 1.1 Causally completed bond graph model of a second order system



Fig. 1.2 Block diagram corresponding to the bond graph in Figure 1.1

As will become clarified, bond graphs reflect the physical structure of a system, as do networks. On the other hand, the computational structure may be superimposed on a bond graph by adding a perpendicular stroke to each edge, turning the initially non-causal model into a causal one. Moreover, such a causally completed bond graph can be systematically transformed into a block diagram if needed. During this transformation, information about the physical structure gets lost, as can be seen from Figures 1.1 and 1.2. As signal blocks in block diagrams can represent any functional relation, the converse does not hold. Not every block diagram can be transformed into a bond graph. Equations represented by bond graphs should comply with the first principles of physics.

#### 1.4.2 Signal Flow Graphs

In signal flow graphs, the role of edges and vertices is essentially interchanged in comparison to block diagrams. Oriented edges represent functional relations between variables, while nodes are used to represent variables and the summation of variables. In that respect, they may be considered the dual of block diagrams. However, signal flow graphs are less general than block diagrams because besides the summation of variables, no functions with more than one input variable can be represented. Like block diagrams, signal flow graphs represent the computational, not the physical structure of a system. In the case of linear models, they can be used to derive the transfer function between two variables by applying Mason's loop rule [27]. Bond graphs can also be transformed into signal flow graphs as shown in Fig-

$$C: C \xrightarrow{1} 1 \xrightarrow{3} TF \xrightarrow{1/n} 4 0 \xrightarrow{6} R: R$$

. .

#### Fig. 1.3 Causally completed bond graph fragment



Fig. 1.4 Signal flow graph corresponding to the bond graph fragment in Figure 1.3

ures 1.3 and 1.4. However, as Brown has shown, Mason's loop rule can be applied directly to a bond graph [9].

#### 1.4.3 Networks

In electrical engineering, it is common to represent models as networks. However, this representation is not restricted to electrical systems. By relating the electrical power variables voltage and current to non-electrical quantities appropriately, networks may be used to represent models of systems in other energy domains. A unified approach to the modelling of engineering systems based on so-called *generalised networks* has been introduced by MacFarlane [26]. Networks have the following essential features.

 Contrary to block diagrams, networks represent the physical structure, but not the computational structure of an engineering system. Submodels are connected like corresponding components or devices in the real system. There is no need to decide whether an interface variable is an input or an output variable. The graphical representation of submodels is not uniform across energy domains as in bond graphs, but depends on the engineering discipline. For instance, hydraulic circuits use different icons for submodels than electrical circuits. In any case,



Fig. 1.5 Circuit with two energy stores

Kirchhoff's laws hold for electrical power variables as well as for corresponding variables in other energy domains.

- Networks are hierarchical in nature. Submodels can have a network structure.
- Networks account for energy flows in a system. Circuit nodes connecting pins of submodels comply with power conservation. Moreover, physical quantities, e.g., charge, are conserved if properly taken into account in the development of submodels. Meyer's NMOS transistor model, for instance, does not ensure conservation of charge ([21], Section 3.4.4)

Bond graphs, which will be introduced formally in the next chapter, on the one hand, reflect the physical system structure like networks. On the other hand, a computational structure can be superimposed so that the causally completed bond graph can be considered a concise representation of a block diagram. As bond graphs have features in common with block diagrams and with networks, both representations have been briefly discussed in this introduction. Moreover, bond graphs can be considered a core model representation for the following reasons.

- Generalised networks can be systematically converted into bond graphs (cf. Section 2.7). If orientations of edges in a bond graph indicating the reference direction of the energy flow across a bond are chosen with care, then the directed bond graph is equivalent to the network from which it has been constructed (Section 2.8). For instance, if the circuit in Figure 1.5 is converted into a bond graph, then the result is the bond graph in Figure 1.3.
- From a causally completed bond graph, a block diagram as well as a signal flow graph can be derived.
- Finally, domain-specific iconic diagrams can be systematically converted into a bond graph if there are bond graph equivalents of basic icons.

#### 1.5 Conclusion

Dynamic system modelling has a long tradition. On the other hand, the views of model developers, methods and corresponding software tools change, and new is-