



FACULTY OF ECONOMICS  
AND BUSINESS ADMINISTRATION

Ghent University  
Faculty of Economics and Business Administration  
Department of Management Information and Operations  
Management

# **Goal-Oriented Requirements Engineering for Business Processes**

---

Ken Decreus

Advisor: Prof. dr. Geert Poels

Submitted to the Faculty of Economics and Business Administration  
of Ghent University in fulfillment of the requirements for the degree  
of Doctor in Applied Economics  
November 2010





FACULTY OF ECONOMICS  
AND BUSINESS ADMINISTRATION

Ghent University  
Faculty of Economics and Business Administration  
Department of Management Information and Operations  
Management

Advisor: Prof. dr. Geert Poels

Ghent University  
Faculty of Economics and Business Administration  
Department of Management Information and Operations Management  
Tweekerken 2, B-9000 Gent, België  
Tel.: +32-9-264.35.17  
Fax.: +32-9-264.42.86

Submitted to the Faculty of Economics and Business Administration  
of Ghent University in fulfillment of the requirements for the degree  
of Doctor in Applied Economics  
November 2010



# Doctoral Jury

- Prof. dr. Marc De Clercq (Dean Faculty of Economics and Business Administration)
- Prof. dr. Patrick Van Kenhove (Academic Secretary Faculty of Economics and Business Administration)
- Prof. dr. Geert Poels (advisor)
- Prof. dr. Monique Snoeck (Katholieke Universiteit Leuven)
- Prof. dr. Roland Paemeleire (Ghent University)
- Prof. dr. Patrick Heymans (Université de Namur)
- Prof. dr. Peter Rittgen (University College of Borås Sweden, Vlerick Leuven Gent Management School Belgium)
- Prof. dr. Dirk Deschoolmeester (Ghent University, Vlerick Leuven Gent Management School Belgium)



I fully dedicate this PhD to my father Freddy and my mother Sonia, as this PhD never would have been realized without them. For twenty seven years, they supported me in everything I did in life and arranged my world in such a way that I could always chase my dreams. As I grow older, I wonder how many of the seven billion people on this planet have had that privilege.





# Acknowledgments

On 20 May 2010, I received an e-mail from my father, titled „Something worthwhile to look at”. The e-mail contained a recorded interview of Jeremy Rifkin, the chief architect of EU’s sustainability plans. Looking at this interview gave me a feeling of discomfort, unbelief and doubt. Mr. Rifkin explains that we passed the stage of merely destroying planet earth, and that mankind is heading for extinction within the coming decades. Intrigued by this statement, I ordered Mr. Rifkin’s book (*The Empathic Civilization*), and started to investigate all six hundred pages. Unfortunately, I could not find any loophole in his reasoning, and began to understand why Merkel, Sarkozy, Zapatero and the European Commission appointed Mr. Rifkin as their direct advisor. In essence, Mr. Rifkin states that the empathy in our society has never been higher than today, and this global empathy requires a massive amount of energy. In this perspective, the energy usage is not the problem, but rather the high amount of energy losses of our current technologies, based on oil, gas and uranium. These energy losses are captured by the atmosphere of our planet, causing an unstoppable trend in global warming which threatens every life form on a very short term (i.e., our current generation). In order to tackle this problem, Mr. Rifkin explains that between 2010 and 2035 the third industrial revolution should take place, based on renewable energy, considering houses as energy plants, commercializing hydrogen power cells and employing smart electrical grids. He makes clear that it will be difficult to avoid human collapse, but he still believes that there is a small chance of success.

Five days later, on 25 May 2010, I received the monthly BPTrends newsletter from Paul Harmon, one of the thought leaders in Business Process Management, titled „Preparing for change”. The purpose of Paul Harmon was not to warn of global disaster, but to be aware that there are going to be major changes in the years ahead, that many will probably arise very quickly, and that our organizations will need to respond just as quickly if we are to continue to survive and prosper. Harmon recommends that every company should have an environmental team and that team ought to be working very closely with the organizations business process management team.

Twenty one days later, on 10 June 2010, I was present at the CAiSE 2010 conference at Hammamet, Tunisia, and I listened carefully to a speech of Mr. Paraszczak, top manager at IBM’s Watson Research Centre, titled „The influence of IT systems on the use of the earth”. Mr. Paraszczak explained that IBM organised a global workshop with all its main customers worldwide, and asked them the question how IBM can improve its current business. The number one concern was global disaster and the need for sustainability, after which IBM physically restructured their organisational model to offer worldwide support for sustainability. IBM is getting ready for change, in the very near future.

In the middle of August, I was watching TV and noticed a new offering of Renault-Nissan. During a one-minute TV commercial, they

showed the upcoming human collapse in Hollywood style, and state that Renault-Nissan will be the first car manufacturer to introduce the electrical car. As electrical cars need electrical batteries, the dominance of the petrol stations will be challenged by new initiatives. Renault-Nissan will cooperate with Better Place, a company founded by Shai Agassi, the former president of SAP's Product and Technology Group. For those who do not know Shai Agassi, he was the architect of the largest strategic reform of the SAP product line (i.e., SAP Netweaver), impacting billions of users worldwide. A mastermind such as Shai Agassi understood the urgency of sustainability, and took a radical career change to start Better Place.

In order to understand the sustainability issue, a team of MIT researchers wrote a book in 1972, called *Limits to Growth*. They applied system theory to the human socio-economic system, which states that overshoot of the human socio-economic system happens when three conditions are met. Firstly, there should be exponential growth, which is the case for all system variables, being population, economy, food production, energy usage and pollution. Secondly, there need to be barriers to this growth, which is the case for our planet, as we consume resources in a non-sustainable rate, and we emit greenhouse gases in a non-sustainable rate. Wackernagel [Wac02] demonstrated in 1999 that we overshoot the carrying capacity of the earth by 20%. Thirdly, there should be a (fatal) delay in response of the exponential growth and the barriers of the earth, which is the case in our society for one or two generations („Did you already see the North Pole melting? Not me!"). To illustrate these three conditions for overshoot, we take the example of a car that approaches a barrier, being traffic lights. When the car driver accelerates from first to fifth gear, speed and distance grow exponentially. In foggy conditions, the driver might not see a red light, causing a delay between realizing that the traffic light is red, and the moment he hits the breaks. We all understand that it's no use to hit the breaks after passing the red light, because the car accident is already occurring. In terms of this metaphor, our planet passed the red lights in the 1980s, and we are about to understand the injuries of the upcoming accident.

So how sure are we that the 1972 simulations of *Limits to Growth* are valid? Validity of simulations is defined by means of two aspects, i.e. the validity of the underlying simulation model, and the validity of the assumptions used to run the simulation model. Firstly, the MIT researchers used mathematical proof to guarantee the validity of the underlying simulation model. Secondly, simulations can prove any statement based on any assumption, so the „garbage in, garbage out" rule is applicable. In the updated version of the book (written in 2004, [MRM04]), the authors provided a first set of historical evidence (ranging from 1972 to 1990) to prove their statements. In 2008, independent scientific research [Tur08] confirmed the trend towards human collapse based on data from 1970 to 2000. From 2000 onwards, we all understand that it is not a difficult exercise to find data that show continued exponential growth in population, economy, food production, energy usage and pollution. When we extrapolate Wackernagel's analysis [Wac02], we currently (in 2010) have a 40% overshoot of the carrying capacity of our planet. The tricky part of our non-sustainable behaviour is that we cannot rewind the damage done, so by the time that we start to feel the human implosion (somewhere between 2020 and 2030), it is a one-way down, and it cannot be stopped any longer.

As you start to understand, the world we are currently living in, is

about to change severely. So how does this relate to my PhD work? To start with, my PhD topic deals with methods to transform strategic requirements into operational business processes. As the first signs of the third industrial revolution are already becoming visible, a lot of changes will occur at the strategic level of businesses, which will impact the operational business processes. Hence, business managers will need -more than ever- support for designing business processes based on these strategic requirements, to which my PhD aims to contribute. Next, in every facet of my work, I found evidence that Mr. Rifkin's empathic civilization is being shaped. The last couple of years, I have been travelling around the globe, and visited places such as Hawaii, South Africa, New York, Peru, Atlanta, Tenerife, Costa Rica, Panama, Gabon and Tunisia. In all these places, the same scenario occurred: I jogged along the coast line while listening to U2's Achtung Baby on my iPod, I paid with my VISA card, I checked my e-mail and Facebook account and used the same cell phone. But more important, on all these places, I was welcomed by friends of mine, or made new friends after a day or two. It was this global empathy, ranging from Belgium to Hawaii, which provided me sufficient energy to focus on my PhD, and to do my best to deliver work of added value to the community. Now, I want to take this opportunity to thank those people who supported me and believed in my work.

In the first place, I want to thank my supervisor Geert for offering me the opportunity to do my PhD. From the start till the end, he allowed me to grow as an academic, and he supported me both financially as well as personally. He was always available to review my writings, and his academic expertise lifted this PhD to a quality level that I once believed was impossible. Secondly, I would like to thank Frederik, as he looked over my shoulder on a day-to-day basis, and guided me through the world of information systems. He also took the lead in the assistance to students, for which I am very grateful to him. Thirdly, I have to thank Wim for the many philosophical discussions during lunch and dinner, and his cheerful way of making the office a nice workplace. Fourthly, it was great to have Maxime and Jan join our research team, and to witness their enthusiasm for methodology and business process management. Next, I would like to thank my colleagues from the neighbouring research unit, including Mario, Thomas, Christophe, and Jeroen. Special thanks goes to Mario, for who I have great respect in teaching, music and vision. Thanks to Mario, listening to The Smiths became an essential prerequisite for writing high-quality journal papers. Finally, I would like to thank the secretary team, including Martine (also known as mARTine or mar10) and Ann.

This PhD consists of three main parts, and while some people helped me with overall research aspects, others supported me in specific parts. To start with, I would like to thank Etienne Kerre, professor in mathematical analysis and fuzzy logic, for giving me the inspiration for my research topic. I was a student of Prof. Kerre during my master in computer science, and he showed me that the world is not black and white as many computer scientists prefer, but that there is a meaningful gradation between humans and computers. Next, I would like to thank Patrick Heymans, Monique Snoeck and Roland Paemeleire for sharing their overall insights in the field of requirements engineering and information systems. I learned a great deal of scientific insights from my collaboration with Monique, who guided together with Geert the first part of this PhD. For the second part of this PhD, I am grateful to Marwane El Kharbili for sharing his enthusiasm with me. I met Marwane

in 2008 at a semantic web conference in Tenerife, and we immediately connected both professionally and personally. For the validation of this second part, I would like to thank Patrick Van Kenhove, Cathy De Meulenaere, Griet Bonamie, Inge Van Doninck, Iris Decreus, and Sonia Van Pachterbeke. Finally, during the validation of the third part of my PhD, support came from Tom Vanacker, Evelyn Van der Hauwaert, Dirk De Mits and Pieter Bourgeois.

I want to show appreciation to the Vlerick school and the Vlerick Alumni for showing me that the impossible is always within reach. Firstly, I'm indebted to the Vlerick school for the fruitful discussions with Dirk Deschoolmeester, Stijn Viaene, and Peter Rittgen. When I initiated my PhD work, Dirk Deschoolmeester gave me a crash course in the history of information systems, and always tried to challenge me to make my research topic of added value to the business world. Dirk allowed me to have a meeting with one of his contacts, Basil Kritis, who participated with James Martin twenty years ago to launch Information Engineering, which had research goals comparable to those of my work. Next, Stijn Viaene helped me in understanding the importance of validity, and taught me to look critical to existing statements. I still recall one conversation, in which Stijn (rightly) levelled my ideas with the ground, which triggered me to validate my statements before making them. From this moment onwards, the quality of my papers grew immensely. Also, I'm grateful to Peter Rittgen for sharing his insights in requirements engineering and business process management, as his critical feedback impacted the third part of this PhD. Finally, I want to thank the president of the Vlerick Alumni Jean-Luc Deleersnyder for demonstrating the unique combination of Vlerick studies, doing a PhD, and working for a large consultancy.

In order to do my PhD at the Ghent University, Deloitte Consulting gave me a leave of absence of three years. Therefore, I want to thank Koen De Staercke, Christian Combes and Jurgen Moenaert for their confidence in me and in my work. Furthermore, I like to show appreciation to Aart Joppe, who guided me through my first consulting steps, and showed me that hard work can be combined with family life. Finally, Nick Van Maele provided me valuable lessons about my personal strengths and weaknesses - which came in handy during my PhD - and provided useful input to my research ideas.

During my master in general management at the Vlerick school, I met a lot of fellow students that shared my ambitions in life. After four years, I think it is safe to state that these fellow students had a true impact on my life, and many of them still do on a regular basis. Hence, I would like to thank the following friends: Maqsud, Denise, Blomme, Viane, Bart 'Pink Sweater' Cuypers, Vieze Freddy, Jens & Jeroen, Charlyboy, de Schrauwen, Kris, Wouter Danckaert, Maarten, Herv, Vandenab, Tom 'Vincent' Cruise, Dieter, Thomas DJ, Vande Cas, Antonio, Sjarel, Wouter Cuypers, Dries, Michiel, Claudio and Aldo.

I preserved a paragraph to thank friends and family who were 'uncategorizable' by means of proper labels. They were there when I needed them, and played an important role in my daily life.

- Den Den, who supported me unconditionally for the last fifteen years, and who stood next to me when I took difficult decisions. Together, we grew from boys to men. His vision and talent are yet to be discovered by the world, but I already reserved my front row seat at one of the future Cannes festivals.
- Soiffel, who is a great friend and guided me through life with em-

---

pathy, financial advice and a lot of alcohol. Our weekly telephone calls bridged the distance between Amsterdam and Ghent, and covered topics from Soiffel's last chest hair to international economy.

- Lode for his friendship and hospitality, as I learned that Gabon (Central Africa) has the world's highest consumption of champagne per capita. I will never forget singing Frank Sinatra's My Way on his terrace overlooking the magnificent African coast.
- My poker friends, including R Man, R Mann, Klama, Puydtje, Frittenboefer, and Freakus did a great job in entertaining me once a month, but unfortunately they still have to learn to let me win from time to time.
- Valerie, who supported me in my PhD vision. I still remember our evening at the Brussels' Metropol hotel, during which my PhD vision was conceived.
- Maqsud, who kindly borrowed a number plate 'DOKTER' from the parking lot of a general practitioner, and gave it to me as a present.
- Blomme, who motivated me during a night at the Walvis bar in Brussels. I am charmed by his humility and sense of realism, as illustrated by our midnight walk in Knokke.
- My fellow karatekas, such as Javier, Andy, Janie, Dirk, and Mark, showed me how to absorb negative experiences in a positive way, and made sure that surrendering is not a word in my dictionary. In retrospect, these acquired budo skills proved to be quite useful in the process of doing my PhD.
- As a friend, Javier took me under his wings, and taught me how to have confidence in life. His trust and support during the last seven years were crucial to become the man that I am today.
- My party and travel friends, including Den Den, Phisepeuk, Zeuben, Kim, Katrien, Den Bissamein, Heunk, Annelies, Kristof De Mey, Luckie and Kevin McHale, were always in the mood for a nice talk at nice places.
- My uncle Patrick and his family, including Olivia, Stijn, Melanie and Jeroen, for showing sincere interest in my PhD process.
- My rabbit Snuffelmans, who pre-tested my PhD document from top to bottom, and who said it was good material.
- My grandmother Alice, who still has a lot of youthful enthusiasm and encouraged me to work hard from the first till the last day of my PhD.
- My sister Iris and my future brother-in-law Pieter, who were always willing to listen to my ideas and nurtured me with proper meals and football tickets.
- My new mother Gina, who took me under her wings and treated me like a son, and who made sure that I survived the toughest moments of my PhD.

- Inke, who took great care of me by complementing my Bicky burger and Kapoentjes diet (from Julien) with fresh vegetables, who advised me to introduce softness in social relations and who tolerated my long working hours during weekdays, weekends and holidays, as even my current words are written on our terrace in Corfu while she is asleep.
- David Venter and his lovely wife Paula, who accepted me as a part of their family in South Africa, and supported me during my PhD process. As commonly recognized by Vlerick students, meeting David Venter is a life changing event. David recommends his students to read one book, i.e. *The Art of Possibility*. My PhD project embedded the principles of this book on a day-to-day basis, with surprising effects as a result. I cannot thank David enough for this advice, and for sharing his human insights with me.
- My mother Sonia and father Freddy, to who this PhD has been dedicated.

*Corfu, July 2010*  
*Schaarbeek, October 2010*  
*Ken Decreus*

# Table of Contents

<b>Doctoral Jury</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Acronyms</b>	<b>xxiii</b>
<b>Nederlandse samenvatting</b>	<b>xxv</b>
<b>Abstract</b>	<b>xxvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Context	1
1.2 GORE for BP Methods	5
1.3 Research Design	7
1.3.1 Problem Analysis	8
1.3.2 Solution Design	9
1.3.3 Solution Validation	10
1.4 PhD Structure	10
1.5 Publications	12
1.5.1 Publications in peer-reviewed international journals	12
1.5.2 Publications in peer-reviewed international conference proceedings	12
1.5.3 Publications in peer-reviewed international scientific workshops	13
1.5.4 Publications in PhD symposia related to international reviewed scientific conferences	13
1.5.5 Presentations at international scientific workshops	13
<b>2 Investigating Goal-Oriented Requirements Engineering for Business Processes</b>	<b>15</b>
2.1 Introduction	15
2.1.1 Challenges	16
2.1.2 Research Questions and Contributions	16
2.2 A Unifying Framework for GORE for BP Methods	18
2.2.1 A Metamodelling Technique for Describing GORE for BP Methods	21
2.2.2 The GORE for BP Method Metamodel	22
2.3 Identifying GORE for BP Methods	24
2.3.1 Develop and Validate Review Protocol	25
2.3.2 Selection of GORE for BP Papers	26
2.3.3 Assess Study Quality	29
2.4 Similarities and Differences of GORE for BP Methods	30
2.4.1 Research Design	31

2.4.2	Alleviating Threats to Validity . . . . .	32
2.4.3	Presentation of Results . . . . .	33
2.4.4	Answering Research Question RQ1.3 . . . . .	40
2.5	Conclusions . . . . .	41
<b>3</b>	<b>Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management</b>	<b>45</b>
3.1	Introduction . . . . .	45
3.1.1	Challenges . . . . .	46
3.1.2	Objectives, hypothesis and overview of the approach	47
3.2	Introduction to the Business Process Compliance Man- agement Example . . . . .	49
3.3	Policy-Enabled Goal-Oriented Requirements Engineering .	50
3.3.1	Formal Tropos . . . . .	51
3.3.2	Policy-extended Formal Tropos . . . . .	52
3.4	Detailed Implementation Architecture . . . . .	54
3.4.1	Implementation of Policy-enabled Formal Tropos . .	55
3.4.2	Implementation of Business Process Modelling On- tology . . . . .	56
3.4.3	Implementation of Transforming Policy-extended Formal Tropos to Business Process Modelling On- tology . . . . .	57
3.5	Evaluation . . . . .	58
3.5.1	Introduction . . . . .	58
3.5.2	Pilot Planning . . . . .	59
3.5.3	Pilot Design . . . . .	59
3.5.4	Pilot Conduct . . . . .	61
3.5.5	Pilot Evaluation . . . . .	63
3.6	Related Work . . . . .	64
3.7	Conclusion . . . . .	65
<b>4</b>	<b>Goal-Oriented Requirements Engineering for BPMN Mod- elling</b>	<b>67</b>
4.1	Introduction . . . . .	67
4.1.1	Introducing Our Approach . . . . .	68
4.1.2	Feasibility of our Approach . . . . .	70
4.1.3	Added Value of our Approach . . . . .	70
4.2	Theoretical Basis . . . . .	71
4.3	Overview of Our Approach . . . . .	72
4.3.1	B-SCP Metamodel . . . . .	73
4.3.2	B-SCP Editor . . . . .	75
4.3.3	Control Flow Annotations . . . . .	75
4.3.4	Model Transformation B-SCP to BPMN . . . . .	76
4.4	Demonstration of our Approach . . . . .	79
4.4.1	The Seven Eleven Japan (SEJ) Case Exemplar . . .	79
4.4.2	First Solution Property in Context of SEJ . . . . .	80
4.4.3	Second Solution Property in Context of SEJ . . . . .	86
4.4.4	Third Solution Property in Context of SEJ . . . . .	86
4.5	Evaluation studies . . . . .	90
4.5.1	Case Study Design . . . . .	90
4.5.2	Case Studies . . . . .	92
4.5.3	Findings for First Solution Property . . . . .	93
4.5.4	Findings for Second and Third Solution Properties .	95
4.6	Conclusion . . . . .	97



---

<b>5 Conclusions</b>	<b>99</b>
5.1 Answering the Research Questions . . . . .	100
5.2 Implications . . . . .	105
5.2.1 Implications for Practitioners . . . . .	106
5.2.2 Implications for Researchers . . . . .	108
5.3 Limitations and Future Work . . . . .	110
<b>References</b>	<b>111</b>
<b>A Appendices for Chapter 3</b>	<b>131</b>
<b>B Appendices for Chapter 4</b>	<b>149</b>
<b>C BSCP2BPMN Details</b>	<b>153</b>
<b>D Graphical Editor Details</b>	<b>165</b>



# List of Figures

1.1	Today's Information System Portfolio Model [HSFM06] . . .	3
1.2	Information System Portfolio of the Future [HSFM06] . . .	3
1.3	Overall GORE for BP Architecture . . . . .	4
1.4	Business Motivation Model of the Object Management Group [OMG09b] . . . . .	5
1.5	Conceptual Modelling Framework (Adapted from Wand and Weber [WW02]) . . . . .	6
1.6	Engineering Cycle and Research Cycle [WH06] . . . . .	7
1.7	Overview of PhD Structure . . . . .	11
2.1	GORE Methods versus GORE for BP Methods . . . . .	18
2.2	The seven GORE for BP knowledge-modelling states (adapted from Decreus et al. [DKPP09]) . . . . .	20
2.3	Overview of a Metamodelling Technique (Adapted from [RSM95, RNG99, RPB99]) . . . . .	21
2.4	The GORE for BP Method Metamodel (Based on Kavakli [KL99b, Kav02]) . . . . .	23
2.5	The Soffer and Rolland way-of-working . . . . .	24
2.6	The Aburub et al. [AOB07, CdPLdMSN01] way-of-working	34
2.7	The Bleistein et al. [BCVP06a, BACR04, BCV04c, BCV04a, BCV06, BCVP06b, BCV04b, BCV05, CPBV05, CP03] way-of-working . . . . .	34
2.8	The Frankova et al. [FMS07b, FYS07, FMS07a, SHF08, GMMZ06, MMZ08] way-of-working . . . . .	35
2.9	The Grau et al. [GFM05, GFA06, GFM08, JM04] way-of-working . . . . .	36
2.10	The Koliadis et al. [KVB <sup>+</sup> 06b, KVB <sup>+</sup> 06a, FLM <sup>+</sup> 04] way-of-working . . . . .	36
2.11	The Lapouchnian et al. [LL06a, LYM07, SGM04] way-of-working . . . . .	37
3.1	Implementing Business Requirements through Business Process Architectural Layer . . . . .	46
3.2	Implementing Policy-Enabled Goal-Oriented Requirements Engineering . . . . .	48
3.3	Policy Model Related to Extraction Business Process . . . . .	50
3.4	First Formal Tropos Example . . . . .	51
3.5	Second Formal Tropos Example . . . . .	51
3.6	Formal Tropos Grammar [FLM <sup>+</sup> 04] . . . . .	52
3.7	Policy-extension to Formal Tropos Grammar . . . . .	53
3.8	Part of Policy-extended Formal Tropos for Company X (Extract from Appendix A - Figure A.1) . . . . .	53
3.9	Eclipse-based Visual Policy-extended Formal Tropos Editor	54
3.10	Implementation Architecture of Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management . . . . .	55

3.11	Employing the Graphical Modelling Framework to Implement Policy-extended Formal Tropos . . . . .	55
3.12	Graphical Output for Company X . . . . .	57
3.13	Part of Textual Output for Company X (Figure A.5) . . . . .	58
4.1	Overview of GORE for BPMN . . . . .	72
4.2	B-SCP Metamodel in Ecore . . . . .	73
4.3	Eclipse-based Visual B-SCP Editor . . . . .	75
4.4	Using OR Decomposition . . . . .	76
4.5	Setting AND Decomposition to Sequence . . . . .	76
4.6	Setting AND Decomposition to Parallel . . . . .	76
4.7	Resulting BPMN diagram for the B-SCP tactic ‘Encourage rental extensions’ . . . . .	79
4.8	BSCP diagram for SEJ . . . . .	83
4.9	BPMN Model Resulting From Problem Diagram C . . . . .	87
4.10	BPMN Model Resulting From Problem Diagram D . . . . .	89
5.1	Interactions between chapters . . . . .	100
5.2	Scylla and Charybdis [Jac00] . . . . .	106
5.3	Positioning BPM to Scylla and Charybdis . . . . .	107
A.1	Policy-extended Formal Tropos Specification Generated by Visual Editor . . . . .	132
A.2	Ecore metamodel of Policy-extended Formal Tropos . . . . .	133
A.3	Ecore metamodel of Business Process Modelling Ontology . . . . .	134
A.4	PFT2BPMO Model Transformations . . . . .	135
A.5	Output of Atlas Transformation Language Mappings for Company X . . . . .	136
A.6	Policy-enabled business process model (with regular business process modelling tool) . . . . .	137
A.7	Policy-extended Formal Tropos model . . . . .	138
A.8	Policy-enabled business process model (based on Policy-extended Formal Tropos model) . . . . .	139
A.9	Subsection of Textual Part for Figure A.8 . . . . .	140
A.10	Policy-enabled business process model (with regular business process modelling tool) . . . . .	141
A.11	Policy-extended Formal Tropos model . . . . .	142
A.12	Policy-enabled business process model (based on Policy-extended Formal Tropos model) . . . . .	143
A.13	Subsection of Textual Part for Figure A.12 . . . . .	144
A.14	Policy-enabled business process model (with regular business process modelling tool) . . . . .	145
A.15	Policy-extended Formal Tropos model . . . . .	146
A.16	Policy-enabled business process model (based on Policy-extended Formal Tropos model) . . . . .	147
A.17	Subsection of Textual Part for Figure A.16 . . . . .	148
B.1	The ATL Implementation of the BSCP2BPMN Model Transformations . . . . .	150
C.1	Example of BPMN MessagingEdge Structure (Part of Figure C.7) . . . . .	155
C.2	Problem Diagram B . . . . .	156
C.3	XML serialization of Problem Diagram B . . . . .	157
C.4	The BPMN metamodel . . . . .	158
C.5	Generated BPMN skeleton) . . . . .	159

---

C.6	Added Hierarchical Structure to BPMN skeleton . . . . .	160
C.7	Completed BPMN diagram . . . . .	161
C.8	Generate diagram file with EMF . . . . .	162
C.9	Completed BPMN Diagram (visualized in BPMN Editor) .	163
D.1	The BSCP Ecore Metamodel . . . . .	167
D.2	New GMF Project . . . . .	168
D.3	BSCP Ecore Metamodel included in GMF Project . . . . .	168
D.4	BSCP Domain Gen Model included in GMF Project . . . . .	169
D.5	BSCP Graphical Def Model included in GMF Project . . . . .	169
D.6	BSCP Tooling Def Model included in GMF Project . . . . .	170
D.7	BSCP Mapping Model included in GMF Project . . . . .	170
D.8	Menu to Create BSCP Generator Model . . . . .	171
D.9	BSCP Generator Model included in GMF Project . . . . .	172
D.10	Menu to Generate Editor Code . . . . .	173
D.11	Run Configuration for BSCP Editor . . . . .	174
D.12	Select BSCP diagram with Wizard . . . . .	175
D.13	Use the BSCP Editor . . . . .	176



# List of Tables

2.1	Initial Search Terms Used . . . . .	25
2.2	Quality Metrics in terms of Completeness of Documentation	27
2.3	Methods Ordered by Completeness . . . . .	29
2.4	Measuring GORE for BP Similarities and Differences . . .	38
3.1	PFT2BPMO Concept Mappings . . . . .	57
3.2	Pilot Study Results . . . . .	63
4.1	BSCP2BPMN Concept Mappings . . . . .	78
4.2	Phenomena of the SEJ problem description . . . . .	84
B.1	Data Format for Case Study Findings . . . . .	151





# List of Acronyms

## A

ACM	Association for Computing Machinery
ACM SAC	Association for Computing Machinery Symposium on Applied Computing
ATL	Atlas Transformation Language

## B

BMM	Business Motivation Model
BPEL	Business Process Execution Language
BP	Business Process
BPM	Business Process Management
BPMDS	Business Process Modelling, Development and Support
BPMN	Business Process Modelling Notation
BPMO	Business Process Modelling Ontology
BPMS	Business Process Management System
BPO	Business Process Orientation
BPR	Business Process Reengineering
B-SCP	Business Strategy Context Process
BSCP2BPMN	Business Strategy Context Process to Business Process Modelling Notation

## C

CAiSE	Conference on Advanced Information Systems Engineering
CEO	Chief Executive Officer
CDC	Combined Delivery Centre
CFO	Chief Financial Officer
COMPSAC	Computer Software and Applications Conference

## D

DIS	Detailed Interaction Script
-----	-----------------------------

**E**

EDOC	Enterprise Distributed Object Computing
EKD	Enterprise Knowledge Development
EMF	Eclipse Modelling Framework
EMMSAD	Exploring Modelling Methods for Systems Analysis and Design
EPC	Event-driven Process Chains
ER	Entity Relationship

**F**

FS	Franchise Store
FSC	Franchise Store Computer
FT	Formal Tropos

**G**

GMF	Graphical Modelling Framework
GO-BPMN	Goal-Oriented Business Process Modelling Notation
GORE	Goal-Oriented Requirements Engineering
GORE for BP	Goal-Oriented Requirements Engineering for Business Processes
GOT	Graphical Order Terminal
GPM	Generic Process Model

**H**

HS	Handheld Scanner
----	------------------

**I**

IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IT	Information Technology

**K**

KAOS Knowledge Acquisition in autOmedated Specification

**M**

MDA Model Driven Architecture

**N**

NFR Non-Functional Requirements

**O**

OMG Object Management Group

**P**

PFT Policy-extended Formal Tropos  
PFT2BPMO Policy-extended Formal Tropos to Business Process Modelling Ontology  
POS Point of Sale  
PRiM Process Reengineering i\* Method

**R**

RAD Role Activity Diagrams  
RE Requirements Engineering  
REBPM Requirements Engineering for Business Process Management  
RESCUE Requirements Engineering with SCenarios in User-centred Environments  
RQ Research Question

**S**

SAP	SystemAnalyse und Programmentwicklung
SBPM	Semantic Business Process Management
SEJ	Seven Eleven Japan
SEJ HC	Seven Eleven Japan Host Computer
SSC	Shared Service Centre
ST	Scanner Terminal
SUPER	Semantics Utilised for Process management within and between EnteRprises

**U**

UML	Unified Modeling Language
-----	---------------------------

**V**

VMOST	Vision Mission Objectives Strategy Tactics
-------	--

**W**

WS	Weather Service
WSMO	Web Service Modelling Ontology





# Nederlandse samenvatting

## –Summary in Dutch–

In de laatste decennia zijn de onderzoeksgebieden *Requirements Engineering* (RE) en *Business Process Management* (BPM) beetje bij beetje naar elkaar toe gegroeid. De algemene doelstelling van ons doctoraatsonderzoek is om beide onderzoeksgebieden nog dichter bij elkaar te brengen, door middel van het gemeenschappelijke concept ‘doel’ te gebruiken. Aangezien bedrijfsprocessen bepaalde doelen van klanten vervullen, is het de moeite waard om te onderzoeken of een doelenmodel kan verbonden worden met een bedrijfsprocesmodel. De focus van ons onderzoek ligt bij een bepaald type van methodes, meerbepaald *Goal-Oriented Requirements Engineering for Business Processes* (GORE for BP). Deze *GORE for BP* methodes dienen om een bedrijfsgebruiker te ondersteunen tijdens het ontwerp van bedrijfsprocesmodellen gebaseerd op doelen en andere strategische vereisten.

In het eerste stuk van dit doctoraat, de probleemanalyse, verwerken we diepere kennis over de bestaande *GORE for BP* methodes. Ten eerste, we introduceren een overkoepelend raamwerk voor *GORE for BP* methodes om bestaande *GORE for BP* methodes op een consistente manier te kunnen voorstellen. Ten tweede, we gaan op zoek naar de huidige *GORE for BP* methodes, and kiezen de methodes die volledig gedocumenteerd zijn. Ten derde, we onderzoeken de gelijkenissen en verschillen tussen de *GORE for BP* methodes door middel van het overkoepelend *GORE for BP* raamwerk.

In het tweede stuk van dit doctoraat, het ontwerp van de oplossing, proberen we de huidige *GORE for BP* methodes te verbeteren om bedrijfsgebruikers gemakkelijker bedrijfsprocessen te laten ontwerpen aan de hand van strategische vereisten. Hiervoor gebruiken we de inzichten van de probleemanalyse om de beste *GORE for BP* methodes te vinden, en lossen we de grootste problemen op van deze methodes. Ons werk heeft twee toepassingsgebieden: (1) semantische bedrijfsprocesmodellen die strategische vereisten en beleid voorstellen, en (2) complexe bedrijfsprocesmodellen die strategische vereisten hebben. De semantische bedrijfsprocesmodellen zijn te situeren in het SUPER project, dat toelaat om bedrijfsprocesmodellen te annoteren met ontologische informatie. De complexe bedrijfsprocesmodellen betreffen levensechte bedrijfsprocessen in onze huidige ondernemingen, waarbij meer dan tweehonderd knopen en relaties getekend worden.

In het derde stuk van dit doctoraat, de validatie van de oplossing, onderzoeken we of onze *GORE for BP* methodes echt voordelen geven voor een bedrijfsgebruiker. Om te beginnen tonen we de haalbaarheid aan van onze *GORE for BP* methodes, door middel van deze methodes toe te passen op gekende bedrijfssituaties. Vervolgens onderzoeken we de toegevoegde waarde van onze aanpak, waarbij we kijken naar de voordelen voor een bedrijfsgebruiker om bedrijfsprocesmodellen te ontwerpen op basis van strategische vereisten, in vergelijking met het ontwerpen van bedrijfsprocesmodellen zonder onze hulp.





# Abstract

In the last decade, the Requirements Engineering (RE) and Business Process Management (BPM) research fields have slowly but steadily grown towards each other. The overall objective of our PhD research is to further bridge the gap between RE and BPM by focusing on the common notion of goal in both areas of research. As business processes are meant to fulfill through their execution certain organizational goals, it is worth investigating whether goal models that document and analyse organizational strategies, could play an active role in the design of business processes. The focus of our research is Goal-Oriented Requirements Engineering for Business Processes (GORE for BP) methods that support business users to design business processes based on goals and other strategic requirements.

In the first part of this dissertation, the problem analysis, we acquire knowledge about existing GORE for BP methods. First, we propose a unifying framework of a GORE for BP method in order to represent all existing GORE for BP methods in the same way. Second, we discover the currently existing GORE for BP methods, and select those GORE for BP methods that are sufficiently complete in terms of documentation. Third, we investigate trends and challenges of GORE for BP methods by comparing the discovered GORE for BP methods to the unifying framework.

In the second part of the dissertation, the solution design, we improve the current GORE for BP methods in order to enable business users in designing business processes that correspond to strategic requirements. More specifically, we use the insights that we gained during our problem analysis to select the current best GORE for BP methods, and we address discovered gaps in the selected GORE for BP methods. Our work focuses on two application areas: (1) semantic business process models that relate to strategic requirements and policies, and (2) complex business process models that relate to strategic requirements. The semantic business process models refer to the SUPER project, which annotates business process models by ontologic information in order to increase the semantics in the business process models. The complex business process models refer to real-world business process diagrams with over two hundred nodes and relations in total.

In the third part of the dissertation, the solution validation, we investigate whether our GORE for BP methods improve the effective usage of GORE for BP methods by business users. First, we demonstrate the feasibility of our GORE for BP methods by applying our approach to case exemplars. Second, we investigate the added value of letting business users design business processes with our GORE for BP methods as compared to business users designing business processes without our support.



# 1

## Introduction

### 1.1 Research Context

In 1776, Adam Smith described the concept that industrial work should be broken into its simplest tasks [MJ00]. This idea became the basic organisation model of business for almost two hundred years, and is our tightly defined, tightly controlled, functionally centred organisation model of today. The functional view of the organisation is best described by the organisation chart, which shows the people that have been grouped together for operational efficiency. Such an organisation chart illustrates reporting relationships, but excludes the customer in the overall picture and frequently lacks coordination of hand-offs between functions. Authors such as Deming [Wal88], Imai [Ima86], Porter [PM85], Drucker [Dru88], Hammer [HC94], and Davenport [Dav93] have all defined what they view as the new model of the organisation. Companies are forced to become fast, flexible, and participative and must focus on customers, competition, teams, time and processes. This new way of thinking or viewing the organisation has been defined by McCormack [MJ00] as business process orientation (BPO).

In the 1980s, the foundations of the BPO principle were developed. First and foremost, there was Edward Deming [Wal88] who developed the Deming Flow Diagram depicting the horizontal connections across a firm, from the customer to the supplier, as a process that could be measured and improved like any other process. Next, Masaaki Imai [Ima86] revealed that the *kaizen* principle was the single most important concept in Japanese management. The major philosophical characteristic of *kaizen* is one of continuous improvement of everything, every day and involving everyone. Imai stressed that management must adopt a process-oriented way of thinking. Further on, Michael Porter [PM85] introduced the concepts of interoperability across the value chain and horizontal organisation as major strategic issues within firms. The recognition of the linkages between functions, according to Porter, has been strongly influenced by Japanese management practices. Finally, Peter Drucker [Dru88] foresaw that the availability of information would

transform the organisational structure into a flat organisation of specialists working on task-focused teams. As all work will be done in cross-functional teams, a sequence of tasks with hand-offs between functional groups will not any longer exist.

In the 1990s, the BPO principle was augmented by the usage of Information Technology (IT). Initially, Michael Hammer [HC94] introduced Business Process Reengineering, which was accomplished by looking at fundamental processes of the business from a cross-functional perspective and would enable a radical new way of operating, using information technology as enabler. Hammer defined a business process as a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A reengineered business is composed of strategic, customer-focused processes that start with the customer and emphasize outcome, not mechanisms. Next, Thomas Davenport [Dav93] provided the foundation for technology-oriented BPO by describing the needed revolutionary approach to information technology in business. This approach was new in how business was viewed, structured and improved. Davenport defined a business process as a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs. In this perspective, the existing hierarchical structure is a slice-in-time view of responsibilities and reporting relationships, whereas a process structure is a dynamic view of how an organization delivers value.

Since 2000, the big-bang approach in Business Process Reengineering made place for a smoother and more iterative method called Business Process Management (BPM). Gartner Research [HSFM06] defines BPM as a structured approach that employs methods, policies, metrics, management practices and software tools to manage and continuously optimize an organization's activities and processes. The software tools to support BPM play an important role, as business users can employ them to control and modify their processes, including manual and automated tasks. One of the main BPM-enabling technologies is the Business Process Management System (BPMS), which Smith and Fingar [SF04] define as a modelling, integrational, and executional environment for the design, manufacture and maintenance of business processes. The importance of BPMS is illustrated by Gartner Research [WG06], which estimates that by 2015 30% of business applications will be developed by means of BPMS technology. Today, the market generally has accepted (Figure 1.1) that packaged applications enable process efficiency (75%), but it also states that some degree of custom build is necessary (25%). Estimated in 2015 (Figure 1.2), companies will decreasingly buy traditional business process applications (60%) and build less custom-made applications (10%), while more business process applications will be developed via BPMS technology (30%).

The popularity of BPMS technology blurs the boundaries between business process modelling (lead by business departments or business process managers) and software development (lead by IT department). Traditionally, business departments use graphical business process models to facilitate human understanding, communication, the improvement or management of processes [CKO92], whereas the IT department collects requirements of desired software applications, and aims at de-



Figure 1.1: Today's Information System Portfolio Model [HSFM06]

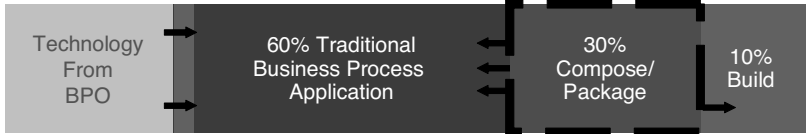


Figure 1.2: Information System Portfolio of the Future [HSFM06]

signing, developing, validating and maintaining these software applications [Som04]. Now, BPMS technology offers a common environment to business users and IT-oriented users, in which business users are responsible for creating the (high-level) business process models, and in which IT-oriented users refine these business process models by making them complete and consistent. In such a setting, there is a convergence between modelling the actual business process and gathering the requirements of the software application (which is in this case the business process model itself).

In other words, the traditional fields of research in Requirements Engineering (RE) and Business Process Modelling (part of BPM) are growing towards each other. The overall objective of our research is to bridge the gap between RE and BPM by focusing on the common notion of *goal* in both areas of research. As business processes are meant to fulfil through their execution certain organizational goals, it is worth investigating whether goal models that document and analyse organizational strategies, could play an active role in the design of business processes. Therefore, the idea of Goal-Oriented RE for Business Processes (GORE for BP) will be the topic of this PhD.

From an architectural point of view, GORE for BP can be represented by means of four layers (Figure 1.3). On the top layer, a *Requirements Model* is used to represent the strategic needs of an organisation (more details about these strategic needs is given in the next paragraph). The second layer in the GORE for BP architecture (Figure 1.3) contains *Business Process Models*, where the models reside that document the business processes, corresponding to the *Requirements Model* of the upper layer. The third layer deals with *Executable Business Process Models*, which are executable versions of the business process models on the layer above, and which contain all technical information that is needed to run these processes by means of software. The bottom layer contains the IT infrastructural services (e.g. web services, service-oriented software applications) that are used by executable business processes in order to be run.

As GORE for BP combines terminology from different research areas, we will start with defining the main GORE for BP concepts. To

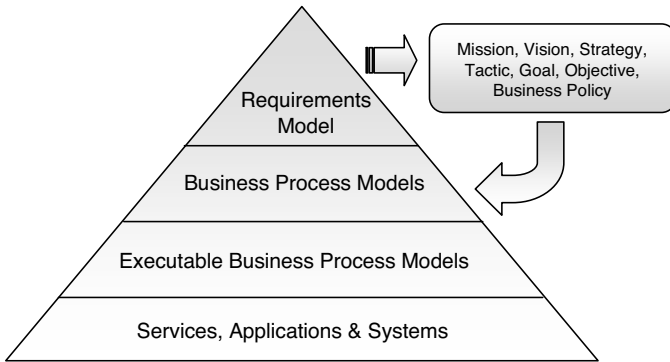


Figure 1.3: Overall GORE for BP Architecture

start with, we define a *business process* as a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer [HC94] (e.g., organisation EU-Rent provides value to the customer by means of business process ‘Rent Car to Customer’). Next, we define *requirements* for business processes as the overall set of requirements that relate to business processes as given by the Business Motivation Model (BMM) [OMG09b] of the Object Management Group (OMG), such as vision, mission, strategy, tactic, goal, objective and business policy. More specifically:

- A *vision* describes the future state of the enterprise, without regard to how it is to be achieved (e.g., Be the car rental brand of choice for business users in the countries in which EU-Rent operates).
- A *mission* indicates the ongoing activity that makes the vision a reality (e.g., Provide car rental service across Europe and North America for both business and personal customers).
- A *goal* indicates what must be satisfied on a continuing basis to effectively attain the vision (e.g., To be a premium brand car rental company, positioned alongside companies such as Hertz and Avis).
- A *strategy* is a long-term activity designed to achieve a goal (e.g., Operate nation-wide in each country of operation, focusing on major airports, competing head-to-head, on-airport, with other premium car rental companies).
- An *objective* is a specific and measurable statement of intent whose achievement supports a goal (e.g., By end of current year, be rated by AC Nielson in the top 6 car rental companies in each operating country within the European Community).
- A *tactic* is a short-term action designed to achieve an objective (e.g., Encourage rental extensions).
- A *business policy* is a non-actionable directive whose purpose is to govern, guide and shape the business processes of an organisation

(e.g., Rental contracts must be made under the law of the country in which the pick-up branch is located).

More general, an *end* is something the business seeks to accomplish, such as a *vision*, a *goal* or an *objective*. A *means* is a way to achieve an end, such as a *mission*, a *strategy*, a *tactic* or a *business policy*. The relationships between these concepts are given by Figure 1.4:

- A business process *is governed by* a business policy
- A business process *realizes* a strategy or a tactic
- A strategy or a tactic *channels efforts towards* a goal or an objective

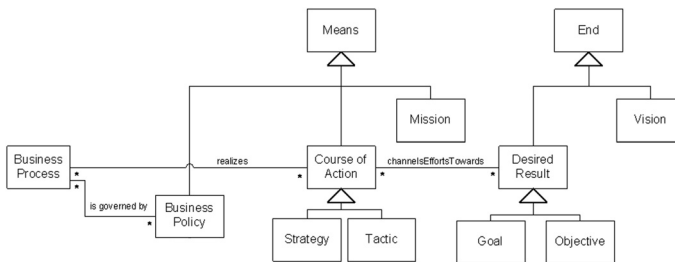


Figure 1.4: Business Motivation Model of the Object Management Group [OMG09b]

## 1.2 GORE for BP Methods

In the field of Information Systems, the task of conceptual modelling involves building a representation of selected phenomena in some domain [WW02]. In order to set the scene, we will introduce a framework for research on conceptual modelling (adapted from Wand and Weber [WW02]). This framework, displayed by Figure 1.5, comprises four elements: conceptual modelling *metamodel*, conceptual modelling *methods*, conceptual modelling *models* (in brief: conceptual models) and conceptual modelling *contexts*.

- A conceptual modelling *metamodel* provides a set of constructs and rules that show how to combine the constructs to model real-world domains. For instance, the metamodel of Business Process Modelling Notation (BPMN) [OMG09a] has the constructs ‘activity’ and ‘sequence flow’ to model real-world activities and sequences of these activities. An example of a rule in the BPMN metamodel is that two activities can only be associated via a sequence flow.
- A conceptual modelling *method* provides procedures by which a metamodel can be used. For instance, Bruce Silver [Sil09] describes how process modellers should employ BPMN during real-world modelling projects. Just as recipes in a cookbook explain

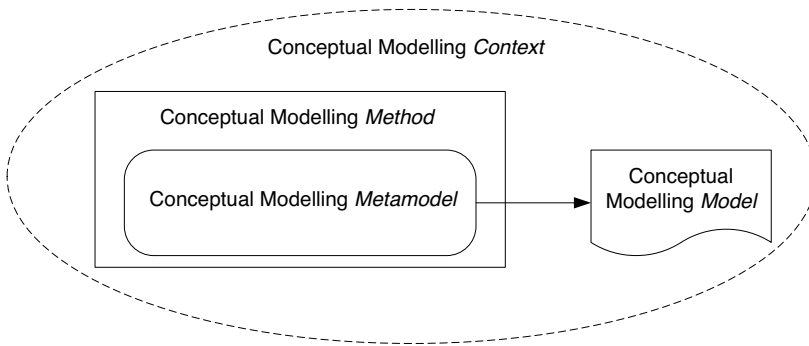


Figure 1.5: *Conceptual Modelling Framework (Adapted from Wand and Weber [WW02])*

how to use the ingredients in a proper way, Silver [Sil09] recommends modellers to define the process scope, create the top-level BPMN model in the best-case scenario, add the exception paths to the best-case BPMN scenario, etc.

- A conceptual modelling *model* is the product of a conceptual modelling method. For instance, when a modeller employs Silver’s BPMN method [Sil09], an initial output is the top-level BPMN model that represents the best-case scenario, and a subsequent output might be a top-level BPMN model that contains both best-case and worst-case scenario.
- A conceptual modelling *context* is the setting in which conceptual modelling occurs, and in which the resulting models are used. In Silver’s BPMN method [Sil09], three stakeholders are involved: the business user, the business analyst, and the IT developer. First, the business user is a domain expert with a deep understanding of real-world business processes, and is interested in modelling some of these business processes by means of BPMN. Second, the business analyst is responsible for making BPMN models complete and consistent with respect to the BPMN meta-model. Third, the IT developer wants to deploy the resulting (i.e. consistent and complete) BPMN model in a BPMS process engine, and focuses on the implementation aspects of BPMN models.

The first publication of a GORE for BP method occurred in 1999, when Kavakli and Loucopoulos [KL99b] introduced an approach of goal-driven business process modelling in the context of the de-regulation of a large European electricity company. Such GORE for BP methods offer *cookbook*-like support to business users to design business processes based on goals and other strategic requirements. Our research is focused on *business users*, instead of business analysts or IT developers, as both BPMN and BPMS are technology-driven initiatives and often lack support from the business perspective. Although other roles such as business process experts and facilitators play an important role in real-world projects, we exclude them from the scope of our research to focus on business users. We took this decision because - in line with the existing GORE for BP literature - we assume that the business user is



empowered to express his own understanding of business processes and corresponding requirements.

One of the main complexities in GORE for BP *methods* is the multitude of underlying *metamodels*, as some originate from goal modelling, others originate from business process modelling, and still others originate from goal-oriented business process modelling. For the last ten years, there has been an explosion of diverting GORE for BP research, which failed to properly support business users in designing business processes based on strategic requirements. Especially in the field of RE, it is well-known that the research results of the last decades had little impact on the industrial practice [KBBJ<sup>+</sup>02, HDK93]. For this reason, we decided not to introduce *yet another RE method*, but to investigate the shortcomings and open issues associated with the use of the currently proposed GORE for BP methods and to improve the existing GORE for BP methods that seem the most appropriate for our purpose.

### 1.3 Research Design

The research design of this PhD is mainly structured via the conceptual framework of Wieringa [WH06, Wie96, WMMR06, Wie05]. The basic distinction in this framework is one between knowledge problems and world problems. A *knowledge problem* consists of a lack of knowledge about the world. To solve a knowledge problem, we need to change the state of our knowledge, and by doing this, we try not to change the world. *World problems* consist of a difference between the way the world is and the way we think it should be. We solve a world problem by trying to change the state of the world: we change an organisation, we build a device, we implement a program, etc. There is a mutually recursive relation between knowledge problems and world problems, as there might be a world problem in the background of a knowledge problem and vice versa. Overall, the distinction between world problems and knowledge problems relates to different goals: when our goal is to solve a knowledge problem, we do something to acquire that knowledge and, if necessary, manage the changes in the world in a way that the acquired knowledge is valid. And when our goal is to solve a world problem, we change the world and are usually happy with any knowledge we gain from this.

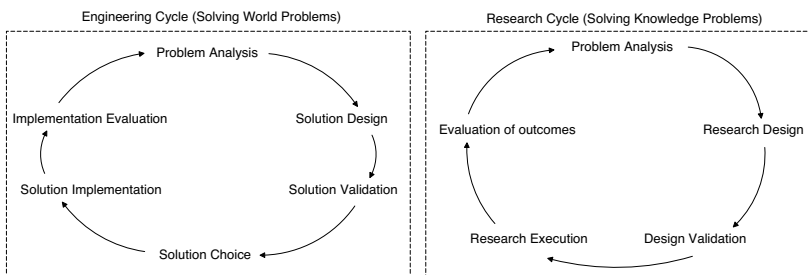


Figure 1.6: *Engineering Cycle and Research Cycle [WH06]*

The reason for distinguishing world problems from knowledge prob-

lems is that the rational methods to solve them are different (Figure 1.6). The *engineering cycle* is a collection of tasks that a rational problem solver would follow to *solve a world problem*. To solve a world problem rationally, we would investigate the problem (Problem Analysis), design a solution that tries to solve the investigated problem (Solution Design), validate whether our designed solution improves the problem (Solution Validation), allow stakeholders to select the best solution for the problem (Solution Choice), realize the chosen solution in the real world (Solution Implementation), and evaluate the implemented solution to discover whether it really solved the problem as expected (Implementation Evaluation). On the other hand, the *research cycle* is the collection of tasks that a rational problem solver would follow to *solve a knowledge problem*. To solve a knowledge problem rationally, we would figure out what we want to know (Problem Analysis), design the research protocol by which we want to acquire this knowledge (Research Design), validate the research protocol (Design Validation), do the research as designed (Research Execution), and relate the acquired knowledge to the original research questions and context (Evaluation of outcomes).

This PhD is constituted by a nested hierarchy of world problems and knowledge problems. Overall, this PhD solves a *world problem*, as we try to improve the current GORE for BP methods, such that business users can employ these GORE for BP methods to design business processes based on strategic requirements. Hence, the *overall structure of this PhD* is defined by the *engineering cycle*. As Wieringa and Heerkens explain [WH06], a PhD project in the field of RE typically analyses a problem, proposes a new solution and does a proof-of-concept validation, but excludes further steps as “it may take two more years to do a proper validation, and an additional ten years to implement it in the real world and evaluate its use” ([WH06], p300). For this reason, the overall structure of this PhD consists of the first three steps of the engineering cycle, i.e. problem analysis, solution design, solution validation.

Then, each of the three steps tackles a new world problem or new knowledge problem. The first part of this PhD, the *problem analysis*, is a *knowledge problem* because we want to acquire knowledge about the problems of the current GORE for BP methods. The second part of it, the *solution design*, is a *world problem* as we want to improve current GORE for BP methods to address the discovered problems. The third part of this PhD, the *solution validation*, is then again a *knowledge problem* because we want to understand the feasibility and added value of our solution design. The following sections introduce the design of our problem analysis (Section 1.3.1), our solution design (Section 1.3.2), and our solution validation (Section 1.3.3).

### 1.3.1 Problem Analysis

During the problem analysis, we want to investigate the challenges for effective usage of GORE for BP methods. First, we are interested in understanding the unifying framework of a GORE for BP method in order to represent all existing GORE for BP methods in the same way. Second, we want to discover the currently existing GORE for BP methods, and select those GORE for BP methods that are sufficiently complete in

terms of documentation, in order to reduce the subjective interpretation of the identified methods as much as possible. Third, we want to discover trends and challenges of GORE for BP methods by comparing the discovered GORE for BP methods to the unifying framework. In order to deal with this matter, we address the following research questions:

- RQ1.1 What is the unifying framework of a GORE for BP method?
- RQ1.2 Which are the currently available GORE for BP methods that are sufficiently *complete* in terms of documentation, to allow for comparison with other methods and investigation of strengths and weaknesses?
- RQ1.3 What are the similarities and differences between these *complete* GORE for BP methods in relationship to the unifying framework addressed by RQ1.1?

### 1.3.2 Solution Design

During the solution design, we want to improve the current GORE for BP methods in order to enable business users in designing business processes that correspond to strategic requirements. More specifically, we will use the insights that we gained during our problem analysis to select the current best GORE for BP methods, and we will address discovered gaps in the selected GORE for BP methods.

In this phase, the research was done in an iterative way. Initially, our research focused on the SUPER project [SUP07], that tried to close the business-IT gap by means of annotating semantics into business process models. Within that SUPER project, there was an interest to relate goals and policies into business process models, by means of the semantic business process modelling formalism called Business Process Modelling Ontology (BPMO). Using the BPMO, elements from organisational ontologies could be incorporated into business process models. For instance, traditional business process models embed information about which actor is responsible for a task, and in what sequence these tasks have to be executed. Making these traditional business process models ‘semantic’ allowed modellers to embed extra information in the model, such as the organisational role of the actor (e.g. Sales Manager) or the business goal that the activity should achieve (e.g. Reduce maverick spending by 5%). More specifically, we investigated how an existing goal-oriented modelling language (Formal Tropos [FLM<sup>+</sup>04]) could be transformed into semantically-enriched BPMO business processes.

The research in the context of the SUPER project resulted into the work that is presented in Chapter 3. Based on this first iteration, we discovered major challenges of GORE for BP methods in real-world, complex settings (i.e., business process diagrams with over two hundred nodes and relations). As a result, we reduced the research scope (excluding the SUPER project and policies), and extended the level of detail of our research, which resulted in the work presented in Chapter 4. During this second iteration, we used the insights of our problem analysis in chapter 2, and replaced Formal Tropos [FLM<sup>+</sup>04] by means of B-SCP [BCVP06b], and replaced the business process models of SU-

PER [SUP07] by BPMN [OMG09a]. In order to deal with these two iterations, this PhD addresses the following research questions:

- RQ2.1 How can business users design semantic business processes in terms of and in correspondence with strategic requirements and policies?
- RQ2.2 How can business users design complex business processes in terms of and in correspondence with strategic requirements?

### 1.3.3 Solution Validation

During the solution validation, we want to know whether our GORE for BP methods improve the effective usage of GORE for BP methods by business users. First, we will focus on demonstrating the *feasibility* of our GORE for BP methods, by applying our approach to case exemplars. Second, we are interested in discovering the *added value* of letting business users design business processes with our GORE for BP methods as compared to business users designing business processes without our support. In order to deal with this matter, we address the following research questions:

- RQ3.1 Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.1?
- RQ3.2 What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.1?
- RQ3.3 Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.2?
- RQ3.4 What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.2?

## 1.4 PhD Structure

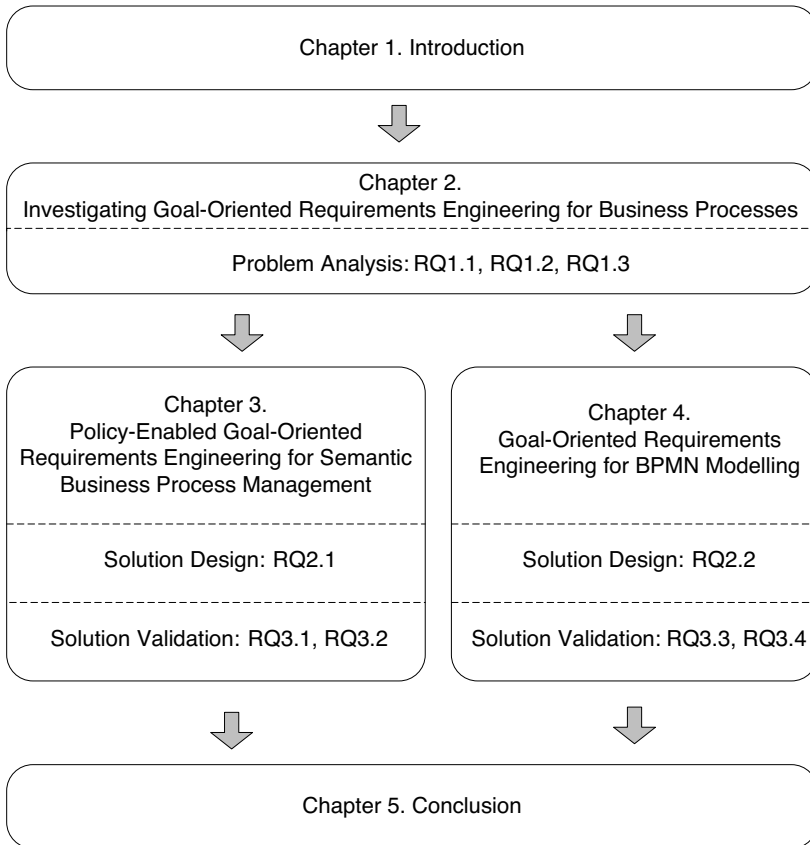
A visual overview of the PhD structure is given by Figure 1.7. It consists of three main parts, i.e. the introduction (Chapter 1), the body of the PhD (Chapter 2, 3 and 4), and the conclusion (Chapter 5). Each chapter of the PhD body was written as a self-contained research paper, so these chapters can be read independently from other PhD chapters. This introductory chapter explains the relations between the research questions, and the chapters in this PhD (that provide an answer to these research questions).

- **Chapter 1. Introduction**

This chapter defines the structure of the research questions, which are answered in Chapter 2, Chapter 3, and Chapter 4.

- **Chapter 2. Investigating Goal-Oriented Requirements Engineering for Business Processes**

This chapter deals with the problem analysis of this PhD, by investigating GORE for BP methods from different perspectives. Research question RQ1.1 is answered in section 2.2, where we



*Figure 1.7: Overview of PhD Structure*

propose a unifying framework for GORE for BP methods. Then, we report about the results of our systematic literature review in section 2.3 to answer RQ1.2, and discuss the similarities and differences of the discovered GORE for BP methods in section 2.4 to answer RQ1.3.

- **Chapter 3. Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management**

This chapter handles the first part of our solution design and solution validation, within the context of the SUPER project [SUP07]. Research question RQ2.1 is answered in section 3.3, as we extend an existing GORE method, called Formal Tropos [FLM<sup>+</sup>04], to handle policy annotations, and we illustrate how our proposal (Policy-extended Formal Tropos) maps to business process designs. The implementation has been done in the IBM Eclipse environment, which is elaborated in section 3.4 (appendix D provides more information about the creation of the PFT graphical editor). Then, research questions RQ3.1 and RQ3.2 are answered by section 3.5, which reports on the findings of three pilot studies. Note that this work extends existing research, so reading the following list of papers is required to understand our full research

context: [Yu97, CKM02, Fux01, FLM<sup>+</sup>04, Kag04].

- **Chapter 4. Goal-Oriented Requirements Engineering for BPMN Modelling**

This chapter introduces the second part of our solution design and solution validation, by proposing a GORE for BP method for BPMN modelling. Research question RQ2.2 is answered in section 4.3, which explains our extension of existing GORE for BP methods [BCVP06a, LYM07], together with presenting the IBM Eclipse implementation details. Although chapter 4 introduces the BSCP2BPMN model transformations, the in-depth technical details about the transformation and editor are discussed in technical appendices C and D. Then, research question RQ3.3 is answered by section 4.4, in which we use the SEJ case exemplar to demonstrate the feasibility of our approach. Finally, research question RQ3.4 is answered in section 4.5, by presenting the findings from two case studies. Note that this work extends existing research, so reading the following list of papers is required to understand our full research context: [LYM07, BCVP06a, BCVP06, BCVP06b, Yu97, Jac00, Sil09].

- **Chapter 5. Conclusion**

This chapter summarizes the answers to the research questions, and discusses the impact of our findings for scholars and for practitioners.

## 1.5 Publications

Parts of this dissertation have already been presented at international conferences and workshops, or have been published in international journals.

### 1.5.1 Publications in peer-reviewed international journals

- K. Decreus, Poels, G., Kharbili, M.E., Pulvermueller, E.: Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management. *International Journal of Intelligent Systems* 25(8):784-812, 2010 [Chapter 3]

### 1.5.2 Publications in peer-reviewed international conference proceedings

- K. Decreus, M. Snoeck, G. Poels (2009). Practical Challenges for Methods Transforming i\* Goal Models into Business Process Models. (Proceedings of 17th IEEE International Requirements Engineering Conference 2009 - Atlanta, Georgia, USA) [Chapter 2]
- K. Decreus, M. Snoeck, G. Poels (2009). Practical Challenges for Methods Transforming i\* Goal Models into Business Process Models. (4th SIKS/BENAIIS Conference on Enterprise Information Systems - Nijmegen, The Netherlands) [Chapter 2]

- K. Decreus, G. Poels (2009). Mapping Semantically Enriched Formal Tropos to Business Process Models. (Proceedings of 24th Annual ACM Symposium on Applied Computing - Requirements Engineering Track - Honolulu, Hawaii, USA) [Chapter 3]

### **1.5.3 Publications in peer-reviewed international scientific workshops**

- K. Decreus, G. Poels (2010). A Goal-Oriented Requirements Engineering Method for Business Processes (Springer LNBI Post-Proceedings of the CaiSE Forum, Tunisia, 2010) [Chapter 4]
- K. Decreus, G. Poels (2010). A Goal-Oriented Requirements Engineering Method for Business Processes (Proceedings of the CaiSE Forum, Tunisia, 2010) [Chapter 4]
- K. Decreus, M. El Kharbili, G. Poels, E. Pulvermueller (2009). Bridging Requirements Engineering and Business Process Management. (Proceedings of 1st Workshop for Requirements Engineering and Business Process Management - Conference on Software Engineering 2009 - Kaiserslautern, Germany) [Chapter 2, 3 and 4]
- K. Decreus, G. Poels (2008). Putting Business into Business Process Models. (Proceedings of the 1st IEEE International Workshop on Semantics for Business Process Management - 32nd Annual IEEE International Computer Software and Applications Conference 2008 - Turku, Finland) [Chapter 4]
- K. Decreus, F. Gailly, G. Poels (2008). A Toolkit For Business Process Owners to Capture Early System Requirements. (Proceedings of 3rd international Workshop on Semantic Business Process Management - 5th European Semantic Web Conference 2008 - Tenerife, Spain) [Chapter 3]

### **1.5.4 Publications in PhD symposia related to international reviewed scientific conferences**

- K. Decreus (2009). Bridging Requirements Engineering and Business Process Management: Developing End User Methods for Goal-Oriented Business Process Modelling (RE2009 PhD Workshop at 17th IEEE Conference on Requirements Engineering - Atlanta, Georgia, USA) [Chapter 1]
- K. Decreus (2008). Requirements-driven development in SUPER (ER2008 PhD Workshop at 27th International Conference on Conceptual Modeling - Barcelona, Catalonia, Spain) [Chapter 1]

### **1.5.5 Presentations at international scientific workshops**

- K. Decreus (2008). Empowering business process owners using SUPER (SUPER International Workshop, Eindhoven, The Netherlands - Published online at <http://www.ip-super.org/content/view/169/63/>) [Chapter 3]

- K. Decreus (2010). Goal-Oriented Requirements Engineering for Business Processes (LASSY Seminar, Luxemburg) [Chapter 4]



# 2

## Investigating Goal-Oriented Requirements Engineering for Business Processes

### 2.1 Introduction

Since organizations moved into business process reengineering [HC94] in the nineties and business process management [SF04] in the 2000s, business process models are considered as valuable assets. As business processes are designed to execute the business activities taking place in an organization, managing business processes is crucial as they are the bridge between the strategy formulation and execution layers in an organization.

Traditionally, the strategy execution layer in an organization contains software applications that support the execution of specific activities, e.g. to write a letter, to print an invoice, to complete a transaction. Currently, there is a shift from command-based applications to workflow-based applications that support the execution of whole business processes [RSB05]. Such applications could benefit from a Business Process Management System (BPMS), which is a modelling, integrational and executional environment for the design of business processes [SHK<sup>+</sup>07]. However, as noted by [RSB05, RB10], Requirements Engineering (RE) techniques and tools for better aligning business processes with their BPMS support are currently lacking.

The importance of this problem is illustrated by Gartner Research [WG06], which estimates that by 2015 30% of business applications will be developed by means of BPMS technology. As traditional software packages are expected to play a less important role, we foresee a growing need for RE techniques that are adapted to BPMS packages. The overall objective of our research is to bridge the gap between RE and Business Process Management (BPM) by focusing on the common notion of *goal* in both areas of research. As business processes are meant to fulfil through their execution certain organizational goals, it is worth

investigating whether goal models that document and analyse organizational strategies, could play an active role in the design of business processes. This idea of goal-oriented RE for business processes (GORE for BP) will be further elaborated in the remainder of this section, after which the research questions addressed in this chapter are formulated.

### 2.1.1 Challenges

In previous research, we conducted a laboratory demonstration during which a realistic example in an artificial environment was used to evaluate six GORE for BP methods [DSP09]. The analysis results indicate that methods to transform  $i^*$  [Yu97] goal models into business process models lack clear descriptions of the responsible role (e.g., in case of wrong method execution, who should take responsibility?), have insufficient concept mappings (e.g., how should an actor in a  $i^*$  goal model be represented in a process model?), often lack formality in the transformation algorithm (e.g., if the transformation is executed by different people, will they interpret the informally described transformations in the same way?), have no full support to model organisational structure (e.g., how should a complex hierarchical network of employees be modelled?), and finally have little attention for inter-model consistency checks (e.g., is the information in the process model consistent with the goal model?). However, as only one kind of goal modelling language was considered ( $i^*$ ) and only top-down methods were included, it is not possible to generalise these results to other GORE for BP methods.

We believe that there is a need for a more general investigation of GORE for BP methods, including all kinds of goal modelling languages and business process modelling languages, and including both top-down and bottom-up methods. To this end, it is challenging to compare GORE for BP methods, because we lack understanding of the unifying framework of such a method (e.g., what are the typical activities of such a method?) and we have no overall view on the currently existing GORE for BP methods.

### 2.1.2 Research Questions and Contributions

Our research design is structured via the conceptual framework of Wieringa and Heerkens [WH06]. The basic distinction in this framework is one between knowledge problems and world problems. A *knowledge problem* consists of a lack of knowledge about the world. To solve a knowledge problem, we need to change the state of our knowledge, and by doing this, we try not to change the world. *World problems* consist of a difference between the way the world is and the way we think it should be. We solve a world problem by trying to change the state of the world: we change an organisation, we build a device, we implement a program, etc. There is a mutually recursive relation between knowledge problems and world problems, as there might be a world problem in the background of a knowledge problem and vice versa.

This chapter addresses the challenges of Section 2.1.1, by considering the lack of understanding of similarities and differences of GORE for BP methods as the overall knowledge problem. In order to solve this overall knowledge problem, we will build a unifying framework for

GORE for BP methods (sub-problem of the overall knowledge problem, i.e. a new world problem) and we need to have a list of currently existing GORE for BP methods (sub-problem of the overall knowledge problem, i.e. a new knowledge problem). In attempting to deal with this matter, this chapter addresses the following research questions (cfr. Chapter 1 - Figure 1.7):

- RQ1.1. What is the unifying framework of a GORE for BP method?
- RQ1.2. Which are the currently available GORE for BP methods that are sufficiently *complete* in terms of documentation, to allow for comparison with other methods and investigation of strengths and weaknesses?
- RQ1.3. What are the similarities and differences between these *complete* GORE for BP methods in relationship to the unifying framework addressed by RQ1.1?

In answering these questions, we make a number of research contributions. Our first contribution tackles the world problem of RQ1.1 and extends Kavakli's [KL99a, Kav02] unified framework of GORE methods for deriving a unified framework of GORE for BP methods. More specifically, we propose a GORE for BP method metamodel that allows us to model and compare individual GORE for BP methods. To this matter, we could not directly reuse the unifying framework of Kavakli [KL99b, Kav02], as GORE methods and GORE for BP methods are created to serve a different purpose. Whereas GORE methods can be applied in different contexts, GORE for BP methods are primarily designed for business process modelling applications (Figure 2.1). Essentially, GORE methods are proposed to capture the *change from current to future situations* whereas GORE for BP methods try to *model the complexities of the current reality* in an overall business process architecture. For instance, a GORE method typically investigates what organisation X is currently trying to achieve, what the requirements for change are and how they can be realized, what organisation X wishes to achieve in the future, and how well an organisational model satisfies the criteria of organisation X's stakeholders. In contrast, a GORE for BP method typically captures high-level strategic requirements and low-level details about business processes, and links these strategic requirements and business process details in a top-down or bottom-up fashion.

Once we have proposed a unifying framework for GORE for BP methods (dealing with the world problem of RQ1.1), we can identify the currently available GORE for BP methods (tackling the knowledge problem of RQ1.2). As the research in GORE for BP methods is fragmented, it is not a trivial task to objectively select and analyse proposed methods. Hence, our second contribution is a systematic literature review in which GORE for BP methods are identified, considering all possible types of goal models and business process models. Typically, a GORE for BP method will start by capturing high-level strategic requirements and transform them into low-level business process details (top-down), but we also included methods that start from business process details and extract the higher-level goals from these business process details (bottom-up).

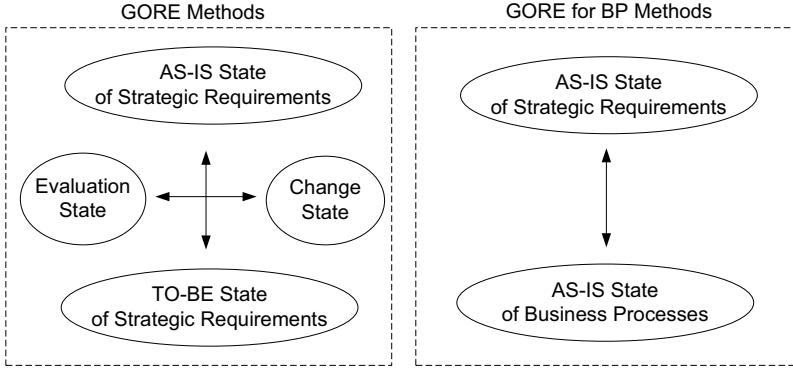


Figure 2.1: GORE Methods versus GORE for BP Methods

While performing the systematic literature review, we discovered large differences in completeness of method descriptions, ranging from vague descriptions to detailed user manuals. As we are interested in comparing these GORE for BP methods, we need to reduce the subjective interpretation of the identified methods as much as possible. As a result, we identified seven GORE for BP methods that are sufficiently complete in terms of documentation to enable an objective evaluation.

Finally, our third contribution deals with the overall knowledge problem of RQ1.3, by applying the GORE for BP method metamodel to the seven identified GORE for BP methods that have a *complete* description. By doing so, we aim to demonstrate the appropriateness and utility of our unifying framework and to enable a discussion on the similarities and differences between these methods.

This chapter is structured as follows: Section 2.2 presents the unifying framework for GORE for BP methods (answering RQ1.1). Section 2.3 reports on the systematic literature review to identify the currently available GORE for BP methods and select those that are *complete* in terms of documentation (answering RQ1.2). Section 2.4 then investigates the similarities and differences of the identified GORE for BP methods (answering RQ1.3). Finally, Section 2.5 concludes the chapter, and introduces future work.

## 2.2 A Unifying Framework for GORE for BP Methods

In search for a unifying framework of GORE for BP methods, one could start from conceptual frameworks that have been developed for understanding the RE activities. Kavakli [KL99b] considers RE as a knowledge-modelling process, in which knowledge models form both the result from RE tasks and the basis for reasoning during the RE process. As our work extends the work of Kavakli [KL99b, Kav02], we will also treat RE as a knowledge-modelling process. More specifically, as this chapter deals with GORE for BP, certain knowledge-modelling activities (and corresponding knowledge-modelling states) will relate to RE and others will relate to BP modelling.

To identify specific GORE for BP knowledge-modelling activities, we looked at the lifecycles of both RE and BPM. To start with, a typical RE lifecycle (e.g. [vL00]) includes domain analysis, elicitation, negotiation and agreement, specification, specification analysis, documentation and evolution. A typical BPM lifecycle (e.g. [vdA04]) includes process design, system configuration, process enactment, and diagnosis. A combination of both lifecycles is proposed in the Requirements Engineering for Business Process Management (REBPM) framework [DKPP09]. The REBPM framework does not prescribe which detailed activities should occur during the RE or BPM lifecycles, but offers a high-level view on the main types of activities in RE and BPM and acknowledges the need for a *transformation* activity between the RE and BPM lifecycles. This *transformation* activity plays an important role, because we are less interested in how a specific RE or BPM activity should be executed, and more interested in how the results of a given RE (or vice-versa BPM) activity can be used as a valuable input for a BPM (or vice-versa RE) activity. More specifically, the *transformation* activity occurs when any state of the RE lifecycle is transformed into any state of the BPM lifecycle, or when any state of the BPM lifecycle is transformed into any state of the RE lifecycle. Overall, the REBPM framework (Figure 2.2) recognizes seven knowledge-modelling states:

- *Elicited Requirements state.* As a result of using RE elicitation techniques, the requirements and domain assumptions are identified. For instance, interviews conducted with top management of car manufacturer X suggested that the cash cycle should be shortened by reducing the net working capital of the company.
- *Specified Requirements state.* Using conceptual modelling languages, the requirements and domain assumptions are specified in a precise way. For instance, an i\* goal model was made to specify the alternative ways to reduce net working capital in company X.
- *Validated Requirements state.* The requirement specifications are checked for deficiencies (such as incompleteness or inconsistency) and for feasibility (in terms of resources or development costs). For instance, alternative ways to reduce net working capital could conflict with each other, which could cause problems in company X.
- *Process Designed state.* The business processes are designed, which could be done in an ad-hoc fashion (without explicitly known requirements) or based on previously identified requirements. For instance, a modeller could use the Intalio Designer tool (which is part of Intalio BPMS) to design the business process *Manage Cash Cycle* based on his ad-hoc knowledge of the process or based on the information coming from an i\* goal model.
- *System Configured state.* Based on the process design, the system configuration is realized. For instance, a modeller takes an existing model of business process *Manage Cash Cycle* and uses Intalio Designer to add web services to implement automated activities, or creates workflow forms to support semi-automated activities.

- *Process Enacted state.* The configured processes are deployed to the run-time component of a BPMS environment to get the process enacted by the end user. For instance, the business process model of *Manage Cash Cycle* (made with Intalio Designer) can be deployed into the application server of Intalio BPMS, with which end users can enact the *Manage Cash Cycle* business process model.
- *Process Diagnosed state.* The currently enacted processes are analysed to improve performance. For instance, the enactment of process *Manage Cash Cycle* could be blocked by a broken link to a web service, which will be visible in the diagnosis reports of the Intalio application server.

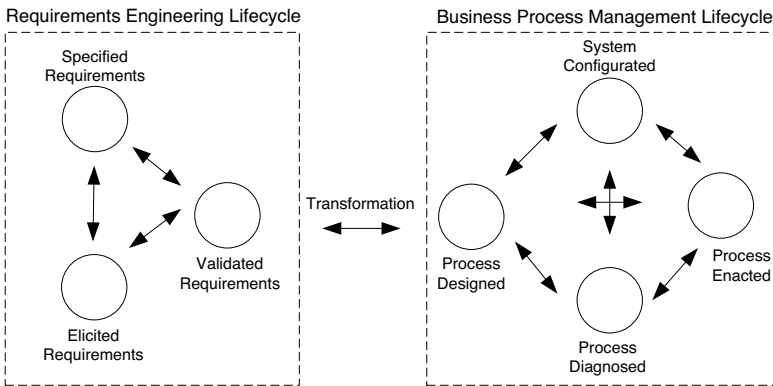


Figure 2.2: The seven GORE for BP knowledge-modelling states (adapted from Decreus et al. [DKPP09])

These seven states and their transitions constitute the unifying framework of GORE for BP methods. Executing such a method can be seen as a systematic progression through the seven knowledge-modelling states, and specific paths of transitions between these knowledge-modelling states define ways-of-working [RSM95]. The REBPM framework recommends modellers to follow the natural flow of transitions between knowledge-modelling states that are inherited from the contributing RE and BPM lifecycle models indicated by the arrows in Figure 2.2, but recognizes the iterative character of modelling and allows the skipping of certain states. For instance, the requirements could be elicited (Elicited Requirements state), specified (Specified Requirements state), and validated (Validated Requirements state), which could lead to a next round of elicitation (Elicited Requirements state) and specification (Specified Requirements state), which is next transformed into a process design (Process Designed state). Or an existing process design (Process Designed state), that has directly been configured (System Configured state) and enacted (Process Enacted state) in a BPMS package, could be subject of a diagnosis (Process Diagnosed state) that inspires modellers specifying the high-level requirements for the business process (Specified Requirements state).

## 2.2.1 A Metamodelling Technique for Describing GORE for BP Methods

When we want to build a conceptual framework for methods, we require proper guidelines and abstraction mechanisms to do so. Within the field of information system development, a basic abstraction mechanism adopted to understand and engineer artefacts is that of metamodelling. Often, metamodelling is used to describe the syntax of conceptual modelling languages [Gui07], such as the metamodel of Ayala et al. [ACC<sup>+</sup>05] that describes i\* concepts (e.g. actor and goal) and relationships between these concepts (e.g. an actor has one or more goals). In contrast, our work employs metamodelling as a way to understand *methods* instead of *languages*.

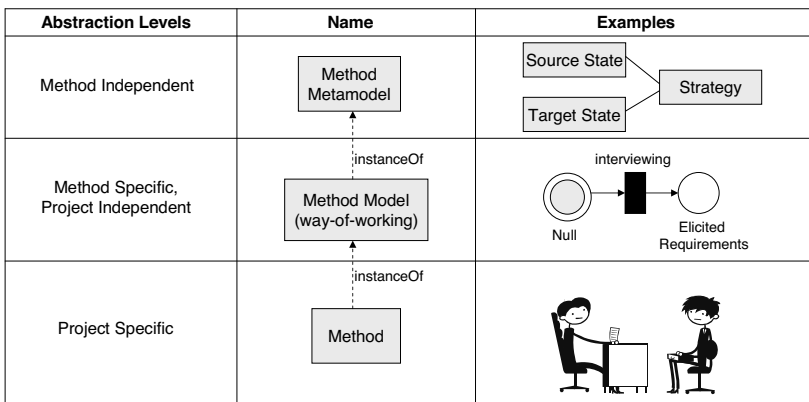


Figure 2.3: Overview of a Metamodelling Technique (Adapted from [RSM95, RNG99, RPB99])

This chapter employs the principles of method metamodelling as introduced by Rolland et al. [RSM95, RNG99, RPB99]. To begin with, a *method metamodel* provides a set of generic concepts to represent any method model. For instance, a simple method metamodel could define the concepts ‘Source State’, ‘Target State’ and ‘Strategy’ and an association relationship between them, such that ‘Strategy’ is associated both to a ‘Source State’ and to a ‘Target State’ (Figure 2.3 - Method Metamodel). Next, a *method model* is used to prescribe ‘how things must/should/could be done’ according to a particular method and is often referred to as a *way-of-working*. For instance, a specific method could have a way-of-working to elicit requirements via interviewing techniques, which could be represented by moving from ‘Source State’ *Null* to ‘Target State’ *Requirements Elicitation* via ‘Strategy’ *Interviewing* (Figure 2.3 - Method Model). As a way-of-working is project independent, the interviewing techniques could be used in different kinds of projects, such as software engineering projects or business process modelling projects. Finally, a *method* is the actual execution of the activities that were recommended by a way-of-working, done in the context of a specific project. For instance, to elicit process requirements during a business process modelling project at Company X, people could use the interviewing techniques as prescribed by the way-of-working in Figure 2.3.

## 2.2.2 The GORE for BP Method Metamodel

In order to use the metamodelling technique presented in the previous section, we need a GORE for BP method metamodel as starting point. As GORE for BP methods are closely related to GORE methods, our metamodel (Figure 2.4) keeps the generic structure of the GORE method metamodel proposed by Kavakli [KL99b, Kav02], but adjusts the knowledge-modelling states to fit the REBPM framework [DKPP09]. In that way, a GORE for BP method describes a particular route through the framework that can be followed in order to reach the seven REBPM knowledge-modelling states. Each specific method is characterized by such a route.

Each step in a route prescribed by a method constitutes a method fragment and expresses the intention to reach state  $S_j$  starting from a source state  $S_i$  using a method-specific strategy  $Str_{ij}$ . A GORE for BP method then consists of a number of method fragments, each of which is a triplet  $\langle S_i, S_j, Str_{ij} \rangle$  whereby  $S_i$  and  $S_j$  are knowledge-modelling states and  $Str_{ij}$  is a method-specific strategy for reaching state  $S_j$  from state  $S_i$ . Note that the metamodelling technique of Roland et al. [RSM95, RNG99, RPB99] uses *strategy* in a different way as defined in OMG's Business Motivation Model [OMG09b].

When a GORE for BP method is used in specific projects, there is a certain state of knowledge at the beginning and at the end of the method. In order to be complete, we introduce two knowledge-modelling states to capture the knowledge extremes, i.e. the lack of knowledge about a GORE for BP method (Null) and the recognition of having sufficient knowledge about a GORE for BP method, such that the project owners can sign off the project (Signed Off). So at the beginning of a project, no knowledge could be present (Null) or a certain REBPM knowledge-modelling state could already be reached (Elicited Requirements, Specified Requirements, Validated Requirements, Process Designed, System Configured, Process Enacted, Process Diagnosed). Our method metamodel specifies this constraint by defining the *Started* relationship between Method and Knowledge-Modelling state. At the end of a project, the last knowledge-modelling state (Elicited Requirements, Specified Requirements, Validated Requirements, Process Designed, System Configured, Process Enacted, Process Diagnosed) should always be succeeded by signing off the project (Signed Off). Hence, a composition of method fragments is only considered as a method when it is possible to sign off the method. Our metamodel specifies this constraints by means of the *Signed Off* relationship between Method and the Signed Off knowledge-modelling state.

Figure 2.5 shows an example of a route through the REBPM framework, by combining different method fragments of the Soffer and Roland method [SR05]. To keep the representation of methods simple, we will represent the route as a linear path through the different states of the REBPM framework. In reality though, most methods assume an iterative process allowing to go back and forth between the different states. Furthermore, a distinct notation is used for the knowledge-modelling state that is indicated by the Started relationship (beginning of the method) and for the Signed Off knowledge state that is constrained by the Signed Off relationship (ending of the method).



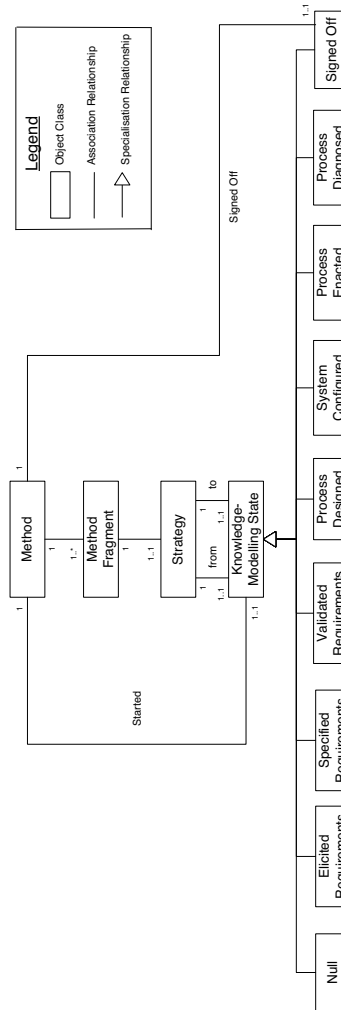


Figure 2.4: The GORE for BP Method Metamodel (Based on Kavakli [KL99b, Kav02])

The Soffer and Rolland way-of-working (Figure 2.5) recommends two alternative strategies for requirements elicitation, i.e. linguistic strategy (defining a goal as a verb with associated semantic functions) and template-driven strategy (asking users to fill in a goal template). Next, the MAP modelling formalism is used to specify the requirements, which are transformed into a process design (Generic Process Model) by following a specific procedure: (1) define the intentions and the related conditions, and (2) define sections as the law and describe the related conditions, and (3) refine the sections where needed. Finally, the process design is formally analysed to discover anomalies and deficiencies, in order to validate the initial set of requirements (specified by the MAP model), after which the method can be signed off. Figure 2.5 further shows that a strategy may define a number of subtasks and intermediate method-specific states that have to be reached in the context of the

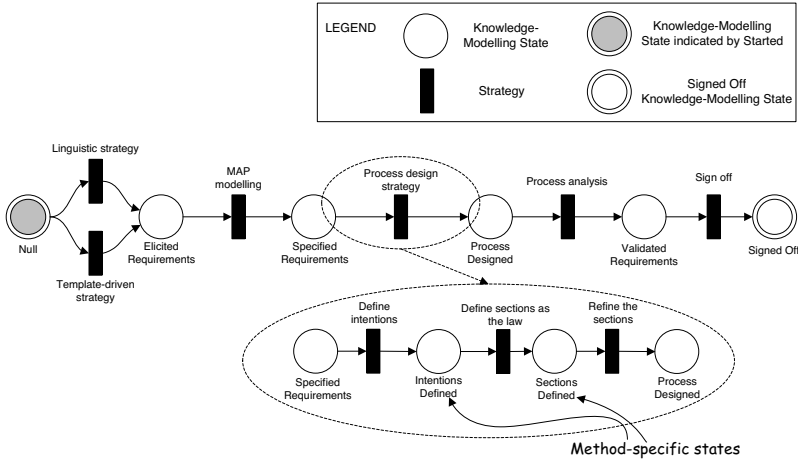


Figure 2.5: The Soffer and Rolland way-of-working

strategy. For instance, the process design strategy is decomposed into three subtasks (Define intentions, Define sections as the law, and Refine the sections) which have two intermediate method-specific states (Intentions defined, and Sections defined).

### 2.3 Identifying GORE for BP Methods

In order to apply our unifying framework to existing GORE for BP methods, we first need to identify GORE for BP methods. To this end, we conducted a systematic literature review to search for existing GORE for BP methods and then select those methods that have a complete description in terms of documentation. A systematic literature review is a specific kind of knowledge problem, and can be defined as a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [Kit04]. In conducting the review we followed the guidelines of Kitchenham et al. [Kit04, KDJ04, BKB<sup>+</sup>07, KPBB<sup>+</sup>09], which have their origin in medical research, but are adapted to the needs of software engineering research. To start with, Kitchenham et al. [Kit04, KDJ04, BKB<sup>+</sup>07, KPBB<sup>+</sup>09] recommend to specify the research questions and to develop the review protocol. The review protocol used for our review was developed based on research question RQ1.2 “Which are the currently available GORE for BP methods that are sufficiently complete in terms of documentation?”, where the requirement ‘complete in terms of documentation’ is motivated by RQ1.3. This review protocol was validated before the actual execution of the protocol by means of a pilot run (Section 2.3.1). Next, the review was executed by following the defined protocol, which resulted in a list of papers in which the original authors describe their method. Often, authors publish different papers about their method, each containing a different part of the overall method. In order to cover missing information about methods, we added previous work of the authors, or external work that the authors build upon, to the initial list of papers (Section 2.3.2). Finally, the quality of each method was as-

essed in terms of the completeness of the description of the GORE for BP method addressed by the study (Section 2.3.3).

### 2.3.1 Develop and Validate Review Protocol

A review protocol specifies the techniques that will be used to undertake a systematic review. A pre-defined protocol is necessary to reduce the possibility of researcher bias. The components of a protocol include the search strategy (Section 2.3.1.1), the selection criteria and procedures (Section 2.3.1.2), and the quality assessment procedures (Section 2.3.1.3). As Kitchenham et al. [Kit04, KDJ04, BKB<sup>+</sup>07, KPBB<sup>+</sup>09] recommend to pilot the review protocol on a subset of discovered papers, three observers executed a pilot run of the protocol. Afterwards, the findings were discussed and the review protocol was corrected to reach consensus among the observers.

#### 2.3.1.1 Search Strategy

The search strategy includes the definition of search terms and the resources to be searched. These resources may include databases, specific journals, and conference proceedings. We decomposed research question RQ1.2 in three main keywords: goal (from GORE), business process (i.e. BP), and method. We also identified synonyms for these keywords (Table 2.1). The search strings were constructed using the Boolean ‘AND’ to join the main terms, and ‘OR’ to include synonyms.

(1) Goal OR Objective OR Intention OR Purpose AND (2) Process OR Workflow AND (3) Method OR Technique OR Approach OR Grammar OR Metamodel
---

Table 2.1: Initial Search Terms Used

We limited the search scope to the main e-libraries (searching the fields ‘title’, ‘abstract’ and ‘keywords’) in the domains of RE and BPM, including Springer (including the Requirements Engineering Journal and the Lecture Notes in Computer Science series which publishes proceedings of major relevant conferences and workshops like CAiSE, BP-MDS, EMMSAD, ER and BPM), ACM (including ACM SAC proceedings), Elsevier (including journals relevant to the target domains such as Information Systems, Data and Knowledge Engineering, and Information and Software Technology), IEEE (including the proceedings of the RE Conference and the COMPSAC and EDOC conferences) and the Business Process Management Journal (Emerald).

The main outcome of the pilot run was that the proposed search strategy is too specific (few search results), and that our search left methods uncovered that were intuitively good candidates. For instance, authors used unforeseen synonyms (e.g. *framework* [KP02] instead of method or technique) or general keywords, titles and abstracts without our specific set of search terms (e.g. ‘Model Driven Architectures for Enterprise Information Systems’ [BN04]). In order to fix this problem, we decided to drop the third series of terms (i.e. method, technique, etc.)

in the conjunction. As a result of this change, we expect a larger number of results per e-catalogue, but also more irrelevant results, which have to be manually eliminated, based on the selection criteria.

### **2.3.1.2 Selection Criteria and Procedures**

Study selection criteria determine criteria for including a study in, or excluding it from, the systematic review. The protocol should describe how the criteria will be applied e.g. how many assessors will evaluate each prospective paper, and how disagreements among assessors will be resolved. In order to reduce possible subjective searches, all three observers executed the search strategy individually, and reached a consensus to consolidate these three searches. Furthermore, this work was done in August 2009, so work published from 1980 up until the 1st of August 2009 was included.

### **2.3.1.3 Quality Assessment Procedures**

In this chapter, we are interested in comparing GORE for BP methods, so we need to reduce the subjective interpretation of the identified methods as much as possible. As a result, we define study quality in terms of completeness of documentation. We distinguished three quality categories (low, medium and high) and provided a set of necessary conditions per category in Table 2.2. Methods that are considered 'low' in terms of completeness, have at least to explicitly identify the concepts 'goal' and 'business process', for instance, by means of a metamodel or grammar that contains these two concepts. Furthermore, a relationship should be described between the two concepts, for instance, business process X 'realizes' goal Y. Methods that are considered 'medium' in terms of completeness, satisfy the conditions for 'low' completeness and have an implicit description of how to link goals and business processes, but lack explicitly distinguished method steps. So relative to 'low' methods, 'medium' methods not only mention a relationship between goals and business processes, but also provide hints to transform one concept into another (goals into business processes, or business processes into goals). These 'medium' methods employ at least an example to illustrate the general ideas behind the method. However, the information provided is not sufficient to allow other researchers to duplicate the original research effort. In contrast, methods that are considered 'high' in terms of completeness satisfy the conditions for 'medium' completeness, and offer explicitly described distinguishable steps to execute the method. Also, a description is given of the type of artefact produced by executing each method step. So relative to 'medium' methods, 'high' methods do not only provide hints to transform one concept into another, but offer repeatable method descriptions that describe which transformation steps are done and what the expected deliverables are. Ideally, a user manual should be provided by the authors to guide end users to execute the method.

## **2.3.2 Selection of GORE for BP Papers**

The application of the search strategy resulted in three lists of papers of which title, abstract and keywords were reviewed by each of the three

Category	Necessary Conditions
Low	- Separate concepts ‘goal’ and ‘business process’ - At least one relationship between them is described
Medium	- Satisfy conditions of ‘low’ - An implicit description of how to link goals and business processes (but no explicitly distinguishable method steps) - An example that illustrates the method
High	- Satisfy conditions of ‘medium’ - An explicitly described method that contains distinguishable steps to execute the transformation - Description of the type of artefact produced by each method step

Table 2.2: *Quality Metrics in terms of Completeness of Documentation*

observers. A united list of titles was obtained by reaching a consensus on the individual searches done by the observers. Full copies of the papers were then reviewed by all three reviewers against the inclusion / exclusion criteria defined in the protocol. This resulted in an initial list of 38 GORE for BP papers, both journal and conference papers, where authors describe the development or evaluation of (parts of) GORE for BP methods, often highlighting specific research issues instead of covering the entire method. In order to cover missing information about methods, we added previous work of the authors or external work that the authors build upon to the initial list of discovered papers. We browsed the World Wide Web for previous work of the authors, and more specifically, the author’s personal publication pages and publication repositories such as DBLP Computer Science Bibliography and Scientific Commons. After grouping related papers, a total of 19 GORE for BP methods was discovered, which we will briefly discuss in alphabetical order:

- Aburub et al. [AOB07] propose a method for modelling non-functional requirements of business processes in the form of goals and to link these goals to the underlying BPs. This method builds upon the work of Cysneiros et al. [CdPLdMSN01].
- Bleistein et al. [BCVP06a,BACR04,BCV04c,BCV04a,BCV06,BCVP06b,BCV04b,BCV05,CPBV05,CP03] propose a requirements analysis framework for validating strategic alignment of organisational IT based on strategy, context and process. The authors combine Jackson problem frames, i\* goal modelling, and Role-Activity Diagrams (RAD) to represent BPs and put forward a well-described top-down method.
- de la Vara et al. [dIVS07,dIVSP08,dIVS08] developed a method for a business process-driven GORE approach that transforms models in Business Process Modelling Notation into Map and i\* goal models.
- Frankova et al. [FMS07b,FYS07,FMS07a,SHF08] introduce a method to translate Secure Tropos [GMMZ06,MMZ08] goal models into a propriety business process language, with focuses on service-level agreements.

- Grau et al. [GFM05, GFA06, GFM08, JM04] suggest an i\*-based business process reengineering method, that deals with modelling business processes and generating operational and strategic goal models from the business process models.
- Greenwood et al. [GR07, Gre08, CG08] introduce their own conceptual modelling language called Goal-Oriented Business Process Modelling Notation (GO-BPMN).
- Kavakli and Loucopoulos [KL99b, Kav04, LK95] present a goal-driven business process analysis application within a larger enterprise knowledge modelling framework, known as the Enterprise Knowledge Development approach [RNG99, BPS01].
- Kazhamiakin et al. [TPR<sup>+</sup>04, KPR04a, KPR04b, PRB04] propose mappings from Formal Tropos to Business Process Execution Language (BPEL) in the context of the Astro project.
- Koliadis et al. [KG06, KVB<sup>+</sup>06b, KVB<sup>+</sup>06a] combine goal languages i\* and KAOS [DLF93] with Business Process Modelling Notation (BPMN) for lifecycle management of business process models.
- Koubarakis and Plexousakis [KP02, KP99a, KP99b, KP00] present a formal framework for representing business process models and goal models, which was influenced by the Enterprise Knowledge Development approach [RNG99, BPS01].
- Lapouchnian et al. [LL06a, LYM07, SGM04] recognize the importance of requirements-driven design of business processes by suggesting a systematic approach to transform i\* goal models into Business Process Execution Language models.
- List and Korherr [KL06, KL07, LK05, LK06] introduce several meta-models that span goals and business processes, such as extending the UML 2 Activity Diagram with business process goals.
- Lo and Yu [LY08] go from business models to service-oriented design by means of a top-down method, using the i\* goal language and proprietary business process diagrams.
- Markovic and Kowalkiewicz [MK08] link business goals to business process models from the perspective of semantic business process modelling.
- Neiger and Churilov [NC03, NC04b, NC04a, NC06] introduce goal-oriented business process modelling with Event-driven Process Chains and Value Focused Thinking as goal language.
- Nurcan et al. [Nur04b, Nur04a, NE05, NEK<sup>+</sup>05] combine the Map goal language with business process chunks and demonstrate how Maps could be used for strategy-driven business process modelling.
- Soffer and Rolland [SR05, RPB99] combine intention-oriented process modelling (using the MAP goal language) and state-based process modelling (using Generic Process Models).

- Soffer and Wand [SW04, SW05] present a formal approach for goal-driven multi-process analysis and for analyzing the dependency of soft-goals on processes.
- Vasconcelos et al. [VCN<sup>+</sup>01] propose a metamodel for modelling goals and business processes done within the context of Organisational Engineering [TWC08].

### 2.3.3 Assess Study Quality

After obtaining the complete set of papers related to GORE for BP methods, the quality procedures were applied to assess the methods, by referring to the set of necessary conditions per category (low, medium and high). All papers related to a same method were jointly assessed as parts of methods can be described in different papers. We started by selecting the methods that obviously scored ‘low’ in terms of documentation, and then identified ‘high’ methods. The remaining methods were expected to score ‘medium’ in terms of completeness, but could well be overlooked ‘low’ or underestimated ‘high’ methods. After discussing these remaining methods and finding a consensus on their quality assessment, we obtained the resulting final list of papers (Table 2.3), clustered by method (which was found to be equivalent to research group).

Method	Completeness Category
Aburub et al. Method	High
Bleistein et al. Method	High
Frankova et al. Method	High
Grau et al. Method	High
Koliadis et al. Method	High
Lapouchnian et al. Method	High
Soffer and Rolland Method	High
de la Vara et al. Method	Medium
Kavakli and Loucopoulos Method	Medium
Kazhamiakin et al. Method	Medium
Koubarakis and Plexousakis Method	Medium
Neiger and Churilov Method	Medium
Nurcan et al. Method	Medium
Soffer and Wand Method	Medium
Vasconcelos et al. Method	Medium
Greenwood et al. Method	Low
List and Korherr Method	Low
Lo and Yu Method	Low
Markovic and Kowalkiewicz Method	Low

Table 2.3: Methods Ordered by Completeness

As Koliadis et al. proposed two variants of their method, i.e. one combining KAOS with BPMN [KG06] and another combining *i\** with BPMN [KVB<sup>+</sup>06b, KVB<sup>+</sup>06a], we selected the variant that combines *i\** with BPMN [KVB<sup>+</sup>06b, KVB<sup>+</sup>06a] due to the higher completeness of documentation. To conclude this section, the answer to research question RQ1.2 (“Which are the currently available GORE for BP methods that are *complete* in terms of documentation?”) is the list of seven methods that score ‘high’ in terms of completeness, i.e. Aburub et al.

[AOB07, CdPLdMSN01], Bleistein et al. [BCVP06a, BACR04, BCV04c, BCV04a, BCV06, BCVP06b, BCV04b, BCV05, CPBV05, CP03], Frankova et al. [FMS07b, FYS07, FMS07a, SHF08, GMMZ06, MMZ08], Grau et al. [GFM05, GFA06, GFM08, JM04], Koliadis et al. [KVB<sup>+</sup>06b, KVB<sup>+</sup>06a, FLM<sup>+</sup>04], Lapouchnian et al. [LL06a, LYM07, SGM04], and Soffer and Rolland [SR05, RPB99].

## 2.4 Similarities and Differences of GORE for BP Methods

Once we understand the unifying framework of a GORE for BP method, and once we have the list of currently existing GORE for BP methods that are sufficiently complete in terms of documentation, we can investigate the similarities and differences of these GORE for BP methods. In order to tackle this knowledge problem, we will apply the methodology proposed by Wieringa and Heerkens [WH06]. The phenomena of our knowledge problem are the GORE for BP methods that have been identified as *High* in Table 2.3. The variables we introduce to investigate these phenomena are given by the following list:

- *Completeness in terms of REBPM lifecycle*: although Section 2.3 identified GORE for BP methods that are *complete in terms of documentation*, we also want to investigate the *completeness of GORE for BP methods in terms of the REBPM lifecycle*. For instance, some GORE for BP methods might only specify requirements and transform these requirements into process designs, while other GORE for BP methods could focus on the elicitation of requirements and the enactment of the processes that correspond to the elicited requirements.
- *Kind of conceptual modelling languages*: different kinds of conceptual modelling languages are used in the investigated GORE for BP methods. For instance, a GORE for BP method might employ different goal-oriented languages (e.g., i\* or KAOS) and different business process modelling languages (e.g., BPMN or RAD).
- *Validation in practical context*: the level of quantitative or qualitative validation of the GORE for BP methods in real-world, practical contexts. For instance, a GORE for BP method could be illustrated by means of a detailed example, or might have been experimentally validated in ten different companies.
- *Applicability restrictions*: the restrictions that were put forward by the creators of the GORE for BP method to have a successful application of the method. For instance, a GORE for BP method might focus on security aspects of an organisation, or could only deal with strategic processes in organisations instead of discussing technical process implementation.
- *Practical guidelines offered to modeller*: the study of Davies et al. [DGR<sup>+</sup>06] investigated the factors that inhibited the use of modelling in organisations, and concluded that the relative advantage and usefulness from the perspective of the modeller was the



major driving factor influencing the decision to continue (discontinue) modelling. Related to this perceived usefulness, the complexity of the modelling method plays an important role, and could be tackled by providing the modeller clear and practical guidelines. For instance, the modeller could be provided by cookbook-like guidelines to create a model, or might just be provided with a software tool without any further guidelines.

- *Maturity of transformation activity*: in previous research [DSP09], we discovered that the transformation of specific GORE models into business process models is often immature and done in an ad-hoc way. As this paper broadens the scope to all kinds of GORE for BP methods, we are again interested in the maturity of the transformation activity of these GORE for BP methods. For instance, a GORE for BP method might have theoretical concept mappings without tool support, or could offer a software-supported transformation algorithm.

Possible relationships among the variables include *validation in practical context* and *practical guidelines offered to modeller* or *maturity of transformation activity*, as in-depth validation would need practical guidelines for modellers and mature transformations. However, we did experience sufficient difference between these variables during our analysis, so we decided to include them as different variables. The priority in tackling our knowledge problem is given to the similarities and differences that could impact a business user from a practical viewpoint, in order to discover possible hinder when transferring these methods to industry.

## 2.4.1 Research Design

### 2.4.1.1 Evaluation Procedure

Initially, all three observers (Ken Decreus, Geert Poels, Monique Snoeck) read the papers related to the selected GORE for BP methods to have the same context during subsequent discussions. During the first session, the three observers applied the metamodelling technique for describing GORE for BP methods as explained in Section 2.2, which resulted in seven ways-of-working related to the identified GORE for BP methods (see section 2.4.3.1 - Short Overview of Methods). These seven ways-of-working played an important role in our evaluation procedure as these models represented the common understanding of all three observers, and allowed us to visualise this common understanding in a consistent way across methods.

During a second session, all three observers used the seven ways-of-working to score the variable *completeness in terms of the REBPM lifecycle*, and to create an overview of the *kinds of conceptual modelling languages*. After discussing these two variables, the observers used the seven ways-of-working in combination with the original research papers to discuss the details of the other variables (*practical guidelines offered to modeller*, *maturity of transformation activity*, *validation in practical context*, and *applicability restrictions*). Deviating results

among observers were discussed and a consensus was found for all results. For two variables (*practical guidelines offered to modeller*, *maturity of transformation activity*), the scores were given in the range of 0% (worst case), 25% (low), 50% (medium), 75% (high), 100% (best case). The four other variables were scored by means of relevant qualitative values (e.g., variable *validation in practical context* contains values *case study*, *laboratory experiment*, *field study* etc).

#### 2.4.1.2 Analysis Method

After evaluating the GORE for BP methods, the scores displayed in Table 2.4 were obtained. For the variables that were scored by means of the value range 0%, 25%, 50%, 75%, 100%, we looked at the relative performance of each method. For the variables with qualitative values, we started to identify the common values among different methods, and then looked at the different values in relation to these common ones. Based on these insights, we formulated the main similarities and differences of the GORE for BP methods.

### 2.4.2 Alleviating Threats to Validity

An important aspect in the research cycle of a knowledge problem [WH06] is discussing the validity of measurements and analysis results. Traditionally, three kinds of validity (construct, internal and external) are discussed in RE research. As we felt that a different structure fits better into the setting of this chapter, we will discuss *completeness of samples*, *completeness of variables*, *correctness of method understanding*, *correctness of scoring*, and *external validity*.

The *completeness of samples* relates to including all important methods without missing one. As we followed the guidelines of Kitchenham et al. [Kit04, KDJ04, BKB<sup>+</sup>07, KPBB<sup>+</sup>09], we organised a pilot run to test (and correct) the initial search strategy, and we applied specific quality procedures to assess the methods, by referring to the set of necessary conditions per category (low, medium and high). As we only selected the methods that scored 'high' in terms of these quality procedures, we believe to have a representative set of samples. Note that the resulting set of samples are mostly i\*-based, which might seem biased towards the i\* community. This can be explained by the fact that the i\* notation was created to investigate actors and social relationships, which is relevant to business process modelling, probably making i\* a preferred modelling language for GORE for BP methods.

The *completeness of variables* should guarantee that all variables were included to operationalize the research question. Based on a brainstorm output, the three observers decided on an initial representative set of variables (*completeness in terms of REBPM lifecycle*, *kind of conceptual modelling languages*, *practical guidelines offered to modeller*, and *maturity of transformation activity*). This set of variables was reviewed by three external researchers, and as a result, two other variables were added to the initial set (*validation in practical context*, and *applicability restrictions*).

The *correctness of method understanding* relates to how correct the observers understood the method by reading the papers written by the

original authors. We hoped to minimise method interpretation errors to choose observers with general Information Systems knowledge complemented with specific Requirements Engineering expertise. All papers were read by all three observers and interpretation errors were discussed. Important was the choice to start with a general systematic literature review, and to select the GORE for BP methods that were complete in terms of documentation (to reduce the subjective interpretation of the identified methods as much as possible).

The *correctness of scoring* deals with how the observers applied the scoring scales and with the exact definition of the scales. The scoring of the qualitative variables did not pose particular problems, as these values were literally present in the original research papers. In contrast, the quantitative scoring scales (0%, 25%, 50%, 75%, 100%) required a special protocol. To start with the extremes, when there was no mentioning of the aspect (e.g. lack of practical guidelines offered to a modeller), a score of 0% was given, and when all method steps detailed the complete aspect (e.g. a well-documented catalogue of practical guidelines offered to a modeller), a score of 100% was given. The score of 50% was given to aspects that were correctly discussed, but lacked completeness (e.g. mentioning some practical guidelines but forgetting others). The nuances of 25% and 75% were used to indicate the position of aspects relative to the other methods (e.g. briefly talking about practical guidelines is better than not mentioning, but not sufficient to achieve a score of 50%; a method that forgets to describe a practical guideline, next to a method that is perfectly complete in that aspect, will be scored 75%).

Finally, *external validity* reports on how we can generalize the discovered similarities and differences of GORE for BP methods to the entire population of GORE for BP methods. As our research started from a unifying framework of GORE for BP methods, our findings should be *relevant* to other GORE for BP methods that align with our unifying framework. Nevertheless, it is *not possible to generalize* the concrete findings of our work, as the engineering of methods is a creative task and cannot be predicted by investigating previous GORE for BP methods.

## 2.4.3 Presentation of Results

### 2.4.3.1 Short Overview of Methods

The Aburub et al. way-of-working is shown in Figure 2.6. The route starts by eliciting non-functional requirements of business processes using observation, by means of interviews or by examining business documents. Next, the specification strategy consists of three steps: (1) defining relationships between goals and subgoals, (2) specifying actors who will achieve the elicited non-functional requirements (which are represented by goals), and (3) operationalizing the goals in a static or dynamic way. After the requirements specification knowledge-modelling state, the positive and negative interactions between goals are analysed in order to discover the most beneficial operationalized goals with the least conflicts. Finally, the route terminates with a process design strategy, during which certain goals are selected for process improvement

and the non-functional requirements graph is linked to the role activity diagram, which offers a functional view on the business process.

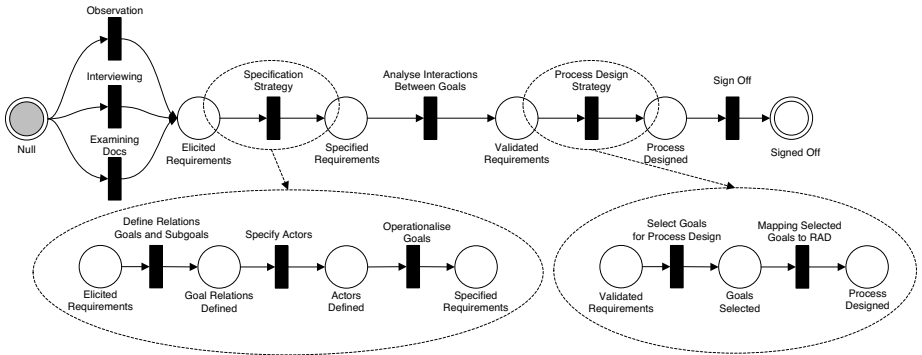


Figure 2.6: The Aburub et al. [AOB07, CdPLdMSN01] way-of-working

The Bleistein et al. way-of-working is described in Figure 2.7. To start with, the elicitation strategy consists of identifying business model participants and their relationships, and employs the VMOST (Vision, Mission, Objectives, Strategies, and Tactics) analysis, which is a technique for deconstructing business strategy into core components. Next, these strategic components are related according the rules of OMG’s Business Motivation Model (BMM) [OMG09b], and both  $i^*$  and Jackson problem frame modelling are used to visualize the related strategic components. Up to that point, only the strategy level has been modelled, so further refinement of the strategy and its context is needed to reach the requirements specification state. The validation of the strategic alignment is done by tracing the lowest-level system requirements to the highest-level strategic business objectives. Finally, the process design strategy consist of cross-referencing process models against both goal models and context diagrams as a means of better understanding the processes supporting business strategy.

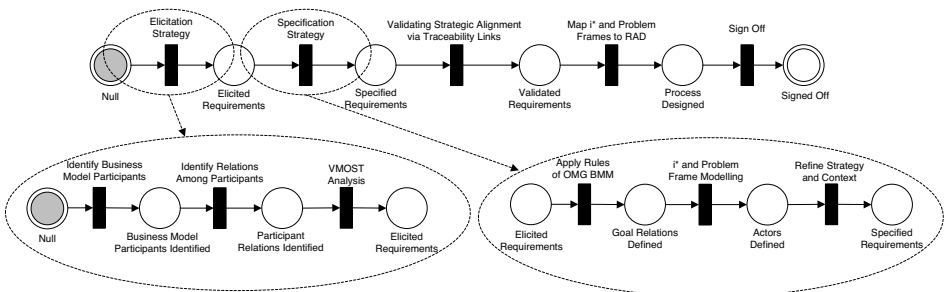


Figure 2.7: The Bleistein et al. [BCVP06a, BACR04, BCV04c, BCV04a, BCV06, BCVP06b, BCV04b, BCV05, CPBV05, CP03] way-of-working

The Frankova et al. method model is given by Figure 2.8. Based on existing elicited requirements, the Secure Tropos modelling language is used to write four parts of the requirements specification: (1) actor modelling, to specify the principal actors and their goals, (2) functional

dependency modelling, to specify actors depending on other actors for obtaining services, (3) permission delegation modelling, to specify actors delegating to other actors the permission on service usage, and (4) trust modelling, to specify actors trusting other actors for services. Next, the process design strategy refines the Secure Tropos models into intermediate structures (called business process hypergraphs and business process hierarchies) to reason about the business processes and their qualities. The designed process is then configured by means of the Secure BPEL language, which is an extension of the Web Service Business Process Execution Language (WS-BPEL) [OAS07]. Finally, the resulting Secure BPEL models are enacted by means of an WS-BPEL engine.

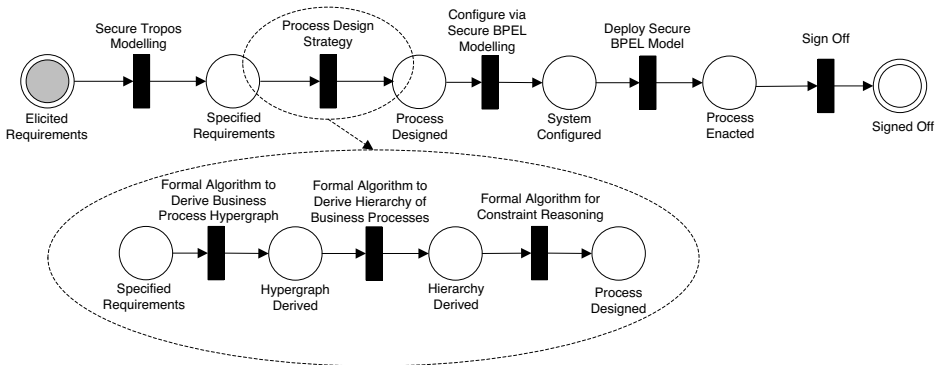


Figure 2.8: The Frankova et al. [FMS07b, FYS07, FMS07a, SHF08, GMMZ06, MMZ08] way-of-working

The Grau et al. way-of-working can be seen in Figure 2.9. The route starts by using the RESCUE method for requirements elicitation, which includes data gathering techniques such as observation of the current process, analysis of software system use reports, informal scenario walkthroughs and interviews with stakeholders. Next, the process design strategy consists of detailed interaction script modelling, which entails a simplified notation for process scenarios that includes goals, actors, preconditions, triggering events and postconditions. Based on these process scenarios, the requirements specification is obtained by following a number of steps: (1) actor specification and modelling their main goals using the  $i^*$  goal language, (2) building the operational  $i^*$  model that deals with descriptive goals, and (3) building the intentional  $i^*$  model that handles prescriptive goals. Finally, the requirements are validated by cross-checking the process scenarios and the resulting requirement specifications ( $i^*$  models).

The Koliadis et al. method has two different ways-of-working: one that transforms requirements into process designs (Figure 2.10 (a)) and another that transforms process designs into requirements (Figure 2.10 (b)). In both ways-of-working, the requirements are already elicited and readily available to be specified. When transforming requirements into process designs (Figure 2.10 (a)), the elicited requirements are specified by means of the  $i^*$  goal language. Next, the process design strategy consists of the following steps: (1) specify whether the actors are internal

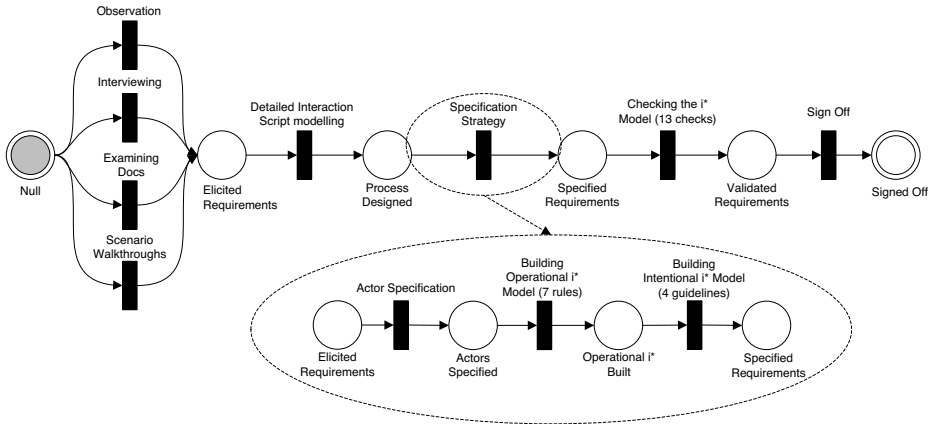


Figure 2.9: The Grau et al. [GFM05, GFA06, GFM08, JM04] way-of-working

or external to the organisation in scope, (2) map i\* concepts to BPMN concepts, (3) sequence the required tasks / sub-processes and introduce control flow links, and (4) elaborate on the resulting sub-processes. When considering the other way around, so transforming process designs into requirements specifications (Figure 2.10 (b)), a BPMN model is provided and a systematic approach is followed to convert the BPMN model into an i\* goal model: (1) map BPMN concepts to i\* concepts, (2) apply intentional reasoning by querying the intention of tasks and control flow links, and (3) specifying i\* soft goals including the dependencies between these i\* soft goals.

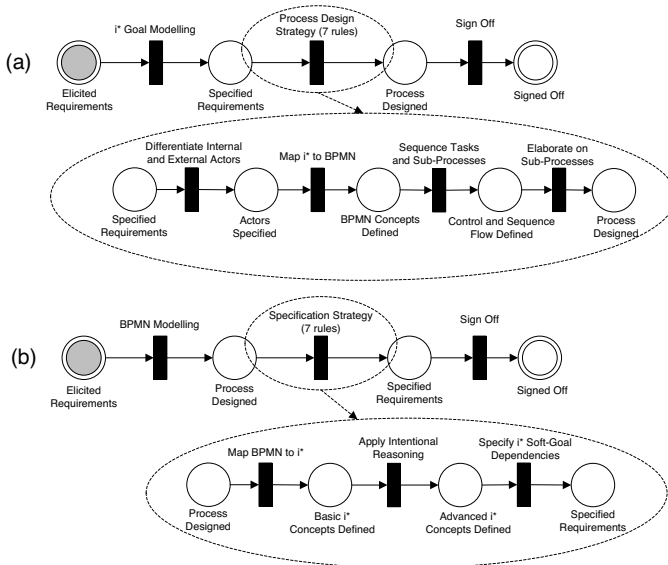


Figure 2.10: The Koliadis et al. [KVB<sup>+</sup>06b, KVB<sup>+</sup>06a, FLM<sup>+</sup>04] way-of-working

The Lapouchnian et al. method aims to configure ‘high-variability’ business processes in terms of business priorities, and the way-of-working is given by Figure 2.11. To start with, requirements are modelled via

high-variability goal modelling, and these goals are enriched by means of control flow and input/output annotations. Next, the requirement specification is validated by analysing the alternative paths in the goal model and the infeasible ones are removed. Based on the validated high-variability goal model, an initial version of the high-variability BPEL model is semi-automatically generated. After this initial process design, the configuration strategy consists of the following steps: (1) completing and deploying the high-variability BPEL process, (2) selecting the process preferences in terms of quality criteria, and (3) picking the business process configuration that matches best with the process preferences. Finally, the resulting configured BPEL process is executed on a BPEL run-time engine.

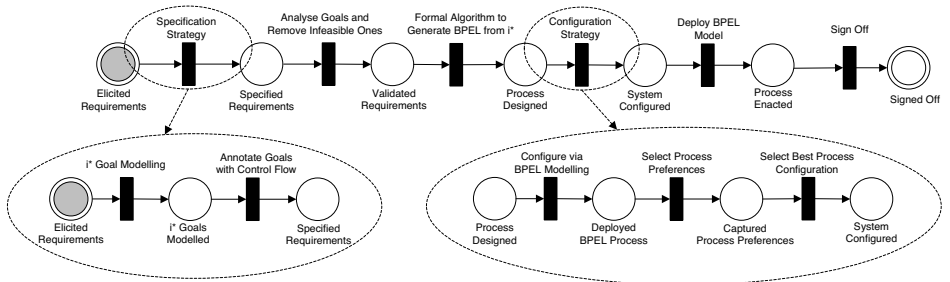


Figure 2.11: The Lapouchnian et al. [LL06a, LYM07, SGM04] way-of-working

Soffer and Rolland combine intention-oriented and state-based process modelling. The way-of-working can be described by Figure 2.5, which has been introduced in Section 2.2.2.

### 2.4.3.2 Evaluation of Methods

The results of the evaluation procedure are summarized in Table 2.4, which will be further discussed in this section.

**Completeness in terms of REBPM lifecycle.** The *Requirements Specification* and *Process Design* activities are covered by all methods. Apart from these core activities, *Requirements Validation* and *Requirements Elicitation* are typically (though not always) included in GORE for BP methods. During the *Requirements Validation* activity, either the requirements specification is validated by means a specific strategy (conflict resolution, strategic alignment via traceability links, analysing and evaluating alternative goals, checks related to a common metamodel) or the process design is validated (e.g. via a state-based formalism). During *Requirements Elicitation*, different strategies are used such as observation, interviewing, examining documents, scenario walkthroughs, identifying business models and VMOST analyses, linguistic strategy, and template-driven strategy. Next, few methods detail on how to deal with the *System Configuration* and *Process Enactment* activities. Only configuration and deployment of Secure BPEL and BPEL are discussed (by two methods out of seven), but in-depth guidelines to configure and deploy these BPEL models are missing. Finally, no methods were found that include the *Process Diagnosis* activity.

	Completeness in terms of REBPM lifecycle	Kind of conceptual modelling languages	Validation in practical context	Applicability restrictions	Practical guidelines offered to modeller	Maturity of transformation activity
Aburub et al.	<ul style="list-style-type: none"> <li>- Requirements Elicitation</li> <li>- Requirements Specification</li> <li>- Requirements Validation</li> <li>- Process Design</li> </ul>	<ul style="list-style-type: none"> <li>- RAD</li> <li>- NFR</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> </ul>	<ul style="list-style-type: none"> <li>- Only non-functional requirements</li> </ul>		
Bleistein et al.	<ul style="list-style-type: none"> <li>- Requirements Elicitation</li> <li>- Requirements Specification</li> <li>- Requirements Validation</li> <li>- Process Design</li> </ul>	<ul style="list-style-type: none"> <li>- RAD</li> <li>- i*</li> <li>- Problem Diagrams</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> <li>- Pilot Study</li> </ul>	<ul style="list-style-type: none"> <li>- No specific restrictions</li> </ul>		
Frankova et al.	<ul style="list-style-type: none"> <li>- Requirements Specification</li> <li>- Process Design</li> <li>- System Configuration</li> <li>- Process Enactment</li> </ul>	<ul style="list-style-type: none"> <li>- Secure Tropos</li> <li>- Secure BPEL</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> </ul>	<ul style="list-style-type: none"> <li>- Only for secure requirements and business processes</li> </ul>		
Grau et al.	<ul style="list-style-type: none"> <li>- Requirements Elicitation</li> <li>- Process Design</li> <li>- Requirements Specification</li> <li>- Requirements Validation</li> </ul>	<ul style="list-style-type: none"> <li>- HAM</li> <li>- DIS</li> <li>- i*</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> <li>- Case Study</li> </ul>	<ul style="list-style-type: none"> <li>- No specific restrictions</li> </ul>		
Koliadis et al.	<ul style="list-style-type: none"> <li>- Requirements Specification</li> <li>- Process Design</li> </ul>	<ul style="list-style-type: none"> <li>- i*</li> <li>- BPMN</li> <li>- Formal Tropos</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> </ul>	<ul style="list-style-type: none"> <li>- No specific restrictions</li> </ul>		
Lapouchnian et al.	<ul style="list-style-type: none"> <li>- Requirements Specification</li> <li>- Requirements Validation</li> <li>- Process Design</li> <li>- System Configuration</li> <li>- Process Enactment</li> </ul>	<ul style="list-style-type: none"> <li>- i*</li> <li>- BPEL</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> </ul>	<ul style="list-style-type: none"> <li>- Focus on high-variable business processes</li> </ul>		
Soffler and Rolland	<ul style="list-style-type: none"> <li>- Requirements Elicitation</li> <li>- Requirements Specification</li> <li>- Process Design</li> <li>- Requirements Validation</li> </ul>	<ul style="list-style-type: none"> <li>- MAP</li> <li>- GPM</li> </ul>	<ul style="list-style-type: none"> <li>- Illustration</li> </ul>	<ul style="list-style-type: none"> <li>- No specific restrictions</li> </ul>		

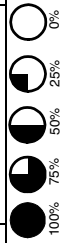


Table 2.4: Measuring GORE for BP Similarities and Differences



**Kind of conceptual modelling languages.** In total, seven different kinds of goal modelling languages (NFR, Secure Tropos, Formal Tropos, MAP and three i\*-variants), and seven different process modelling languages are employed (Role Activity Diagrams - RAD, Secure BPEL, Human Activity Modelling - HAM, Detailed Interaction Scripts - DIS, BPMN, BPEL, Generic Process Model - GPM). More specific, During the *Requirements Specification* activity, the requirements specifications are created based on different kinds of goal modelling (NFRs, i\*, Problem Diagrams, Secure Tropos, and MAPs) or are obtained via mapping rules from process designs (Detailed Interaction Scripts to i\*, BPMN to i\*). Next, during the Process Design activity, the process designs can be directly constructed from elicited requirements (Detailed Interaction Script or BPMN), generated from non-validated requirement specifications (Secure Tropos to Secure BPEL, i\* to BPMN, MAPs to GPM), or generated from validated requirement specifications (NFR to RAD, i\* and problem frames to RAD, i\* to BPEL).

**Validation in practical context.** All seven GORE for BP methods used a lightweight validation technique, i.e. illustration of their method by means of an example. Only two out of seven methods complemented these illustrations with additional validation techniques. Firstly, Grau et al. experimented with their method in different ways, and leveraged the case study insights [MJM<sup>+</sup>04] of the RESCUE method (that Grau et al. extend). Secondly, Bleistein et al. report on a pilot study that has been done at a major Australian financial institution [CBRT06].

**Applicability restrictions.** Most GORE for BP methods were created with a generic business setting in mind, and focus on the strategic requirements and operational business processes of an organisation. Nevertheless, three methods have specific restrictions for their application. Firstly, Aburub et al. only deal with non-functional requirements, and exclude functional requirements of business processes. Secondly, Frankova et al. restrict the applicability of their method to secure requirements and secure business processes. Thirdly, Lapouchnian et al. require high-variability in the business processes, with many decision points on different levels in the organisation.

**Practical guidelines offered to modeller.** In terms of supporting the modeller with practical guidelines, the method by Grau et al. outperforms all others (score = 100%). More specifically, Grau et al. offer the modeller seven rules to build the operational i\* model, four guidelines to build the intentional i\* model, and thirteen checks to verify the resulting i\* models. The method of Bleistein et al. also offers practical guidelines to modellers (score = 75%), but these guidelines relate more to the elicitation of the strategic information (by means of the VMOST guidelines [Son99] and the guidelines of Weill and Vitale [WV01]), and less to the practical modelling of GORE for BP models. Next, Koliadis et al. and Lapouchnian et al. discuss modelling guidelines, but do not provide exact details on how to use these modelling guidelines in a practical context (score = 50%). Finally, the remaining methods (Aburub et al., Frankova et al., Soffer and Rolland) focus on presenting their method by means of an example, but have less attention to the guidelines for modellers to employ their method in a practical context (score = 25%).

**Maturity of transformation activity.** In GORE for BP methods, the transformation activity occurs when (elicited, specified or validated) requirements are transformed into (designed, configured, enacted or diagnosed) processes or vice versa (see Figure 2.2). The work of Grau et al. and Koliadis et al. have a mature transformation activity, but lack full details on how the transformation should be replicated (score = 75%). Four other methods (Bleistein et al., Lapouchnian et al., Soffer and Rolland, Frankova et al.) provide an illustration of how the transformation should look like, but their work lacks detailed concept mappings between the possible language concepts (score = 50%). Finally, Aburub et al. briefly touches the idea of transforming goals into business process, but does not illustrate these transformation steps (score = 25%).

#### 2.4.4 Answering Research Question RQ1.3

Taking into account the results of Table 2.4, we are able to answer RQ1.3 (What are the similarities and differences between the *complete* GORE for BP methods in relationship to their unifying framework?). It seems that GORE for BP methods cover most of the typical RE life-cycle but are quite incomplete with respect to their coverage of the typical BPM lifecycle. So, GORE for BP methods generally focus on how to *specify* requirements and *design* processes, and how these requirements specifications and process designs can be *mapped* onto each other. Less attention is given to *elicit* or *validate* such requirements/designs, and little focus is given to the implementation of process designs by means of a BPMS system. Furthermore, we observed that no ways-of-working feed back into the RE states Elicited Requirements, Specified Requirements, Validated Requirements once the System Configured state is reached. As a result, the connection from process design to high-level requirements is lost, which might obstruct a correct diagnosis of an enacted process. For instance, the business process ‘Manage Account Receivables and Account Payables’ could put pressure on the payment terms of the suppliers, in order to achieve the goal ‘Shorten Cash Cycle’. Once parts of the business process ‘Manage Account Receivables and Account Payables’ are implemented by means of an BPMS system, the diagnosis of the enacted processes might benefit from understanding its strategic importance (which is to support the business goal ‘Shorten Cash Cycle’). Current GORE for BP methods fall short of providing backward traceability from process execution results to the goal-oriented business process requirements.

When we compare the GORE for BP methods in terms of the other variables, we discover little common ground in the different conceptual modelling languages used by GORE for BP methods. An exception to this are the different  $i^*$ -variants (used by Bleistein et al, Frankova et al, Grau et al, Koliadis et al, and Lapouchnian et al), which are comparable in terms of visual syntax, but differ in terms of semantics. Next, most GORE for BP methods have been illustrated by means of an example, but lack further validation in practical context. Related to this lack of detailed validation, is the low priority for offering practical guidelines to the modeller, which might block industry adoption of GORE for BP methods. Furthermore, the maturity of the transformation activity is

overall sufficient, but again misses concrete details about concept mappings or about practical guidelines for step-by-step transformation. Finally, the majority of the GORE for BP methods were created for generic business settings with typical strategic requirements and operational business processes, but some GORE for BP methods were restricted for application to secure business processes, non-functional requirements for business processes and high-variability processes.

## 2.5 Conclusions

Since 1999, a steady increase on research in Goal-Oriented Requirements Engineering for Business Processes (GORE for BP) is observed, but little attention has been given to the analysis of the available methods. We believe that there is a need for a general investigation of GORE for BP methods. To this end, it is challenging to compare GORE for BP methods as we lack understanding of the unifying framework of such a method (e.g. what are the typical activities of such a method) and we have no overall view on the currently existing GORE for BP methods. Hence, the knowledge claim [WMMR06] made in this chapter is that we will provide insights in the comparison of GORE for BP methods, and therefore need to answer two knowledge sub-problems, i.e. we need to find the unifying framework of GORE for BP methods to enable the method comparison, and we need to identify the currently available GORE for BP methods that are sufficiently well-documented to understand the methods as an outsider that was not involved in the development of these methods.

This chapter made a number of research contributions, which we will summarize in this section, and elaborate on the importance of the contribution to the RE community. Our first contribution (answering RQ1.1) is extending Kavakli's [KL99b, Kav02] unified framework for GORE methods for deriving a unified framework for GORE for BP methods. More specifically, we propose a GORE for BP method metamodel that allows us to model and compare individual GORE for BP methods in terms of method fragments that apply particular strategies to go from one knowledge-modelling state to another. The comparison between methods is possible because these knowledge-modelling states are the states of the Requirements Engineering for Business Process Management (REBPM) lifecycle, so each method fragment is positioned within the REBPM meaning that fragments from different methods that take the same position in the REBPM can be directly compared. As each of the seven GORE for BP methods that resulted from the systematic literature review could be analyzed using the proposed metamodel, we believe its utility is demonstrated. We also hope that our metamodel supports future researchers in their efforts to create well-structured and well-scoped GORE for BP methods as well as practitioners in their efforts to evaluate candidate GORE to BP methods that are considered for adoption.

Our second contribution (answering RQ1.2) is a systematic literature review in which GORE for BP methods are identified, considering all possible types of goal models and business process models. While performing the systematic literature review, we discovered large dif-

ferences in completeness of method descriptions, ranging from vague descriptions to detailed user manuals. As we are also interested in comparing GORE for BP methods (rather than just identifying them), we needed to reduce the subjective interpretation of the identified methods as much as possible. As a result, we identified seven GORE for BP methods that are sufficiently complete in terms of documentation to enable an objective evaluation. In order to guarantee a sufficient level of validity of our systematic literature study, we followed the guidelines of Kitchenham et al. [Kit04, KDJ04, BKB<sup>+</sup>07, KPBB<sup>+</sup>09] in a rigorous way, including the creation of a review protocol before the actual start of the literature study, conducting a pilot run to test and correct the review protocol and by selecting papers based on consensus of the three authors of this paper. In terms of relevance to the RE community, we hope to provide other researchers an interesting overview of the existing GORE for BP work, and stimulate current GORE for BP researchers to document and publish their work as much as possible in a clear and well-documented way.

Our third contribution (answering RQ1.3) is the application of the GORE for BP method metamodel to the seven identified GORE for BP methods that have a *complete* description, which we used to feed the discussion on similarities and differences of GORE for BP methods. To this end, we focused on six variables (*completeness in terms of REBPM lifecycle, kind of conceptual modelling languages, validation in practical context, applicability restrictions, practical guidelines offered to modeller, maturity of transformation activity*). Firstly, we believe that GORE for BP research should redirect attention from the specification activity to further investigate the elicitation and validation activities. Secondly, we would like to recommend researchers investigating GORE for BP to abstract their work from the individual goal and business process modelling languages, such that the methodological aspects of their work get separated from syntax aspects and a common methodological ground for GORE for BP can emerge. Thirdly, we would like to motivate future GORE for BP method researchers to support the modeller in a more explicit way by providing clear and practical guidelines. Fourthly, we would like to motivate current and future research on GORE for BP to focus on this transformation activity, and reuse work from related research areas on model-to-model transformations such as the Atlas Transformation Language (ATL) project [Ecl10a].

A limitation of our research is that it is not immediately applicable for method engineering purposes. Motivated by findings in method engineering [Bri96, BSH99], Kavakli's GORE framework was conceived to combine various GORE methods together into one improved framework that could bring together the benefits of the different GORE methods. Hence, the expression of different methods in terms of compatible method fragments might help requirement engineers to assemble fragments from different method models, thus generating new methods that better fit the needs of real projects and their contexts. In contrast, we extended Kavakli's unified framework [KL99b, Kav02] to investigate the challenges of GORE for BP methods and to discover the global trends in GORE for BP research, without claiming combinability of GORE for BP method fragments. Further, we only focused on

how GORE for BP methods could be expressed in terms of an overall *method metamodel*, but did not go into detail on the *metamodels of the conceptual modelling languages* per GORE for BP method. Hence, each GORE for BP method will yield a number of deliverables that are expressed in terms of different semantics and syntactic rules. As long as the deliverables of GORE for BP methods are not aligned to each other, we cannot claim combinability of GORE for BP method fragments.

To tackle the weaknesses of current GORE for BP methods, our future work aims at developing a new GORE for BP method. The research presented in this chapter will allow systematically developing this new method starting from the REBPM-based GORE for BP metamodel as well as using the knowledge of existing GORE for BP methods, possibly reusing method fragments from existing methods. The final aim is to present a tool-supported method to model goals as part of the requirements for business processes and to generate business process design skeletons that respond to these requirements. We are looking at the Eclipse platform to implement the tool-support, and are planning to use related Eclipse projects such as the Eclipse Modelling Framework (EMF) [Ecl10b], the Graphical Modelling Framework (GMF) [Ecl10c] and the Atlas Transformation Language (ATL) [Ecl10a].



# 3

## Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management

### 3.1 Introduction

Since organizations moved into Business Process Reengineering in the nineties [Har91, HC94, Dav93] and Business Process Management in the 2000s, business process designs are considered as valuable assets. As business processes are designed to execute the business activities taking place in an organization, managing business processes is crucial as they are the bridge between the strategy formulation and execution layers in an organization. Just as an organization's information needs sets requirements for its information systems, the formulation of strategies leads to business requirements for the organization's business processes. In a business process-centred organization, the architectural view on implementing business requirements through Business Process Management Systems is given by Figure 3.1. On the top layer, called *Strategy Thinking Layer*, artefacts such as business strategies, goals and policies are positioned. The layer below is the *Business Process Architecture Layer*, where the business process models that document the business processes reside. The third layer is the *Business Process Execution Layer*, where executable versions of the business process models on the layer above are managed in order to run the business. The bottom layer, called *Business Process Infrastructure Layer*, contains the IT infrastructural services (e.g. web services, service-oriented software applications) that are used by executable business processes in order to be run.

The ability to model *Strategy Thinking Layer* artefacts as business requirements for the business processes that constitute the organization's *Business Process Architecture Layer* would greatly facilitate the

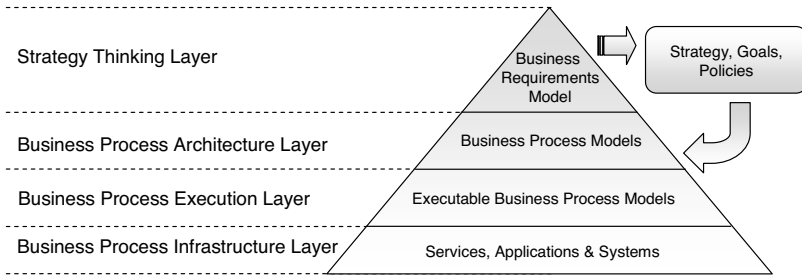


Figure 3.1: Implementing Business Requirements through Business Process Architectural Layer

realization of this architectural view. Modelling such business requirements would, for instance, allow for checking their consistency (e.g. is a newly formulated policy consistent with the policies in place?) as a new or modified business requirement model could be integrated with current business requirement models as part of one common company-wide *Business Requirements Model*, which is embodied in the *Strategy Thinking Layer*. By annotating the business process models in the *Business Process Architecture Layer* with elements from the *Business Requirements Model*, business process models can be checked for completeness with regard to the implementation of the business requirements. Model transformations could even create business process design skeletons out of business requirement models, by making sure that each business requirement model element is mapped to some description in the target business process models.

The overall goal of our research is to realize the architectural view captured by Figure 3.1. In this chapter, we focus specifically on the development and evaluation of a tool-supported method to model policies that reside at the *Strategy Thinking Layer* and to support the (re)design of business processes at the *Business Process Architecture Layer* with the goal of compliance management (i.e. making sure the business processes comply to the new or modified policies in the *Business Requirements Model*). Before elaborating on the concrete objectives of the chapter, we first provide a brief overview of the current challenges in Business Process Management with respect to our research goal.

### 3.1.1 Challenges

In order to model *Strategy Thinking Layer* artefacts, and policies in particular, as part of the *Business Requirements Model* and to generate business process design skeletons that correspond to the *Business Requirements Model*, the current state-of-the-art in Business Process Management still needs to tackle a number of challenges. Firstly, chapter 2 has shown that current techniques to transform strategy-incepted business requirement models into business process models do not offer satisfactory solutions. Although Requirements Engineering as a discipline has made significant advances in managing information system requirements, requirements engineering for business process design is a field that needs further development.



Secondly, current solutions for modelling policies and compliance checking on the level of business process models [EKSMP08a,EKSMP08b] do not take strategy-incepted business requirements into account. The European project SUPER [HR07] has proposed the Semantic Business Process Management approach which seeks to overcome the gap between business and IT by defining means for annotating the artefacts in the *Business Process Architecture* and *Execution Layers*; an operation meant to clarify their business semantics. The idea behind this approach is to enable the sharing of meaning between the organization's business/strategy level where business requirements for business processes are formulated and the operational/IT level where business processes are executed. Business Process Management is more and more taking responsibility for non-operational aspects of enterprise governance. For instance, as organizations are usually subject to a number of regulations, it has to be ensured that business is conducted in accordance to these regulations. Business process compliance management is the discipline that tackles the problem of how to model constraints in business process models and how to enforce them [EKSMP08a].

### 3.1.2 Objectives, hypothesis and overview of the approach

This chapter addresses these challenges and discusses and motivates the need for dedicated requirement engineering techniques for Semantic Business Process Management. In doing so, the chapter focuses on the specific issue of the early capture and specification of policies for the sake of business process compliance management. In attempting to deal with this matter, this chapter addresses the following research questions (cfr. Chapter 1 - Figure 1.7):

- RQ2.1. How can business users design semantic business processes in terms of and in correspondence with strategic requirements and policies?
- RQ3.1. Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.1?
- RQ3.2. What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.1?

In answering these questions, the chapter proposes an approach to model *Strategy Thinking Layer* policies as part of the business requirements for business processes and to generate business process design skeletons (captured in models on the *Business Process Architecture Layer*) that respond to these business requirements, and hence to take the formulated policies into account. In order to allow business users to model business requirements in a specific, yet intuitive way, we need to have an easy-to-use graphical tool and an accompanying method to build the business requirements space. Our first contribution is taking an existing early requirements specification language, i.e. Formal Tropos [FLM<sup>+</sup>04], and extending this language to incorporate policies, called Policy-extended Formal Tropos. Furthermore,

we created a graphical editor to allow business users to visually develop business requirement models and generate Policy-extended Formal Tropos instances from this. Our second contribution is offering a model transformation to create business process design skeletons out of the Policy-extended Formal Tropos models from the *Business Requirements Model*.

To realize this approach, a layered implementation architecture (Figure 3.2) has been developed in IBM's Eclipse environment. As our implementation architecture is detailed in section 3.4, we will start by introducing here the general idea underlying the architecture. The fundamentals of our solution are built upon the different abstraction layers of the OMG Model Driven Architecture [OMG09c].

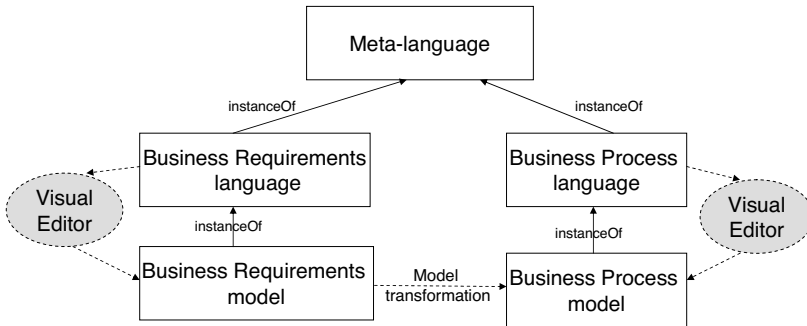


Figure 3.2: Implementing Policy-Enabled Goal-Oriented Requirements Engineering

On the top, a high-level meta-language (e.g. defining elementary constructs like Class and Relationship) is used to define medium-level languages (e.g. containing the instance of a Class called Policy), of which models are defined on the lowest-level (e.g. containing the instance of a Policy called ‘Do not enter toxic room’). In this chapter, we use two different medium-level languages, i.e. one language to define business requirements (Policy-enabled Formal Tropos) and another language to represent business processes (Business Process Modelling Ontology). Both languages have associated tool support to allow business users to generate visual model instances. As these visual tools only understand structural representations of languages, we introduce the concept of metamodel, which is a description of the language’s abstract syntax [Gui07]. We employ one business requirement metamodel to define the business requirement language and another business process metamodel to define the business process language.

The differentiation between a business requirement language and a business process language is based upon the belief that it adds value for a domain expert to create a business requirement model in order to generate a business process design skeleton instead of the direct creation of business process models. In particular with respect to policy modelling and compliance management, our working hypothesis is that it is useful and valuable to first model new or changed policies in a business requirement model and next to generate business process design skeletons out of this business requirement model, done in such a way that the changes to business process designs, needed to comply with the

new/changed policies, can be done more easily/effectively (compared to directly changing existing business process models).

It is important to understand that the target user of the proposed approach (called business user) is a domain expert who works in a business process-centred organisation and understands both high-level strategy concepts (such as business goals and policies) and low-level operational details (such as the way the current work is done).

The chapter is structured as follows: Section 3.2 introduces a running example on the introduction of a new regulatory policy in a company. This example will be used throughout sections 3.3 and 3.4 to illustrate our approach. Section 3.3 describes how we extended the Formal Tropos goal-oriented requirements specification language with policy-related concepts. Section 3.4 presents the different elements of our implementation architecture, including the generation of business process design skeletons out of the business requirements model. Section 3.5 presents three iterations of a pilot study that we used as a first empirical test of our approach. This pilot study provided user feedback to further improve the approach as well as first indications of its potential value as a solution to the problem of how to easily and effectively adapt business process designs to new or changed policies. Finally, section 3.6 contains related work and section 3.7 concludes this chapter.

## 3.2 Introduction to the Business Process Compliance Management Example

Let us observe a fictitious company X in the light of the introduction of a new regulation from an international trade organization. We will use this example in the remainder of the chapter to illustrate how our approach to incorporating policies works and what specific benefits it can offer. Company X has to observe new limitations on the trade of a certain mineral M (Figure 3.3). First, the mineral is extracted in Togo, then brought to South Africa for chemical refinement, and toxic defects are sent to Holland for treatment. Next, the refined mineral is sent to Greece in order to be shipped to customers. Company X has to ensure that not only all existing business processes (and by extension, the organizational entities and resources) involved in these steps respect the new limitations, but also that all business processes that will be created in the future, follow the latter.

The four process steps of the value chain shown in Figure 3.3 are each a separate complex business process. Each of these steps is carried out by a separate branch of company X, which may operate independently from the other branches. For each of these steps, specific governance requirements may be set. For instance, the third step, ‘Treat defects’, is carried out in the Netherlands and has a *risk* attached which expresses that not treating toxic defects can be very negative for the image of the company and generates unnecessary costs due to possible pursuits. Similarly, the final step called ‘Ship Mineral’ has a *business goal* attached to it, stemming from the contract template it has with its clients, where a clause specifies that all shipments to the US will always last less than three weeks.

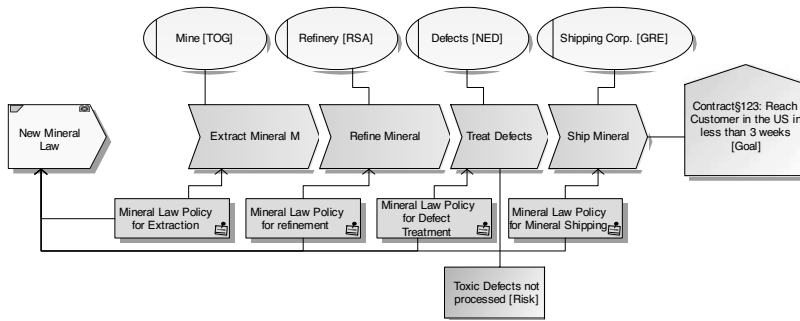


Figure 3.3: Policy Model Related to Extraction Business Process

The introduction of the new law on mineral M provokes the definition of four specific *policies*, each for one of the four steps for processing and trading mineral M. To comply to these new policies, the design of the existing business processes may need to be changed by process experts. These experts need to consult each other for potential new dependencies between their changed business process models. Then, an entity supervising the global enforcement of the new mineral law needs to check all changed business process models individually in order to guarantee compliance. Typically in business process modelling projects, teams of consultants gather business requirements (such as those originating in regulatory policies) that are hidden in artefacts, documents, corporate knowledge or simply the minds of people. These requirements are then directly modelled in business processes with no means of checking them for completeness or consistency (besides tedious and error prone manual checking).

### 3.3 Policy-Enabled Goal-Oriented Requirements Engineering

In the domain of Goal-Oriented Requirements Engineering [MCY99], several requirements specification languages have been proposed that explicitly recognize that systems are created to fulfil goals and that these goals can thus be considered as high-level (or early available) solution requirements [Kav02]. The *i\** modelling formalism [Yu97], for instance, can be used to clarify the organisational environment (e.g. Which actors with what intentions? Who depends on whom for realizing what?) in which systems are set. Although *i\** allows modelling various *Strategy Thinking Layer* artefacts related to goals, goal dependencies and goal realization, it currently lacks expressiveness to model policies. During the last decade, different *i\**-based languages [ACC<sup>+</sup>05] were created such as Eric Yu's original *i\** variant [Yu97], Goal-Oriented Requirement Language [LY04] and Formal Tropos [FLM<sup>+</sup>04, CKM02], which specifies a formal grammar of the *i\** language that is enriched with temporal constraints checking features (Formal Tropos was proposed as part of the Tropos [CKM02] project, and offers a formalisation of a specific subset of the Tropos project). Based on the capacity to verify constraints, we chose Formal Tropos as the basis for a policy-extended

goal-oriented requirements modelling language.

The first subsection introduces the main elements of Formal Tropos. The second subsection presents the extension with policies which we refer to as Policy-extended Formal Tropos. The third subsection applies policy modelling with Policy-extended Formal Tropos to the example introduced in section two.

### 3.3.1 Formal Tropos

We briefly describe the Formal Tropos grammar via the Backus-Naur-Form format, as presented in the work by Fuxman et al. [FLM<sup>+</sup>04, Fux01]. An example of a Formal Tropos model is given in Figure 3.4 and Figure 3.5.

```

Entity Course
Entity Exam Attribute constant course : Course
Actor Student
Actor Teacher
Goal PassCourse Actor Student Mode Achieve Attribute constant course : Course
Task GiveExam Actor Teacher Mode Achieve Attribute constant exam : Exam
Softgoal Integrity Actor Student Mode Maintain
Resource Mark Actor Teacher Mode Achieve

```

Figure 3.4: First Formal Tropos Example

Elements of the Formal Tropos grammar (given in Figure 3.6) are referred to as classes (which can represent, for instance, actors or goals). Intentional elements describe Goals (e.g. PassCourse), Softgoals (e.g. Integrity), Resources (e.g. Mark) or Tasks (e.g. GiveExam). They have a name, an explicit actor and a mode that describes the modality of fulfilment (i.e. Achieve, Maintain, Achieve&Maintain, Avoid). Non-intentional elements are represented as Entities (e.g. Exam), which can be used as attributes of other elements (e.g. Task GiveExam Attribute constant exam : Exam).

For each class described by the Backus-Naur-Form, several instances may exist, which have their own behaviour during the execution of the system. Similarly to classical object-oriented approaches, each class has a list of attributes, and each attribute has a type, called sort, which can be a primitive type (e.g. Attribute passed : boolean) or a class (e.g. Attribute exam : Exam). In addition, each attribute has one or many facets, which specify properties such as whether the attribute is constant or optional (e.g. Attribute constant exam : Exam).

```

(1) Resource Dependency Mark
(2)   Depender Student Dependee Teacher Mode Achieve
(3)   Attribute exam : Exam, passed : Boolean
(4)   Creation Not Fulfilled(passed)
(5)   Invariant Not  $\exists m : \text{Mark} (m \neq \text{self})$ 
(6)   Fulfilment condition  $\exists im : \text{InitialMarking} (\text{Fulfilled}(im))$ 

```

Figure 3.5: Second Formal Tropos Example

Furthermore, Formal Tropos includes three different types of constraints: creation-properties, invariant-properties and fulfilment-properties. Firstly, creation-properties are constraints that are checked and must hold at the instant in time when a class is instantiated (e.g. Figure 3.5 - Line 4: No mark may be created when the student already passed the exam). Secondly, invariant-properties are constraints that should always be enforced throughout the lifetime of each instance of a class

(e.g. Figure 3.5 - Line 5: There exists no second instance of the same Mark). Thirdly, fulfilment-properties are constraints that are enforced at the moment in time when an intentional element is fulfilled (e.g. Figure 3.5 - Line 6: an initial marking should exist to have a mark).

```

specification ::= (entity | actor | int_element | dependency | global_properties | int_rel |
policies)*;
entity ::= "Entity" name [attributes] [creation_properties] [invar_properties];
actor ::= "Actor" name [attributes] [creation_properties] [invar_properties];
int_element ::= type name mode "Actor" name [attributes] [creation_properties]
[invar_properties] [fulfil_properties];
dependency ::= type "Dependency" name mode "Depender" name "Dependee" name [attributes]
[creation_properties] [invar_properties] [fulfil_properties];
type ::= ("Goal" | "Softgoal" | "Task" | "Resource");
mode ::= "Mode" ("Achieve" | "Maintain" | "Achieve&Maintain" | "Avoid" );
attributes ::= "Attributes" attribute*;
attribute ::= facets name : sort;
facets ::= [constant] [optional];
sort ::= name | integer | boolean;
creation_properties ::= "Creation" creation_property+;
creation_property ::= property_category event_category temporal_formula;
invar_properties ::= "Invariant" invar_property+;
invar_property ::= property_category temporal_formula;
fulfill_properties ::= "Fulfillment" fulfil_property*;
fulfill_property ::= property_category event_category temporal_formula;
property_category ::= "constraint" | "assertion" | "possibility";
event_category ::= "trigger" | "condition" | "definition";
global_properties ::= "Global" global_property+;
global_property ::= property_category temporal_formula;
int_rel ::= "IntentionalRelationship" typeRel "source" int_element "target" int_element;
typeRel ::= ("MeansEnd", "ANDTaskDecomposition", "ORTaskDecomposition");
    
```

Figure 3.6: Formal Tropos Grammar [FLM<sup>+</sup>04]

The original Formal Tropos grammar was proposed to include temporal reasoning on top of the graphical i\* language, but lacks relations between intentional i\* nodes (e.g. there is no explicit reference between the Goal PassCourse and the Task GiveExam). As one of our intended contributions is to offer intuitive tool support to business users, we enriched the original Formal Tropos grammar with three kinds of intentional links as proposed by Ayala et al. [ACC<sup>+</sup>05] (MeansEnd, AND-TaskDecomposition or ORTaskDecomposition). A MeansEnd link indicates a relationship between an end (e.g. Goal PassCourse) and a means (e.g. Task GiveExam) for attaining it. A task node is linked to its component nodes by TaskDecomposition links, and could be arranged in logical AND or OR groups (e.g. Task GiveExam consists of Task CreatingExam AND Task MarkingExam).

### 3.3.2 Policy-extended Formal Tropos

We extended the Formal Tropos Backus-Naur-Form class specification (Figure 3.6) with the element policies, which allows defining a set of policies (as given by Figure 3.7). The reference work we rely on for this extension is the policy management framework Rei [Kag04]. Rei has two big advantages, it is semantically defined, which allows integrating it in a semantic framework such as the European project SUPER, and supports semantic domain models, which map to the domain ontologies that should be supported by our approach for modelling policies.

A Policy.Set contains several Policies related to one Actor, and are attached to each other with a Boolean operator (e.g. Figure 3.8: Policy.Set NED\_POLSET is attached to actor DT\_NED, which has Policy NED\_POL1 AND Policy NED\_POL2). A policy has a name and specifies policy targets, which are intentional elements, actors, or entities (e.g. Task TREAT\_TOXIC\_DEFECTS is targeted by both policies in Figure 3.8). Furthermore, a policy has a mode, which is composed of two types, a deontic (prohibition, permission, obligation, dispensation) and

```

policies ::= "policy_set" name "Actor" name (policy boolean_operator)* policy;
policy ::= "Policy" name policy_targets policy_mode [dependency_properties]
[policy_constraint] [attributes] [creation_properties] [invar_properties]
[fulfill_properties];
policy_targets ::= "PolicyTarget" policy_target+;
policy_target ::= (intentional_element | actor | entity);
policy_mode ::= "Polmode" mode dmode | amode;
dmode ::= "Deontic" ("prohibition" | "permission" | "obligation" | "dispensation");
amode ::= "Alethic" ("necessity" | "non-necessity" | "possibility" | "non-possibility");
dependency_properties ::= " PolicyDependency" dependency_property+;
dependency_property ::= "ActiveDepender" name "PassiveDepender" name ("delegation" |
"invocation" | "revocation" | "cancellation") name;
boolean_operator ::= "AND" | "OR" | "XOR" | "NOT";
policy_constraint ::= "PolicyConstraint" property_category event_category (temporal_formula |
structural formula);

```

Figure 3.7: Policy-extension to Formal Tropos Grammar

an alethic (possibility, impossibility, necessity and non-necessity) mode. Deontic modes are useful when modelling behavioural constraints on elements, while alethic modes are useful for modelling structural constraints. For instance, Policy NED\_POL1 *obliges* DT\_NED to fulfil task TREAT\_TOXIC\_DEFECTS, which is a *deontic* concept (Figure 3.8). In contrast, from an *alethic* point of view, DT\_NED might not be expected to treat the toxic effects themselves, which could be modelled by means of a *non-necessity*. In that context, Policy NED\_POL2 can *permit* DT\_NED to delegate the fulfilment of task TREAT\_TOXIC\_DEFECTS to the actor SUB.CONTRACTOR (Figure 3.8).

```

ACTOR DT_NED
  Attribute localization : String, subcontractor : SUB_CONTRACTOR
ACTOR SUB_CONTRACTOR

TASK TREAT_TOXIC_DEFECTS
  Mode Achieve
  Actor DT_NED
  Attribute reception_date : Date, finalization_date : Date

POLICY_SET NED_POLSET
  Actor DT_NED
  POLICY NED_POL1
    PolicyTarget TREAT_TOXIC_DEFECTS
    Deontic obligation
    PolicyConstraint Constraint Condition TREAT_TOXIC_DEFECTS.finalization_date -
TREAT_TOXIC_DEFECTS.reception_date ≤ 90 days

  AND
  POLICY NED_POL2
    PolicyTarget TREAT_TOXIC_DEFECTS
    Deontic permission
    PolicyDependency ActiveDepender DT_NED
    PassiveDepender SUB_CONTRACTOR delegation NED_POL1

```

Figure 3.8: Part of Policy-extended Formal Tropos for Company X (Extract from Appendix A - Figure A.1)

A policy constraint restricts the behaviour of a policy, and is defined similarly to Formal Tropos creation constraints. Policy constraints can be defined by specifying a property category (e.g. PolicyConstraint Constraint) and an event category (e.g. PolicyConstraint Condition), and can have either structural constraints or temporal constraints (e.g. Figure 3.8: make sure that 3 months after reception the treatment of defective goods is finalized during Task TREAT\_TOXIC\_DEFECTS, i.e.  $finalization\_date - reception\_date \leq 90days$ ).

A policy can define dependencies between Formal Tropos-model classes which are the targets of the policy. Policy dependencies always specify the two parts of such a dependency, namely the active depender and the passive depender. Both are either intentional elements, actors, entities or normal Formal Tropos dependencies (not policy dependencies). There are four possible policy dependencies: delegation, invocation, revocation, cancellation. For instance, company X defines a permission for Actor DT.NED to delegate fulfilling task TREAT\_TOXIC\_DEFECTS

to another actor called SUB\_CONTRACTOR. This delegation creates a dependency between NED\_POL1 and NED\_POL2, because the subcontractor now needs to comply to NED\_POL1 (i.e. fulfil the task TREAT\_TOXIC\_DEFECTS within 3 months after reception of the goods).

When using the visual Policy-extended Formal Tropos editor to model the policies induced by the new mineral law (Figure 3.3), we obtain results like those displayed in Figure 3.9. The visual editor consists of three main areas, i.e. the concept menu (right), the property part (below) and the modelling area (middle). All concepts in the right-menu correspond to Policy-extended Formal Tropos concepts existing in our Policy-extended Formal Tropos grammar. Appendix A - Figure A.1 includes the complete Policy-extended Formal Tropos specification for the company X example obtained via the visual editor that we have developed.

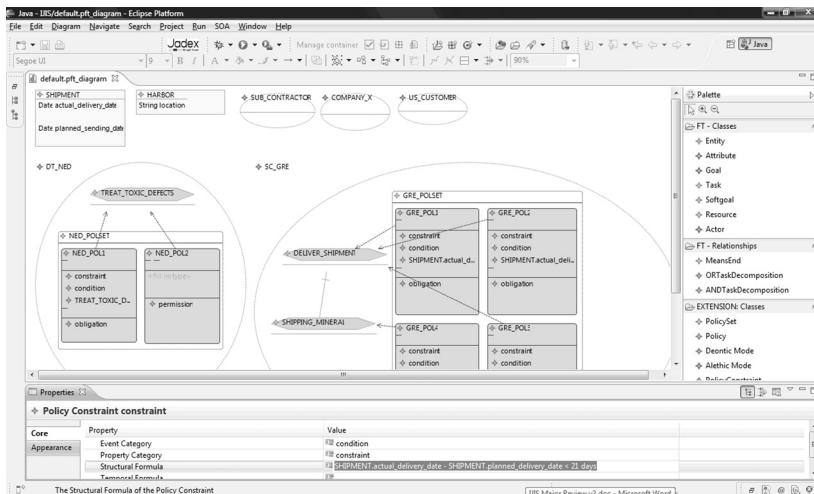


Figure 3.9: Eclipse-based Visual Policy-extended Formal Tropos Editor

### 3.4 Detailed Implementation Architecture

In this section we detail the layered implementation architecture that has been introduced before (Figure 3.2). This implementation architecture has been developed in IBM’s Eclipse environment. For using Eclipse for conceptual modelling, several projects have contributed heavily to standardise metamodel specifications and surrounding metamodel tool support. For that reason, all implementation work was done by means of Eclipse projects, more specifically the Eclipse Modelling Framework [Ecl10b], the Graphical Modelling Framework [Ecl10c] and the Atlas Transformation Language [Ecl10a]. Henceforth, we chose Eclipse’s Ecore metamodel as the specification of the high-level meta-language of our implementation architecture such that the Policy-extended Formal Tropos metamodels and the Business Process Modelling Ontology metamodels are specified as Ecore models. As given by Figure 3.10, our detailed implementation architecture consists of three main parts, i.e. business requirement modelling (in specifically policy modelling) with



Policy-extended Formal Tropos (Section 3.4.1), business process modelling with Business Process Modelling Ontology (Section 3.4.2), and transforming Policy-extended Formal Tropos models into Business Process Modelling Ontology models (Section 3.4.3).

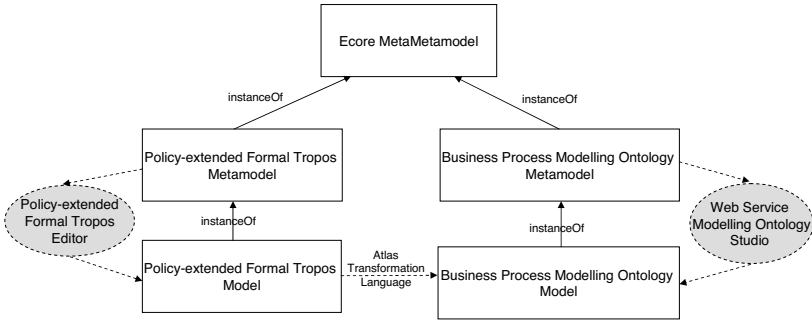


Figure 3.10: Implementation Architecture of Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management

### 3.4.1 Implementation of Policy-enabled Formal Tropos

The Eclipse Modelling Framework project offers a graphical framework to create a new metamodel as an instance of the Ecore metamodel. As our Policy-extended Formal Tropos grammar (displayed in Figures 3.6 and 3.7) is written in Backus-Naur-Form, we converted the Policy-extended Formal Tropos grammar into a Policy-extended Formal Tropos metamodel (in Ecore) following the steps described in Alanen and Porres [AP03]. The resulting Ecore metamodel of Policy-extended Formal Tropos is included in Appendix A - Figure A.2.

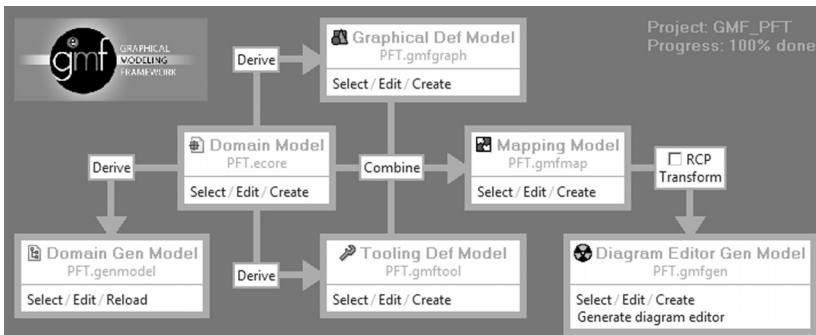


Figure 3.11: Employing the Graphical Modelling Framework to Implement Policy-extended Formal Tropos

In order to generate Policy-extended Formal Tropos models from the Policy-extended Formal Tropos metamodel, modellers might use text editors to manually write Policy-extended Formal Tropos instances. As writing formal specifications without graphical syntax is not very intuitive for humans, this approach would be costly, error-prone and time

consuming. Vessey [Ves91] explains that the intuitiveness of a language is a matter of suitability with respect to the task and the user profile, which again stresses the importance of adding a visual syntax for business users that want to use Policy-extended Formal Tropos. Fortunately, the Graphical Modelling Framework project takes an Ecore metamodel (such as our Policy-extended Formal Tropos metamodel) as an input and offers a step-by-step approach to generate a fully functional graphical editor. An overview of these steps during our Graphical Modelling Framework project is displayed in Figure 3.11.

Firstly, the Policy-extended Formal Tropos metamodel (referred to as the Domain Model in Figure 3.11) is used to derive a Java implementation of the domain model (Domain Gen Model). Next, a graphical definition model and tooling definition model are derived from the Policy-extended Formal Tropos metamodel, both of which are combined into a Policy-extended Formal Tropos mapping model. This mapping model includes the essential understanding of the Policy-extended Formal Tropos concepts, as explicit links are made between the Policy-extended Formal Tropos metamodel and the graphical environment that the business user will employ. For instance, in the Policy-extended Formal Tropos metamodel (see Appendix A - Figure A.2) the Ecore class `IntentionalElement` maps to a graphical node, whereas the Ecore class `IntentionalRelationship` maps to a graphical link that defines source and target attributes pointing to `IntentionalElement` nodes. Finally, the visual editor (Diagram Editor Gen Model in Figure 3.11) is generated based on the Policy-extended Formal Tropos mapping model.

### 3.4.2 Implementation of Business Process Modelling Ontology

Similarly to what we did for Policy-extended Formal Tropos, we created an Ecore metamodel that corresponds to the Business Process Modelling Ontology [SUP09]. This Ecore model can be found in Appendix A - Figure A.3. As the Business Process Modelling Ontology is a superset of the Business Process Modelling Notation, typical concepts such as `Process`, `SubProcess`, `Events` and `Tasks` are included, together with additional business semantics concepts like `BusinessPolicy`, `BusinessProcessGoal` and `BusinessDomain`. Furthermore, there was no need to extend the Business Process Modelling Ontology the way we extended Formal Tropos because the Business Process Modelling Ontology is already policy-enabled.

In order to create Business Process Modelling Ontology models from the corresponding metamodel, existing tool support offered by the SUPER project can be used. For instance, the Web Service Modelling Ontology Studio [SUP07] allows to graphically create semantic business process models and allows modellers to add ontological annotations to the business process view.

### 3.4.3 Implementation of Transforming Policy-extended Formal Tropos to Business Process Modelling Ontology

The Atlas Transformation Language project offers a transformation language and tools to execute the transformations. The basic requirements to run an Atlas Transformation Language project are having a source Ecore metamodel, a target Ecore metamodel and an instance of the source Ecore metamodel. Based on the defined Atlas Transformation Language mappings, an instance of the target Ecore metamodel will be generated from the source instance.

Related to transforming goal models (enriched with business semantics) into Business Process Modelling Ontology diagrams, initial research was done by Decreus and Poels [DP09]. Unfortunately, the proposed mappings are informal and incomplete, which makes them difficult to implement using Atlas Transformation Language. A revised version of these mappings is given in Table 3.1, and their full implementation is displayed in Appendix A - Figure A.4.

Policy-extended Formal Tropos	BP Modelling Ontology
Actor	Process
Task (With Children)	SubProcess
Task (Without Children)	Task
Goal	BusinessProcessGoal
Policy	BusinessPolicy

Table 3.1: PFT2BPMO Concept Mappings

In order to implement the transformation as displayed in Figure 3.10, we put the Policy-extended Formal Tropos metamodel (Appendix A - Figure A.2) as source, the Business Process Modelling Ontology metamodel (Figure A.3) as target, and used the Company X example to test the transformation. As current business process modelling languages such as the Business Process Modelling Ontology only encode typical business process constructs as graphical symbols (e.g. activity is encoded as a box, but policies have no graphical encoding), we have to show the results partly graphically (Figure 3.12) and partly textually (Figure 3.13). The full textual result is included in Appendix A - Figure A.5.

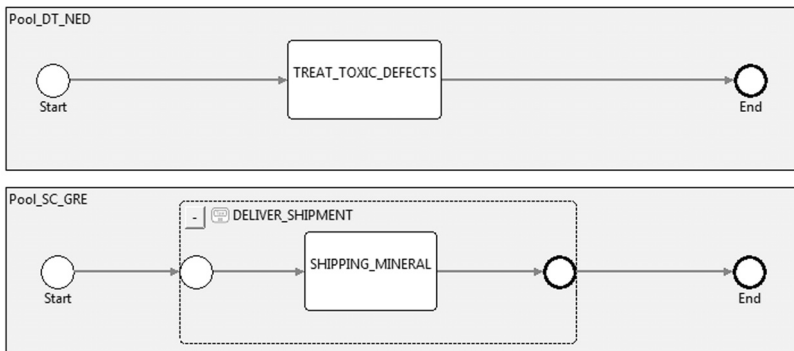


Figure 3.12: Graphical Output for Company X

```

instance NED_POL1 memberOf BusinessPolicy
hasName hasValue "NED_POL1"
hasPolicyType hasValue "ConstraintPolicy"
policyActsOn hasValue DT_NED
hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Obligation"
structuralConstraint hasValue "TREAT_TOXIC_DEFECTS.finalization_date -
                                TREAT_TOXIC_DEFECTS.reception_date ≤ 90 days"
isImplemented hasValue TRUE

```

Figure 3.13: Part of Textual Output for Company X (Figure A.5)

## 3.5 Evaluation

### 3.5.1 Introduction

In order to validate proposed solutions, Wieringa [Wie05] recommends to investigate the properties of the specified solution and to evaluate whether the proposed solution will work in real settings. To reach this objective, different empirical methods have been suggested [ML08, ZW98,SR07], such as surveys, history analysis, laboratory experiments, field experiments and case studies.

The decision matrix of Yin [Yin09] offers three conditions to identify the appropriate research methods: (a) the type of research question posed, (b) the extent of control an investigator has over actual behavioural events, and (c) the degree of focus on contemporary as opposed to historical events. We are looking for a research method that deals with exploratory ('how') questions as we want to get feedback on the use and perceived value of our method. Furthermore, we are interested in reducing the risks of employing our method into a real world setting, which does not require research methods with high control over behavioural events (like controlled experiments). There is also no need to focus on historical events. Hence, based on Yin's decision matrix [Yin09], the case study research method seems like a good candidate.

As our proposed solution is a novel method that introduces new techniques and makes new use of existing models and tools, it is possible that solution errors are found early on in the case study process. Therefore, before undertaking one or more case studies, with participants that have a real stake in the results, we decided to conduct a number of small-scale pilot studies (where results, if negative, wouldn't harm the people and organizations involved) in order to evaluate and further refine our solution before considering a larger-scale deployment.

The importance of conducting and reporting on pilot studies, meaning trial runs done in preparation for full-scale studies, is well-known in literature [vTH03, vTRHG01, PS89]. For instance, Prescott and Soeken [PS89] report that pilot studies play a critical role in the success of well-conducted research, while being underdiscussed, underused and underreported. Furthermore, Turner [Tur05] acknowledges the role of a pilot study to mitigate the risk on projects and to prove the technical feasibility of new techniques. Within the field of information systems, Glass [Gla97] confirms that there is surprisingly little written on how to conduct pilot studies, and that current usage is all too frequently either biased or inadequate.

In order to meet this need, Glass [Gla97] proposes a detailed set of

steps to be performed during a pilot study and states that a rigorously executed pilot study should be treated as a valid research method. In this perspective, a pilot study is “an objective study of a new technological concept in a somewhat realistic setting” ([Gla97], p86). In the next sub-sections, we will employ the detailed guidelines of Glass [Gla97] in order to discover and mitigate the current risks of our method not being adequate, to adjust or refine it if necessary, and to find first indications of the added value of our method.

### 3.5.2 Pilot Planning

The guidelines of Glass [Gla97] with respect to the planning of a pilot study and the way we approached them are:

- *Define the problem.* We define the problem as how to design business processes such that they comply with a given set of policies (as a special case of the more general problem of how to make business processes respond to strategy-level business requirements). We refer to these (documented) business process designs as policy-enabled business process models.
- *Select alternatives to study.* In order to create policy-enabled business process models, one could use a business process modelling tool to create new or adapt existing business process models (to given policies), or one could employ our method in which policies are modelled at the *Business Process Architecture layer* (i.e. *Strategy Thinking Layer*) and next generate policy-enabled business process models using the concept mappings that we developed.
- *Identify key independent variables.* Although pilot studies can not be used to correlate independent and dependent variables, because their small-scale does not allow ensuring statistical conclusion validity, Glass [Gla97] recommends to explicitly mention the key independent variables. The variable we would like to investigate is ‘Kind of Method Used’, as we introduce a new kind of method for policy-enabled business process modelling.
- *Identify variables to control.* Although pilot studies can not control variables, Glass [Gla97] recommends to identify control variables for guiding the focus of the pilot study. As we want to focus on the effect of introducing our modelling method and associated techniques, we identify the differences in modelling skills and differences in domain knowledge as possible variables to control.
- *Define success criteria.* We will consider our pilot study a success when it provides useful indications on how to improve our method as well as first indications of its potential value as a solution for the defined problem.

### 3.5.3 Pilot Design

Similarly to the guidelines for pilot planning, we discuss here how we approached guidelines with respect to the design of the pilot study.

- *Iterations of our pilot study.* We will organise three iterations of the same pilot study design, and based on the lessons learned from each iteration, we will adjust the necessary elements in the subsequent iteration.
- *Selection of organisations.* During our pilot study research, we envision a generic industry setting to explore the current limitations and potential of our method, so no explicit choices were made to target certain industries. Nevertheless, we wanted to execute our three pilot study iterations in different industries to increase chances in discovering different items.
- *Define pilot tasks.* Each pilot study iteration will be done with two participants, who are colleagues and who share responsibility in executing a certain business process. We will then induce a number of policies that the business process should meet and which require changes to the business process design. One participant will use a regular business process modelling tool to make the changes, and the other participant will use the method presented in this chapter.
- *Elaborate operational success criteria.* Indications for possible improvement of our method will be considered useful if they result in actionable problem items. Indications of potential value are those that are expressed by the study participants based on their perceptions of the ease of using the method and the quality of the resulting business process model, when comparing the use of our method and the use of regular business process modelling tools.
- *Define mechanisms for obtaining data.* Our data will be obtained in two ways, i.e. in the form of feedback obtained from participants during and after the study and by means of reviewing the models created by the participants.
- *Define evaluation mechanism.* After each pilot study iteration, the feedback collected from participants and the resulting policy-enabled business process models will be evaluated and compared (as two different approaches are used) in terms of the operational success criteria.
- *Define data validation approach.* One of the researchers is present and assists the participants with their pilot study tasks. Another researcher reviews the notes made by the first researcher to check for obvious (human) mistakes (e.g. attributing participant comments to the wrong method).
- *Cost vs. benefit definition.* When considering the cost of using our method balanced against the benefits achieved by its use, we see the costs as the possible risk of the method not being adequate (e.g. too difficult to apply, too much effort to apply, generated models of lower quality than models obtained through the use of business process modelling tools, etc.) and the benefits as the discovered indications of added value, meaning that the method is an improvement of the current practice (or lack of it) in developing policy-enabled business process models.

- *Define milestones and deliverables.* We will start each pilot study iteration by doing an interview with both participants to select an appropriate business process and related set of policies (first milestone). Next, one participant should design the policy-enabled business process directly using a business process modelling tool (second milestone - first deliverable), whereas the other participant will independently from the first participant use our method (third milestone - second deliverable). The discussion of the pilot study iteration results can be considered a fourth and last milestone.
- *Define ways to control variables.* Although pilot studies lack the facilities for controlling variables compared to experiments, Glass [Gla97] recommends to control them as much as possible. We plan to control independent variables ‘modelling skills’ and ‘domain knowledge’ by choosing pairs of participants that share as much as possible a same background with respect to their experience with and knowledge of business process modelling and the business process chosen for the study. More information about participants of each pilot study interaction is given in Section 3.5.4.
- *Choose statistical approaches.* The number of data points will be too low to justify the use of statistical techniques. We therefore opt for a qualitative interpretation of the data.
- *Define confidence factor approach.* As it is important how much one believes in the data, good numbers can be given more weight than questionable ones. Our pilot study deals with participants feedback and their created models, which makes it difficult to give certain results more weight than others. As a result, we give all resulting findings equal weights.

### 3.5.4 Pilot Conduct

The pilot was conducted according to the pilot design, incorporating the execution of the processes, gathering the needed data and delivering the required deliverables. As we focused our research towards business process-centred companies, we selected three companies that are client-focused with strong knowledge of their way of working, and that maintain good documentation of their main business processes.

#### 3.5.4.1 First Pilot Study Iteration

The *Vlaamse Karate Federatie* is a small to medium sized sports organisation that is officially recognized and subsidized by the Flemish government and has 8000 members across 212 karate clubs. Typical responsibilities of the *Vlaamse Karate Federatie* concern management of membership, organisation of competition and education of karate trainers. The administrative director and management assistant were chosen to be the participants of this pilot study iteration, as they both have an end-to-end visibility on all business processes of the *Vlaamse Karate Federatie*. In the context of this first pilot study iteration, we focused on the business process ‘Referee Management’ because both employees are familiar with this business process and because this business

process has a clear set of related policies. In particular, the business process 'Referee Management' handles the coordination of all referees needed to organise a karate competition event. Important to note is that both participants are frequent modellers and active karate practitioners, which makes them suitable candidates for modelling the 'Referee Management' business process. Next, we selected three policies related to 'Referee Management': (a) In order to comply to the Referee Regulations, there must be six referees per competition area (b) When referees are chosen to assist during a competition match, the higher qualified referee is preferred to a lower qualified referee (Level A is the highest, Level D the lowest) (c) Referees should reply to Invitations within a fixed period of two weeks.

#### **3.5.4.2 Second Pilot Study Iteration**

The second organisation is a global marketer of athletic footwear, apparel and equipment, which tries to bring inspiration and innovation to every athlete in the world. Around the globe, the company operates in more than 160 countries and gives work to more than 30.000 employees. We conducted the second iteration of our pilot study in the Belgian branch of this company, and we selected the administrative director and a brand manager. Both employees have university-level degrees and have experience in business process modelling. The business process 'Flight Management' was chosen as both the administrative director and brand manager had thorough knowledge of this business process and clearly definable policies for this process could be selected. The business process 'Flight Management' is internationally standardized and obliges company employees to book their flights via a recognized travel agent. We identified two main policies related to 'Flight Management': (a) the type of flight ticket must be flexible enough to allow the traveller to change departure or arrival times without any extra charges, and (b) the flight reservation process should be finished within two working days.

#### **3.5.4.3 Third Pilot Study Iteration**

The third organisation is a secondary school, that provides education to 425 kids between the age of 12 and 18. The school employs different kinds of technology to communicate with students and teachers and to streamline the teaching processes. Hence, we identified two teachers who were responsible for technological support. For the study, the business process 'Student Exchange' was chosen because both participants had deep knowledge of this business process and the related policies were well understood. The business process 'Student Exchange' allows foreign students to visit the school, to participate in classes and to enjoy local culture. We identified three policies related to the housing part of 'Student Exchange': (a) the housing, including food and drinks, should be offered to the students without any charges, (b) a family that offers housing to foreign students may only accept a maximum of two students, and (c) foreign students that share the same housing should have the same gender.



### 3.5.5 Pilot Evaluation

Based upon the gathered data and deliverables from each pilot study iteration, the operational success criteria are applied to draw conclusions. As defined during the pilot design phase, indications for possible improvement and potential value were discovered via feedback of the participants during and after conducting the pilot study and based on our comparative review and the participants' perception of the quality of the resulting models. An overview of the results of the pilot study is given in Table 3.2.

	<b>Indications for possible improvement</b>	<b>Indications of potential value</b>
<b>Pilot Study Iteration 1</b>	(a) Difficult understanding of policy vocabulary	(b) Regular business process modelling resulted in semantically overloaded models
<b>Pilot Study Iteration 2</b>	(c) Lack of explicit method step description (d) Example-based introduction needed	(e) Complementary modelling, separate focus on policies and on business processes
<b>Pilot Study Iteration 3</b>	(f) No particular problems found (g) Importance of long preparation phase	(h) Policy modelling on a higher level as business processes

Table 3.2: Pilot Study Results

During the first pilot study iteration, we discovered (a) problems related to the policy vocabulary we used, such as reaching a common understanding of deontic and alethic modes (e.g. prohibition, permission, obligation, etc). We used the definitions provided by the Rei [Kag04] framework to explain our understanding, but this seemed to be too abstract. Hence, during the second pilot study iteration, we decided to enrich the Rei definitions with real-world examples to lower the barrier to understand the Rei terminology. Furthermore, we learned (b) that the policy-enabled business process model, resulting from regular business process modelling, semantically overloaded the business process modelling concept of Activity. Activities in the business process model were used to represent real activities performed in the course of a business process (e.g. Appendix A - Figure A.6 - Select Appropriate Referees) but also to represent the verification of policies (e.g. Appendix A - Figure A.6 - Check whether the number of referees is a multiple of six). Using Policy-extended Formal Tropos, no semantically overloaded concepts were found (e.g. Appendix A - Figure A.7 - Six Referees Needed is a structural constraint on BusinessPolicy Policy\_1).

During the second pilot study iteration, we took more time to explain the different policy concepts, and made sure that we reached a common understanding with the participants. This time, we discovered (c) a problem issue related to the use of our method, concerning the lack of explicit method step description (i.e. the participant that used our method wanted to have a more detailed description of the steps we asked him to do). For instance, the participant wanted to know whether Policy-extended Formal Tropos models should be constructed in a top-

down or bottom-up fashion (Appendix A - Figure A.11), and was recommended the top-down approach of Formal Tropos [FLM<sup>+</sup>04]. Another useful suggestion was (d) to teach the method to users by means of examples, for instance by using previous applications of the method to introduce Policy-extended Formal Tropos to modellers. Next, the overall feedback of the participants, when comparing the models they created with our method and the 'standard' approach and discussing their experiences, pointed at (e) the potential value of separately modelling business processes and the policies that these processes must adhere to. In their daily work, the participants focus on control-flow aspects of the business process, which frequently coincide with the 'best case' scenario of the business process. As a result of the pilot study, they discovered how they implicitly implemented policy activities in their daily work. For instance, several telephone calls and emails are sent to remind actors in the process to speed up their work (as the flight ordering should be done within 2 days), although this is not represented in the business process model of 'Flight Management' (Appendix A - Figure A.10).

During the third pilot study iteration, we paid attention to thoroughly explaining the policy vocabulary and to introducing the method and its use of tools and techniques by means of the example of Company X (Cfr. Section 3.2). Due to these changes in our approach, we needed more time for preparing the participants in understanding our approach, but as a result (f) we did not experience particular problems with our approach. The main concern (g) was that this pilot study iteration took considerable more time to complete as the previous two iterations, due to the comprehensive preparation work. Next, the user feedback was positive on (h) separate thinking about business processes and about policies, and having tool support to link both concepts.

### 3.6 Related Work

When looking for related work, two types of research can be identified. The first kind of research studies relate to applying goal-oriented requirement engineering to business process models in a generic setting, without focusing on compliance (see chapter 2). The second type of related research are studies that investigate the role of compliance in requirements engineering. These studies differ from our work by not providing methods to derive business process models from goal models. The studies we reviewed were:

- Ghanavati et al. [GAP07, GAP09, GSP<sup>+</sup>09], who introduce a requirements management framework to help organisations document their compliance and to manage the evolution of laws and business processes. In their framework, compliance is tracked with links between artefacts at three levels: official source documents that define legislation and organisational structures, goal models (using Goal-Oriented Requirements Language [Amy03]) that capture the objectives and requirements of both organisation and legislation, and business process models (using Use Case Maps [Amy03]) that define the business processes implementing organisational policies and that represent the steps mandated by legislation.

- Breaux and Anton, who propose a systematic methodology for acquiring legal requirements from regulations [BA05, BVA06, BA08, BAD08], in which text is annotated to identify fragments describing actors, rights or obligations, and a semantic model is constructed from these annotations.
- Kiyavitskaya et al. [KZB<sup>+</sup>08], who extend the work of Breaux and Anton and present a tool intended to provide automatic support for analyzing policy documents.
- Other work by Anton et al. presents a requirements management framework that combines delegation and refinement in a distribution system [BAS09], analyses privacy policy documents from nine financial institutions [AEH<sup>+</sup>04], and discusses techniques to align privacy policies with system requirements [AEC03]. The set of works by Anton et al. is complementary with the results presented in this chapter in that they can provide the necessary input for building Policy-extended Formal Tropos models.
- Darimont et al. [DL06] employ the goal-oriented requirements engineering methodology KAOS to model regulations by transforming regulation documents into three models related to goals, objects and threats.
- Siena et al. [SML<sup>+</sup>08] present the normative i\* framework for modelling laws that integrate legal requirements with system requirements, and evaluate the effectiveness and efficiency of this i\* extension in the European food sector.
- Rifaut and Dubois [RD08] explain how a goal-oriented approach can be used together with an ISO standard in order to measure the compliance of business processes against regulations and their associated requirements.
- Bandara et al. [BLMR04] elaborate on a goal-based approach to policy refinement, which allows the inference of system level goal that satisfy operational goals.

### 3.7 Conclusion

Until the 1990s, research in field of Requirements Engineering focused in the ‘what’-‘how’ range [vL00], i.e. requirements on data and operations were specified, but one could not capture ‘why’ they were there and whether they were sufficient for achieving higher-level objectives. The trend of Goal-Oriented Requirements Engineering tries to meet this need, and includes motivational semantics such as goals and objectives. Recently, due to the European project SUPER [HR07], traditional business process models are made ‘semantic’ to allow modellers to embed extra information in the model, such as the business goal that an activity should achieve (e.g. Reduce maverick spending by 5%) or the business policy related to activities in a business process (e.g. Flight tickets should be ordered via the appointed travel agent).

To answer RQ2.1, this chapter presents an approach for extending an established Goal Oriented Requirements Engineering technique, i.e.

Formal Tropos, to policy modelling, and to generate business process design skeletons out of these models. The main claim of this work is two-fold:

- The richness of early requirement specifications is augmented by the policy extension, which avoids the late discovery of compliance discrepancies while designing business processes.
- The task of creating semantic business process model descriptions which implement business requirements is made both easier (because of the generation of semantic business process design skeletons) and more accurate (since early-verified policies are taken into account in the semantic business process model generation).

Further on, this chapter presents three iterations of a pilot study that we used as a first empirical test of our approach. This pilot study demonstrates the feasibility of our approach and provided user feedback to further improve the approach (answering RQ3.1) as well as first indications of its potential value as a solution to the problem of how to easily and effectively adapt business process designs to new or changed policies (answering RQ3.2).

The main limitations of our approach relate to the user aspects of employing our tools and techniques. Firstly, there is a need for a more complete method covering our policy extension to guide the user in expressing his business requirements. Secondly, clear definitions on terminology should be used to reach a common understanding about policies and business processes. For instance, we need a clear definition of the intended semantics of the deontic and alethic modalities introduced in Policy-extended Formal Tropos, as compared to the semantics defined in the Rei policy management framework.

Although the three pilot study iterations provided a first round of feedback about our approach, evidence should be collected by means of a full-scale empirical study to confirm the early indications of added value. In doing so, we need objective measures to define user perception (on ease of use) and user satisfaction (on quality of result), and follow strict methodological guidelines such as those proposed by Yin [Yin09] in conducting case study research.

# 4

## Goal-Oriented Requirements Engineering for BPMN Modelling

### 4.1 Introduction

Central to the development of BPMS technology was the promotion of a new language, Business Process Modelling Notation (BPMN), which could be used to represent business processes. As given by the BPMN specification [OMG09a], the primary goal of BPMN is “to provide a notation that is readily *understandable* by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.” (p1, [OMG09a]). Silver [Sil09] stresses the importance of using BPMN as a *common language* between business and IT. Furthermore, Silver [Sil09] distinguishes different types of BPMN modelling, depending on the user category. Firstly, BPMN Level 1, or *descriptive* modelling, is geared towards the business user and offers a basic set of BPMN elements. Secondly, BPMN Level 2, or *analytical* modelling, supports the business analyst in using the complete BPMN notation to describe the activity flow precisely, including the exception paths. These models should be complete and consistent, but not yet contain technical details to make them executable. Thirdly, BPMN Level 3, or *executable* modelling, allows the technical developers to add process data, service interfaces and human task assignment that are needed to execute the BPMN models using BPMS technology.

When we look at BPMN Level 1, the business user is already expected to understand and work with BPMN concepts such as pool, lane, task, subprocess, start event, stop event, exclusive gateways, parallel gateways, sequence flow, and message flow, which are terms that, maybe apart from task, do not belong to the ordinary language used by business people. It is doubtful whether they (e.g., business process managers/owners, business process consultants, accountants, marketers, sales people, auditors, finance officers, stock managers, etc.)

think of business processes in terms of ‘lanes’, ‘pools’, ‘gateways’, and ‘events’. Havey [Hav06] warns that BPMN is not suited for business users, and stresses the importance of capturing requirements based on an approach that business users can understand. Fernandez et al. [FPGGD<sup>+</sup>10] confirm this finding and state that BPMN scores low on usability for business users.

Business managers also frequently need to deal with complex real-world problems (e.g., how to react to the entry of a new, low-cost service provider in the market?) that require considering simultaneously high-level strategic requirements and low-level operational details. However, Recker [Rec10] states that BPMN currently lacks appropriate concepts to support process decomposition and organisational modelling. Recker [Rec10] suggests to use a different, easier, and more business user adapted approach to process modelling with BPMN, by providing dedicated symbols for placing a process into its organisational and hierarchical context.

What seems to be missing in BPMN Level 1, i.e., the way that business users are supposed to use BPMN, is an explicit consideration of the strategic rationale of having certain business processes as well as support for describing business processes in terms familiar to business people. In attempting to deal with this matter, this chapter addresses the following research questions (cfr. Chapter 1 - Figure 1.7):

- RQ2.2. How can business users design complex business processes in terms of and in correspondence with strategic requirements?
- RQ3.3. Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.2?
- RQ3.4. What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.2?

### 4.1.1 Introducing Our Approach

To answer RQ2.2, we developed a new BPMN approach to business process modelling targeted specifically at business users (i.e., BPMN level 1 as referred to by Silver [Sil09]). We assume that the context of our approach consists of a real-world environment in which there is a strategic interest of business users in the design of the business processes. As Wieringa and Heerkens [WH06] explain, the solution design phase proposes an improvement to a problematic situation, and is based on specific solution properties. In our approach, the main solution properties are:

- consideration of the strategic rationale of having certain business processes, and having them organized in certain ways
- support for describing business processes in terms familiar to business users
- explicit linkage between business processes and strategic requirements

This approach heavily relies on previous Goal-Oriented Requirements Engineering (GORE) research, which aimed at developing ways to capture high-level strategic business requirements and use them to drive the system development process. To this end, we investigated the GORE for BP literature to find studies that apply goal-oriented requirements engineering to business process design (see chapter 2). We found that GORE for BP methods generally lack clear mappings between goal concepts and business process concepts and are short of detailed transformation descriptions. Therefore, we were not able to reuse their transformations in our research. Some methods, however, provide a sound basis on which we can build our approach, i.e., the B-SCP framework of Bleistein et al. [BCVP06b] and the work of Lapouchnian et al. [LYM07].

To start with, the B-SCP framework [BCVP06b] is a requirements engineering framework for organizational IT that directly addresses an organization's business strategy and the alignment of IT requirements with that strategy. Goal modelling is used to represent business strategy as requirements, and Jackson context diagrams [Jac00] to represent business and system model context. The strategy and context parts are integrated using a problem diagram framework [Jac00]. Strategy is first elicited using VMOST [Son99], an organizational alignment analysis technique. Then, an *i\** goal model [Yu97] is constructed using goal modelling rules for organizational motivation proposed by OMG's Business Motivation Model [OMG09b]. To refine requirements from a strategic, high-level problem diagram down to the lowest operational level, a progression of problem diagrams is used to represent this top-down hierarchy. In addition, the problem diagrams are briefly mapped to Role Activity Diagrams (RAD) [Oul95], but we did not reuse these mappings due to the lack of documentation, the mixture of goals and business processes, and the redundancy in the proposed mappings (see chapter 2).

Next, Lapouchnian et al. [LYM07] propose a requirements-driven method for configuration of high-variability business processes in terms of business priorities. This method is characterized by textual annotations to add control flow detail to goal models, which we will reuse in this chapter. For instance, the sequence annotation (‘;’) can be added to AND decomposition to indicate that all the subgoals are to be achieved in sequence from left to right. As we aim at BPMN Level 1 [Sil09], we only consider annotation of sequential AND decomposition, parallel AND decomposition, and OR decomposition. This annotation of control flow is organised per group of decomposed requirements (e.g., all sub-requirements of one requirement have a sequential AND decomposition).

In [DP10] we elaborated on the basis of these methods an initial outline of a *GORE for modelling with BPMN* approach. The first contribution of this chapter is the introduction of the completed approach with all implementation details. This chapter details how we adopted the B-SCP framework and operationalized it for our purposes (instead of using B-SCP for business-IT alignment purposes) by creating a *B-SCP metamodel* to define the abstract syntax of the B-SCP language and a *B-SCP editor* for the visual creation of problem diagrams. Our approach

thus offers an explicit serialization of the B-SCP model via the B-SCP editor and provides a tool-supported business process modelling method that is adapted to the needs of business users. In addition, we implemented the *control flow annotations* of Lapouchnian et al. [LYM07] in our B-SCP editor, and constructed *model transformations* from B-SCP (modelling the strategic requirements to which business process must correspond) to BPMN (modelling the business processes that correspond to strategic requirements) to support the business analyst in digesting the output of the business user.

The properties of our approach are supported by our implementation as follows. The consideration of the strategic rationale of having certain business processes, and having them organized in certain ways, is given by the new *B-SCP metamodel* and corresponding *B-SCP editor* that we developed to support business users in the visual creation of problem diagrams. The support for describing business processes in terms familiar to business users is given by the *adoption of the B-SCP framework* for goal modelling as well as the business priorities driven *annotation of control flow* in problem diagrams. The explicit linkage of business processes with strategic requirements is ensured by the *BSCP2BPMN transformation* of problem diagrams into business process diagrams.

## 4.1.2 Feasibility of our Approach

In order to answer RQ3.3, we demonstrate the feasibility of our approach by means of applying our approach to a case exemplar. Feather et al. [FFFVL97] define a *case exemplar* as a self-contained, informal description of a problem in a specific application domain. As Hevner et al. [HMPR04] explain, using different scenarios of a case exemplar is a descriptive design evaluation method, which demonstrates the feasibility of the designed artefacts. Originally, Bleistein et al. [BCVP06b] used the Seven-Eleven Japan (SEJ) case exemplar [NW04, BUMN97, Kun97, Li03] to demonstrate the feasibility of their B-SCP framework. As we extend their work, we also selected the SEJ case exemplar, which allows us to compare our work to the original B-SCP framework.

## 4.1.3 Added Value of our Approach

In order to answer RQ3.4, we conducted two case studies based on the principles of Yin [Yin09], Eisenhardt [Eis89] and Miles and Huberman [MH94]. In the first case study, we investigated why business users would benefit from using our approach. This setup is based on Yin's mechanism of *literal replication* [Yin09], which predicts and evaluates why our approach is useful. In contrast, the second case study looks at what happens when business users are not equipped with our approach (but instead use BPMN Level 1 [Sil09]), to investigate what problems business users experience when directly modelling business processes using BPMN. This mechanism is called *theoretical replication*, which Yin [Yin09] defines as predicting contrasting results, but for anticipatable reasons. We expect to find added value of our approach when the findings are contrasting in both case studies, i.e., when we find evidence that the solution properties of our approach support the



business user and when the lack of these properties in BPMN Level 1 hinders the business user.

As Yin [Yin09] and Eisenhardt [Eis89] explain, the development of theory plays a central role in case study research. Initially, before the conduct of any case study, a firm theoretical basis is needed to understand the propositions to investigate. Then, after conducting the case studies, the expected or contrasting findings should be compared to the developed theory, and the theory should be adjusted where needed. To this end, section 4.2 provides some insights into the theoretical basis for our approach. Next, section 4.3 provides an overview of our approach and introduces the full implementation details of it. Section 4.4 demonstrates the approach by means of the Seven-Eleven Japan case exemplar. Section 4.5 reports on the findings of our two case studies. Finally, section 4.6 concludes the chapter.

## 4.2 Theoretical Basis

The first solution property considers the strategic rationale of having certain business processes, and having them organised in certain ways. In order to position the strategic rationale for business users, we see that Locke and Latham developed the Goal-Setting Theory [LL90] using inductive research over a 25-year period, based on some 400 laboratory and field studies. These studies showed that specific goals lead to a higher level of task performance than do vague, abstract goals [LL06b]. The context in which these goals are set can differ. Business users have goals that come from top management [KN08], while IT-oriented users are dependent on goals originating from the business side [MCY99, vL00]. As businesses cope with chaotic environments that are continuously changing [Cib96], the goals of business users tend to change frequently, which can change the goals of IT-oriented users.

The second and third solution properties ensure that business users can describe business processes in familiar terms and explicitly link business process design to the strategic requirements for business processes. The key point of these properties is that they offer the business user (e.g., business process managers/owners/consultants) an appropriate level of abstraction to deal with the inherent complexity of considering simultaneously high-level strategic requirements for business processes and the low-level operational details of business process design. Moody [Moo09] introduces the *principle of complexity management*, which recommends to include explicit mechanisms for dealing with complexity during modelling. In this context, *complexity* refers to diagrammatic complexity, which is measured by the number of elements (symbol instances or tokens) on a diagram. As a result, visual notations must provide mechanisms for *modularization* and *hierarchical structuring* to effectively represent complex situations. Firstly, *modularization* reduces complexity of large systems by dividing them into smaller parts or subsystems. Cognitive load theory [MM03, Swe94] shows that reducing the amount of displayed information within the limitations of a human's working memory, improves speed and accuracy of understanding and facilitates deep understanding of information content. Secondly, *hierarchical structuring* allows systems to be

represented at different levels of abstraction, with complexity manageable at each level. When we apply these principles of complexity management to the B-SCP framework, we see that Jackson’s principle of projection and Yu’s intentional relations support modularization and hierarchical structuring. Jackson’s principle of projection [Jac00] refers to the ability to describe domain context according to various viewpoints, levels of abstraction, and degree of detail, and is used to decompose problems in subproblems. Furthermore,  $i^*$  [Yu97] relates the requirements of problems and subproblems by means of  $i^*$  relationships such as Means-End links or Task Decomposition links. As our approach is based on and extends the B-SCP framework, this support for complexity management in modelling is inherited by our approach (note that B-SCP simplified the  $i^*$  relationships as compared to the original  $i^*$  proposal by Yu [Yu97]).

### 4.3 Overview of Our Approach

Our approach to BPMN modelling for business users consists of four steps. First, the business user applies our visual B-SCP editor to create a hierarchy of problem diagrams. Secondly, the business user adds control flow annotations [LYM07] that are needed for BPMN model generation. Thirdly, the business user or analyst uses the computer-based model transformations to generate BPMN process model skeletons. Fourthly, the business analyst takes the BPMN process model skeleton as input for his work and creates a consistent and complete BPMN business process diagram (i.e., BPMN level 2 as referred to by Silver [Sil09]).

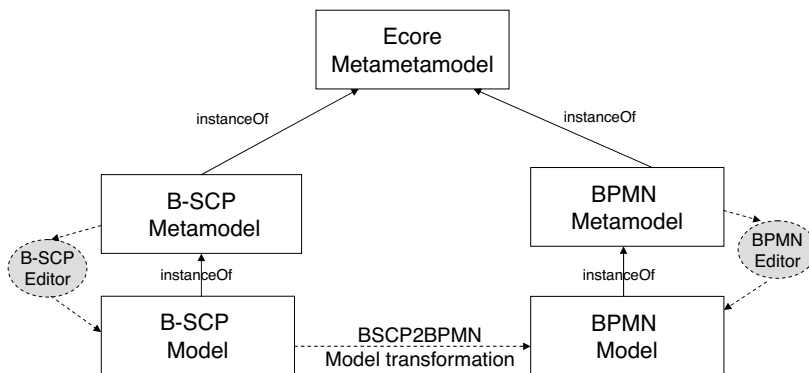


Figure 4.1: Overview of GORE for BPMN

To realize our approach, a layered implementation architecture (Figure 4.1) was developed in IBM’s Eclipse environment [Ecl10b]. The fundamentals of our solution are built upon the different abstraction layers of the OMG Model Driven Architecture [OMG09c]. On top, the high-level Ecore metamodel (e.g., defining elementary constructs like Class and Relationship) is used to define medium-level metamodels (e.g., containing the instance of a Class called Goal), of which models are defined on the lowest-level (e.g., containing the instance of a Goal called ‘Shorten Cash Cycle’). In this chapter, we use two differ-



regard to how it is to be achieved, and *Mission* indicates the ongoing activity that makes the vision a reality. For instance, EU-Rent could have a vision to ‘Be the car rental brand of choice for business users’, and a mission to ‘Provide car rental service across Europe for both business and personal customers’. Next, a *Goal* indicates what must be satisfied on a continuing basis to effectively attain the vision, and a *Strategy* is a long-term activity designed to achieve a goal. For instance, the goal ‘Be a premium brand car rental company’ tries to attain EU-Rent’s vision, and a strategy ‘Target major airports to find business users’ supports the achievement of the EU-Rents’ goals. Finally, an *Objective* is a specific and measurable statement of intent whose achievement supports a goal. All these requirements can be considered as strategic requirements for business processes. A *Tactic*, on the other hand, is a short-term action designed to achieve an objective. For instance, the objective ‘Be rated by AC Nielsen in top 6 car rental companies’ supports the EU-Rent’s goal to be a premium brand, and the tactic ‘Encourage rental extensions’ would be a short-term action to score better in listings such as AC Nielsen. Tactics guide the design of business processes such that these correspond to and help realize the strategic requirements.

*Requirements* described in *RequirementDiagrams* are interconnected via *Relationships*, such as *MeansEnd*, *ORDecomposition*, and *ANDDecomposition*. A *MeansEnd* link indicates a relationship between an end and a means for attaining it [BCVP06b]. For instance, the vision ‘Be the car rental brand of choice for business users’ is an end supported by its mission ‘Provide car rental service across Europe for both business and personal customers’ as means. Next, an *ORDecomposition* link indicates that a requirement is fulfilled if at least one of the lower-level requirements are fulfilled [BCVP06b]. For instance, a tactic ‘Handle Rental Extensions’ could be fulfilled by lower-level tactics such as ‘Use own staff to extend rental’ or ‘Use airport staff to extend rental’. Finally, an *ANDDecomposition* link indicates that a requirement is fulfilled if all lower-level requirements are fulfilled [BCVP06b]. In this chapter, we distinguish between sequential and parallel fulfilment of *ANDDecomposition* links. For instance, the tactic ‘Encourage rental extensions’ can be decomposed into two sequential tactics, of which ‘Persuade airport customers’ is the first in time and ‘Handle rental extensions’ is the second. In contrast, the tactic ‘Persuade airport customers’ might be decomposed into parallel tactics that can be executed at the same time, such as ‘Offer extra flight miles’ and ‘Offer free cabrio upgrade’.

A *ContextDiagram* contains at least two *DomainsOfInterest* and at least one *Interface* to connect a pair of *DomainsOfInterest*. For instance, *DomainOfInterest* EU-Rent has an interface with *DomainsOfInterest* business customer, personal customer and airport. An *Interface* should contain at least one *SharedPhenomenon* that is controlled by a specific *DomainOfInterest*. For instance, domains EU-Rent and airport might share phenomena such as airport location, welcoming of customer, or holiday season.

A domain of interest in the context diagram *describes* a part of the real-world, whereas a requirement *prescribes* the domain of interest in the context diagram. The connection between requirements and context is made by using the *refersTo* and *constrains* relations from a *Require-*

ment to a *DomainOfInterest*. For instance, the requirement ‘Be the car rental brand of choice for business users’ *refers to* domain EU-Rent, as this requirement involves the EU-Rent domain without constraining the way that EU-Rent becomes the car rental brand of choice. In contrast, the requirement ‘Use own staff’ *constrains* the domain EU-Rent Airport centre in making the staffing planning, as this requirement restricts the way that EU-Rent Airport organizes its staffing.

### 4.3.2 B-SCP Editor

In order to instantiate B-SCP models from the B-SCP metamodel we developed the B-SCP editor. The Eclipse graphical modelling framework [Ecl10c] project takes an Ecore metamodel (such as our B-SCP metamodel) as an input and offers a step-by-step approach to generate a fully functional graphical editor (details can be found at [Dec10]). When using the Eclipse-based B-SCP editor to model the running example of EU-Rent, we obtain results like those displayed in Figure 4.3. The visual editor consists of three main areas, i.e., the concept menu (right), the property part (below) and the modelling area (middle). All concepts in the right-menu correspond to B-SCP concepts existing in our B-SCP metamodel.

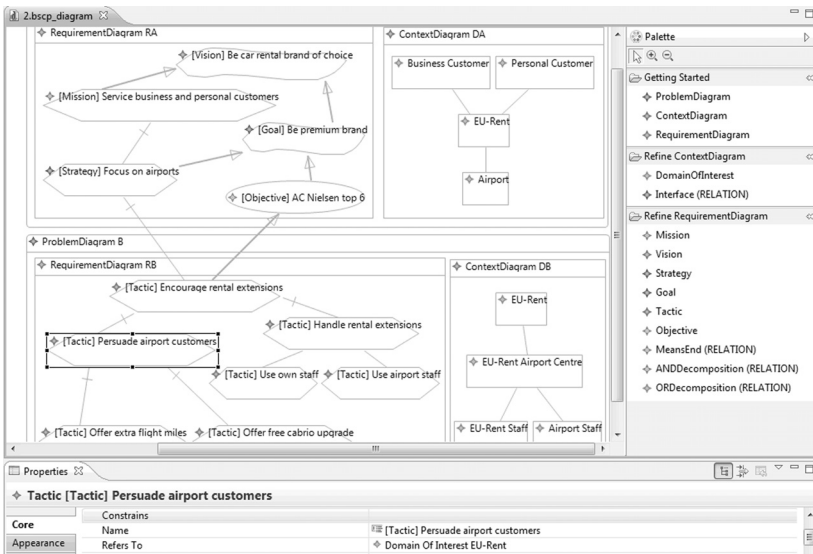


Figure 4.3: Eclipse-based Visual B-SCP Editor

### 4.3.3 Control Flow Annotations

In our approach, the *tactics* that are modelled in *RequirementsDiagrams* play a central role. Instead of forcing the business user in expressing the difference between business processes, subprocesses and activities, the business user just employs tactics wherever he specifies how strategic requirements (i.e., *mission*, *vision*, *strategy*, *goal* and *objective*) are or should be achieved. Consequently, a B-SCP model might contain numerous problem diagrams, some without *tactics*, some with

both *tactics* and other requirements, and some solely consisting out of *tactics*. So, when a business user creates a B-SCP *ProblemDiagram* that solely exists of tactics, we consider this problem diagram to represent (a part of) a business process, and it becomes useful to add control flow annotations. Although Lapouchnian et al. [LYM07] recommend textual annotations to add control flow detail to requirement models in a explicit and visual way, we choose to add such annotations via the *properties* pane of the B-SCP editor to lower the visual complexity of the models. As indicated by the B-SCP metamodel (Figure 4.2), the different types of *Relationships* (*MeansEnd*, *ANDDecomposition*, *ORDecomposition*) have a source and target *Requirement*. From a vertical perspective, a Relationship considers the upper *Requirement* as source, and the lower *Requirement* as target. For instance, Figure 4.4 shows that *Handle rental extensions* is OR-decomposed by *Use own staff*, Figure 4.5 shows that *Encourage rental extensions* is AND-decomposed by *Persuade airport customers* (in *sequence* with other AND-decompositions such as *Handle rental extensions*), and Figure 4.6 shows that *Persuade airport customers* is AND-decomposed by *Offer extra flight miles* (in *parallel* to other AND-decompositions such as *Offer free cabrio upgrade*).

◆ OR Decomposition		
Core	Property	Value
Appearance	Has Source	◆ Tactic Handle rental extensions
	Has Target	◆ Tactic Use own staff

Figure 4.4: Using OR Decomposition

◆ AND Decomposition Sequence		
Core	Property	Value
Appearance	Has Source	◆ Tactic Encourage rental extensions
	Has Target	◆ Tactic Persuade airport customers
	Type	☰ Sequence

Figure 4.5: Setting AND Decomposition to Sequence

◆ AND Decomposition Parallel		
Core	Property	Value
Appearance	Has Source	◆ Tactic Persuade airport customers
	Has Target	◆ Tactic Offer extra flight miles
	Type	☰ Parallel

Figure 4.6: Setting AND Decomposition to Parallel

### 4.3.4 Model Transformation B-SCP to BPMN

When a B-SCP *ProblemDiagram* meets the transformation criteria, our model transformations can be used to transform this diagram into the

skeleton of a BPMN business process diagram that can be further refined by a business process analyst to achieve completeness and consistency (that is, BPMN Level 2 [Sil09]). Our transformation criteria are as follows. As business process designs should not embed the definition of strategic requirements [Oul95], only the problem diagrams that solely consists of *tactics* are suitable to be syntactically transformed into BPMN business process diagram skeletons. To avoid the generation of non-linked BPMN lanes and activities, we require that each tactic should refer to or constrain one domain of interest (in a context diagram), and there should be at least one shared phenomenon on each interface between domains of interest. Next, the top tactic of the requirement diagram (e.g., tactic *Encourage rental extensions* in Figure 4.3) gives the name of the business process in scope of the transformation. Finally, as Lapouchnian et al. [LYM07] advises, control flow annotations should be consistent per group of tactic decompositions. For instance, tactic *Encourage rental extensions* (Figure 4.3) has two sequential AND decompositions, tactic *Persuade airport customers* (Figure 4.3) has two parallel AND decompositions, and tactic *Handle rental extensions* (Figure 4.3) has two OR decompositions.

Next, we will elaborate on the B-SCP to BPMN concept mappings (Table 4.1) that we created, which are implemented by means of the atlas transformation language [Ecl10a] (the implemented rule expressions can be found in Appendix B - Figure B.1, and technical details are given in Appendix C). In general, Rules 1 to 4 are used to transform the main concepts, Rules 5 to 9 transform the control flow annotations, and Rule 10 takes care of the generation of message flows.

- Rule 1 transforms a top node in a *RequirementDiagram* (e.g., Figure 4.3 - *Encourage rental extensions*) into business process diagram (e.g., the diagram shown in Figure 4.7).
- Rule 2 transforms a domain of interest (e.g., Figure 4.3 - *EU-Rent*) into a pool, a start event, a sequence edge, and an end event (e.g., Figure 4.7 - Labelled with (2)).
- Rule 3 transforms a medium node (e.g., Figure 4.3 - *Persuade airport customers*) of a *RequirementDiagram* into a sub-process (e.g., Figure 4.7 - Labelled with (3)).
- Rule 4 transforms a leaf node (e.g., Figure 4.3 - *Offer extra flight miles*) of a requirement diagram into a task (e.g., Figure 4.7 - Labelled with (4)).
- Rule 5 transforms the first occurrence of an OR Decomposition (e.g., Figure 4.3 - Link between *Handle rental extensions* and *Use own staff*) into two Gateway Data-Based Exclusive activities and two sequence edges (e.g., Figure 4.7 - Labelled with (5)).
- Rule 6 transforms the other occurrences of an OR Decomposition (e.g., Figure 4.3 - Link between *Handle rental extensions* and *Use airport staff*) into two sequence edges (e.g., Figure 4.7 - Labelled with (6)).

- Rule 7 transforms an AND Decomposition with sequence (e.g., Figure 4.3 - Link between *Encourage rental extensions* and *Persuade airport customers*) into two sequence edges (e.g., Figure 4.7 - Labelled with (7)).
- Rule 8 transforms the first occurrence of a parallel AND Decomposition (e.g., Figure 4.3 - Link between *Persuade airport customers* and *Offer extra flight miles*) into two Gateway Parallel activities and two sequence edges (e.g., Figure 4.7 - Labelled with (8)).
- Rule 9 transforms the other occurrences of parallel AND Decomposition (e.g., Figure 4.3 - link between *Persuade airport customers* and *Offer free cabrio upgrade*) into two sequence edges (e.g., Figure 4.7 - Labelled with (9)).
- Rule 10 transforms a shared phenomenon, between two domains of interest (e.g., Figure 4.3 - shared phenomenon x between *EU-Rent* and *EU-Rent Airport Centre*), into a message edge x with a ‘send’ and ‘receive’ task and two sequence edges (e.g., Figure 4.7 - Labelled with (10)).

Rule Nr.	B-SCP Concept	BPMN Concept
1	Top Node (in Requirement Diagram)	BPMN Diagram
2	Domain of Interest (in Context Diagram)	Pool Start Event Sequence Event End Event
3	Medium Node (in Requirement Diagram)	SubProcess
4	Leaf Node (in Requirement Diagram)	Task
5	OR Decomposition - first occurrence	2 x Gateway Data-Based Exclusive Activity 2 x Sequence Edge
6	OR Decomposition - other occurrences	2 x Sequence Edge
7	AND Decomposition (Sequence)	Sequence Edge
8	AND Decomposition (Parallel) - first occurrence	2 x Gateway Parallel Activity 2 x Sequence Edge
9	AND Decomposition (Parallel) - other occurrences	2 x Sequence Edge
10	Shared Phenomenon	‘Send’ Task (1st pool) Sequence Edge (1st pool) ‘Receive’ Task (2nd pool) Sequence Edge (2nd pool) Messaging Edge (from ‘Send’ to ‘Receive’ Task)

Table 4.1: BSCP2BPMN Concept Mappings



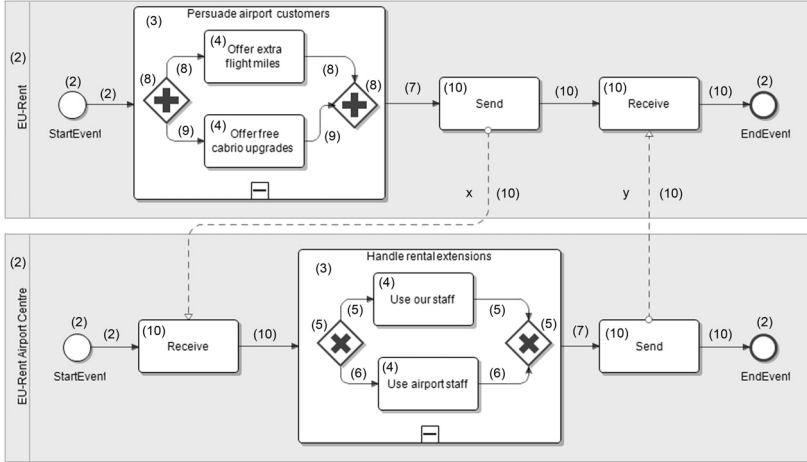


Figure 4.7: Resulting BPMN diagram for the B-SCP tactic ‘Encourage rental extensions’

## 4.4 Demonstration of our Approach

In this section, we will use the SEJ case exemplar to illustrate how our solution properties support a business user like a business process manager/owner/consultant in designing complex business processes that correspond to strategic requirements. Section 4.4.1 introduces the SEJ case exemplar. Section 4.4.2 demonstrates what our first solution property (implemented by the B-SCP metamodel and B-SCP editor) looks like in the context of SEJ. Section 4.4.3 demonstrates the second solution property (implemented by the control flow annotations), and finally, section 4.4.4 demonstrates the third solution property (implemented by the BSCP2BPMN model transformations) in the context of SEJ.

### 4.4.1 The Seven Eleven Japan (SEJ) Case Exemplar

Seven-Eleven Japan (SEJ) manages a national network of convenience stores and has successfully established an innovative business model that is changing the retail industry in Japan. Several case studies [NW04, BUMN97, Kun97, Li03] describe the information-based strategies that have helped SEJ become a top performing retailer in Japan, selling high quality products through an industry-wide supply chain network of manufacturers, distributors, third-party logistics providers and franchise shops.

On the *strategic* level, SEJ selected three key principles. First, a missed opportunity to sell an item because it is sold out is believed to represent up to three times the value of the actually realized profit. Hence, SEJ has to accurately determine when, where, in which quantities and at which price these products are needed by the customer. A second principle is the supply of products to the convenience stores just-in-time and in the quantity required, thereby eliminating slow selling items and replacing them by the faster selling ones. Third, the franchise system, in which the franchisee is an independent business, gives

SEJ large royalties and a long-term commitment, and concentrates on the tasks of selling and managing his inventory. In exchange, the franchiser provides the information capabilities, implements efficient operational systems to support planning and delivery of products, negotiates with the suppliers, advertises on a national scale, and develops new products to satisfy changing customer needs.

On the *operational* level, the strategic principles need to be supported by an adequate supply chain organisation and information system and human support. Firstly, to support the just-in-time delivery system, SEJ developed the Combined Delivery System, whereby the same kind of products coming from different suppliers can be centralized in a Combined Delivery Centre (CDC). Secondly, the connection between franchiser and franchisee was partly operationalized via the exchange of information. The franchisee sends information about local sales and customer behaviour to SEJ, enabling SEJ to track nationwide sales and to generate forecasts for subsequent deliveries to individual stores.

On the *technology* level, the information system is based on two main computer systems, i.e., the franchisee's store computer and the SEJ host computer. To begin with, the function of the franchisee's store computer is to load data from auxiliary hardware, to send that original data to the SEJ host computer, and to receive processed data from the SEJ host computer. Auxiliary hardware includes: the Point of Sale system, that reads bar codes on purchases to automatically enter sales in the system and that collects consumer profile data; the Graphical Order Terminal (GOT), that enables easy product ordering for employees and confirm product display orders and out-of-stock items; and the Scanner Terminal, that is used for data input when merchandise is delivered to stores and to change the order screens of the store computer and GOT to the sequence in which items are arranged on shelves. Next, the SEJ host computer receives all POS data and ordering data from the franchisee store computer, and develops fine grained predictive models of consumer purchase behaviour. As a result, distribution data are sent back to the franchisee's store computer, and to vendors and manufacturers.

#### 4.4.2 First Solution Property in Context of SEJ

The first solution property considers the *strategic rationale of having certain business processes*, and *having them organised in certain ways*. To start with, the *strategic rationale of having certain business processes* is demonstrated by the BSCP diagram containing problem diagrams A, B, C and D (Figure 4.8). Problem diagram A contains only strategic requirements, problem diagram B contains a mixture of strategic rationale and operational tactics, while problem diagrams C and D solely consist of tactics (i.e., business process information). As given by Figure 4.8, the strategic rationale of problem diagram C is contained in problem diagram B (objective O7 *Collect consumer data*), and the strategic rationale of problem diagram D is specified by problem diagram A (objective O2 *Provide decision support to stores*).

Then, *having business processes organised in certain ways* relates to the alternative choices that are given by the strategic rationale. For

instance, the objective O3 *Coordinate supply chain participants* is decomposed into objective O4 *Control store inventory real-time*, leading to objective O7 *Collect consumer data* that requires the consumer age and gender during consumer checkout. Any alternative choice to support objectives O3, O5 or O7 could result in a different organisation of the consumer checkout business process. In the following sub-sections, we will further explain the details of problem diagrams A, B, C and D.

#### 4.4.2.1 Domain diagram DA and requirement diagram RA

On top of the BSCP diagram we find problem diagram A. Context diagram DA consists of domains of interest *SEJ*, *Consumer*, *Franchise Store (FS)*, *Combined Delivery Centre (CDC)*, and *Supplier*. These domains of interest have shared phenomena on their interfaces (cfr. Table 4.2): (a) *Franchise stores* provide products for purchase that *consumers* want when they want them; (b) *SEJ* provides IT support enabling *Franchise Store* to maximize the use of limited resources to meet consumer demand; (c) the *Combined Delivery Centre* coordinates the delivery of products to *Franchise Stores*; (d) *SEJ* shares information with the *Combined Delivery Centre* about the expected deliveries; (e) the *Combined Delivery Centre* pick-ups the products of the *Suppliers*, and the *Suppliers* organise ex-factory delivery to the *Combined Delivery Centre*; and finally (f) *SEJ* informs the *Suppliers* about the expected product orders.

The requirements in RA reference and constrain the domains of interest in DA. *SEJ* promises each owner of a *Franchise Store* (bb, see Table 4.2) that it will enable the store to reduce lost opportunities (goal G1), minimize unsold perishables (G2), maximise the usage of the available floor space (G3), shorten the inventory runs (G4), maintain the freshness of perishables (G5), in order to offer customers what they want when they want it (G6). To this end, *SEJ* needs to coordinate the just-in-time delivery of stock to *Franchise Stores* (objective O1), which constrains *SEJ*, *Combined Delivery Centres* and the *Suppliers* (cc, dd in Table 4.2). Finally, the just-in-time coordination requires that *SEJ* provides decision support to the *Franchise Stores* (O2, bb in Table 4.2).

#### 4.4.2.2 Domain diagram DB and requirement diagram RB

Problem diagram A (containing RA-DA), which describes what *SEJ* intends to achieve as a business, can be decomposed into more refined problem diagrams, like problem diagram B, which refines the objective ‘deliver stock to FS just-in-time’ (O1). The context diagram DB consists of domains of interest *Clerk*, *Handheld Scanner*, *SEJ Host Computer*, *Combined Delivery Centre* and *Supplier*. When we investigate the shared phenomena between these domains of interest, we see that the *SEJ Host Computer* is responsible for sharing logistics service requests with the *Suppliers* (j), and for sharing information about product orders with the *Franchise Store Computer* (i). The *Handheld Scanner* is used by the *Clerk* for product shipment reception and scanning (g), and exchanges information with the *Franchise Store Computer* about consumer profile, purchase and store inventory data (h).

When looking at the requirements diagram RB, the need to send purchase orders to the *Suppliers* (T(actic) 1) and to send a service re-

quest to *Combined Delivery Centres* (T2) refer to the *SEJ Host Computer* (jj, kk in Table 4.2). The product shipment has to be scanned (T3) and the scanned data needs to be checked (T4), so this constraints the *Clerk* (gg, Table 4.2). Finally, the *Franchise Store Computer* is referred (ii, Table 4.2) by the requirement to coordinate supply chain participants (O3), to control store inventory real-time (O4), to update inventory in real-time (O5), and to regularly update the *SEJ Host Computer* (O6).

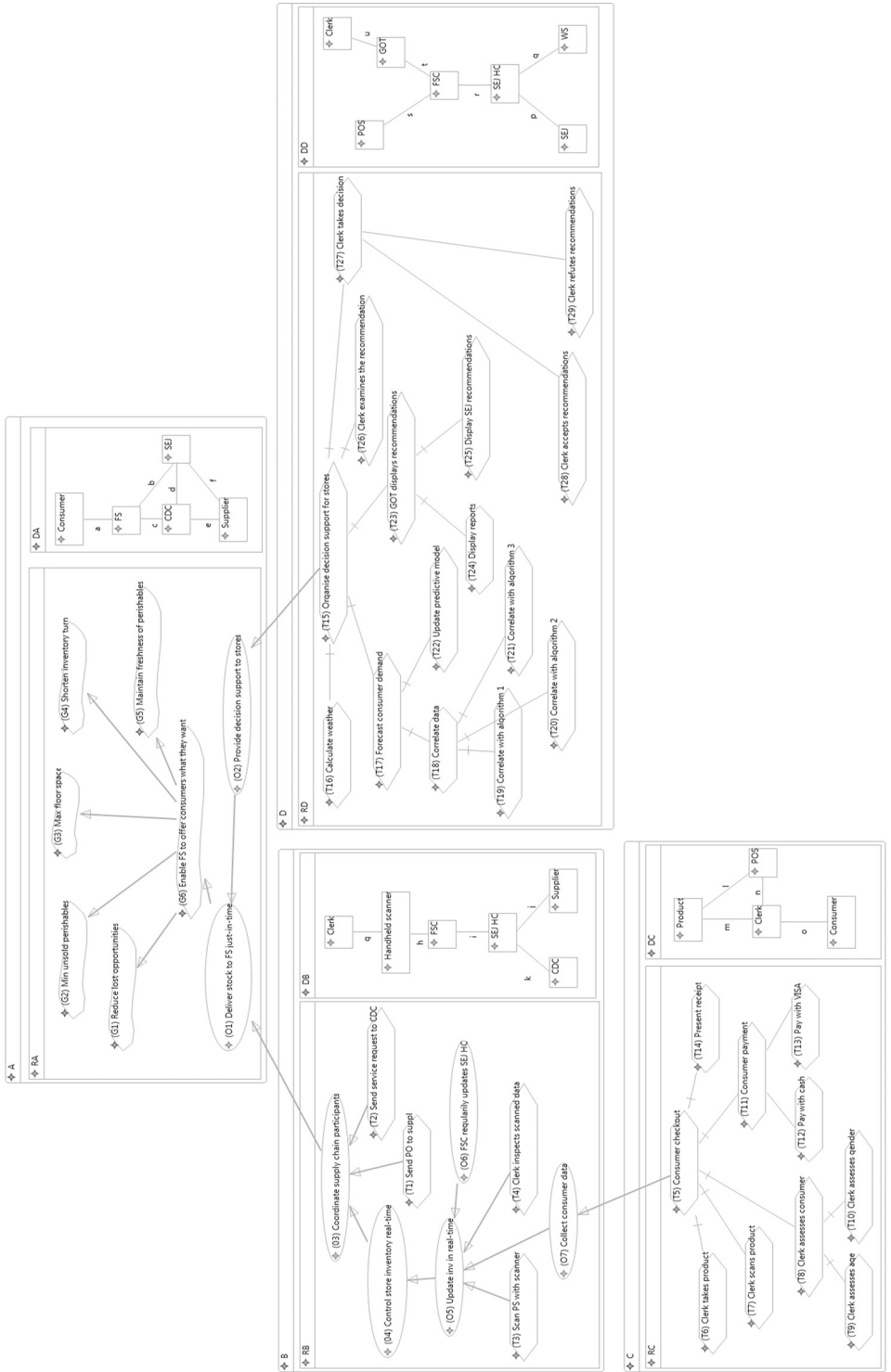


Figure 4.8: BSCP diagram for SEJ

	Interfaces		Requirements		
Do mai n Cont ext	Responsible DoI	Shared Phenomena	Requirements Set	Responsible DoI	Requirements
DA	a: Franchise Store !	{Provision of products for purchase that consumers want when they want them}	RA	aa: Franchise Store !	{G6}
	b: SEJ !	{IT support enabling franchise stores to maximize use of limited resources to meet consumer demand}		bb: SEJ !	{G1- G6, O2}
	c: Combined Delivery Center !	{Delivery of product to franchise stores}		cc: SEJ !	{O1}
	d: SEJ !	{Logistics coordination}		cc: Suppliers !	{O1}
	e: Combined Delivery Center !	{Pick-up of products from suppliers}		cc: Combined Delivery Center !	{O1}
	e: Suppliers !	{Ex-factory delivery}		dd: SEJ !	{O1}
	f: SEJ !	{Product Order}		dd: Supplier !	{O1}
DB	h: Handheld Scanner !	{Consumer profile, purchase, and store inventory data collection}	RB	gg: Clerk !	{T3, T4}
	j: SEJ Host Computer !	{Logistics services request}		ii: Franchise Store Computer !	{O3 – O6}
	i: SEJ Host Computer !	{Product order}		jj: SEJ Host Computer !	{T1}
	h: Franchise Store Computer !	{Stock monitoring}		kk: SEJ Host Computer !	{T2}
	g: Clerk !	{Product shipment reception scanning}			
DC	o: Clerk!	{Payment request}	RC	oo: Consumer!	{T5,T6}
	o: Consumer!	{Payment acknowledgment}		nn: Clerk!	{T1, T2, T3, T4, T7}
DD	s: POS !	{Consumer purchase and profile data}	RD	pp: SEJ!	{T15, T17}
	r: FSC !	{Consumer purchase and profile data}		qq: WS!	{T16}
	r: SEJ Host Computer !	{Stock order recommendation}		rr: SEJ Host Computer !	{T18 – T22}
	r: FSC !	{Updates in stock}		tt: GOT!	{T23 – T25}
	q : SEJ Host Computer !	{Interesting locations and times}		uu: Clerk!	{T26 - 29}
	q : WS !	{Weather data}			
	t : GOT !	{Query reports and recommendation}			
	t : FSC !	{Reports and recommendation}			
	t : GOT !	{Updates in stock}			
	u : Clerk !	{Query report and recommendation}			
	u : GOT !	{Updates in stock}			
	u : Clerk !	{Report data}			

*Table 4.2: Phenomena of the SEJ problem description*

#### 4.4.2.3 Domain diagram DC and Requirement diagram RC

An essential step in understanding what consumers want when they want them (G6), is the collection of consumer data (O7). To this end, the *Clerk* uses the *Point of Sale (POS)* system during consumer check-out to enter the necessary information. When we focus on the *Point of Sale* system, the context diagram DC consists of domains of interest *Consumer*, *Clerk* and *Product*. The consumer checkout (T5) can be expressed in terms of specific subtasks. The *Clerk* takes the *Product* (T6), the *Clerk* scans the *Product* (T7), the *Clerk* assesses the *Consumer* (T8), the *Consumer* pays the requested amount of money to the *Clerk* (T11), and the *Clerk* gives the product and receipt to *Consumer* (T14). During the assessment of the *Consumer* by the *Clerk* (T8), the *Clerk* tries to estimate age (T9) and gender (T10) in the same moment. In contrast, when the *Consumer* settles the payment (T11), he has to pay with cash (T12) or pay with a VISA card (T13), not both at the same moment. When we have a look at the shared phenomena of DC, we see that the *Clerk* is responsible for the payment request, and the *Consumer* needs to acknowledge the payment (o).

#### 4.4.2.4 Domain diagram DD and requirement diagram RD

The just-in-time delivery of stock to *Franchise Stores* (O1) depends on the provision of decision support to these *Franchise Stores* (O2). When we focus on the *SEJ Host Computer*, the context diagram DD consists of domains of interest *SEJ*, *Weather Service (WS)*, *Point of Sale (POS)*, *Franchise Store Computer (FSC)*, *Graphical Order Terminal (GOT)*, and the *Clerk*. The organisation of decision support for *Franchise Stores* (T15) can be expressed by means of specific subtasks. First of all, the *Weather Service* calculates the weather (T16), and *SEJ* forecasts the consumer demand (T17) by letting *SEJ Host Computer* correlate all data based on different algorithms (T18 - T21) and by updating the predictive model in the *SEJ Host Computer* (T22). Then, the *Graphical Order Terminal* displays recommendations (T23) by plotting graphical reports (T24) and showing textual recommendations (T25). When the *Clerk* examined the recommendations (T26), the *Clerk* takes a decision to accept (T27) or to refute (T29) these recommendations. Among these domains of interest, different shared phenomena can be witnessed. The *Point of Sales* system shares consumer purchase and profile data with the *Franchise Store Computer* (s). The *Franchise Store Computer* sends the consumer purchase and profile data to the *SEJ Host Computer*, along with updates in stock, and the *SEJ Host Computer* responds to the *Franchise Store Computer* by delivering stock order recommendations (r). The *SEJ Host Computer* shares information about interesting locations and times with the *Weather Service*, and the *Weather Service* provides weather data to the *SEJ Host Computer* (q). The *Graphical Order Terminal* is used to query reports, query recommendations, and to provide updates in stock to the *Franchise Store Computer*, after which the *Franchise Store Computer* provides the *Graphical Order Terminal* with the requested reports and recommendations (t). Finally, the *Clerk* uses the *Graphical Order Terminal* to execute the queries for reports and recommendations and to update the stock, and the *Graphical Or-*

der Terminal provides the report data (u).

### 4.4.3 Second Solution Property in Context of SEJ

The second solution property supports business users to *describe business processes in familiar terms*. This property relates to an appropriate level of modelling abstraction given to business users, in which B-SCP starts with high abstraction levels at the top, and incrementally allows business users to refine the desired models, until the lowest-level of abstraction is reached (and control flow annotations become relevant). At this lowest-level of abstraction, business users can *describe business processes in familiar terms* as they only have to work with OMG's BMM concept tactic, which they can decompose into other tactics as many times as they want. For instance, problem diagram D (Figure 4.8) contains fifteen tactics, of which our approach will consider one tactic as a business process (T15), four tactics as subprocesses (T17, T18, T23, T27), and ten tactics as activities (T16, T19, T20-T22, T24-T26, T28, T29). So after applying our BSCP2BPMN model transformations, Figure 4.9 displays the resulting BPMN diagram corresponding to tactic T15, in which the subprocesses and activities are embedded in their corresponding BPMN lanes (e.g., tactic T17 *Forecast consumer demand* relates to domain of interest *SEJ HC*, so the BPMN lane *SEJ HC* contains subprocess T17 *Forecast consumer demand*). The messaging flows between the different lanes in Figure 4.9 correspond to the shared phenomena contained in Table 4.2, which specify one main flow that is initiated by the *Point of Sales* to *Send consumer purchase and profile data* (via interface s) to the *Franchise Store Computer*, and another main flow is initiated by the clerk to *Send query reports and recommendations* (via interface u) to the *Graphical Order Terminal*.

### 4.4.4 Third Solution Property in Context of SEJ

The third solution property concerns the *explicit linkage of business processes and strategic requirements*, as Figure 4.8 integrates the strategic requirements with business process T5 *Consumer checkout* (represented by problem diagram C), and business process T15 *Organise decision support for stores* (represented by problem diagram D). The resulting BPMN business process diagram (Figure 4.9) does not contain any strategic requirements, as BPMN Level 2 [Sil09] requires the business analyst to focus on BPMN details such as events, gateways and exceptions. In the following sub-sections, we will further explain the details of the BPMN models resulting from problem diagrams C and D.

#### 4.4.4.1 BPMN Model Resulting From Problem Diagram C

When problem diagram C is transformed into a BPMN diagram, the top node of the requirement diagram (T5 *Consumer Checkout*) indicates the business process in scope. The resulting BPMN model is given by Figure 4.9. The four domains of interest in DC are transformed into four pools (Figure 4.9 omits the *POS* pool and *Product* pool as these have no corresponding requirements). In these pools, the medium nodes (T8, T11) are presented by BPMN subprocesses and leaf nodes (T6, T7, T14)



are modelled by means of BPMN tasks. Most control flow annotations are sequential AND decompositions, except the parallel AND decomposition between T9 and T10 (Figure 4.9 - *Clerk assesses age* and *Clerk assesses gender* with two Parallel Gateways) and the OR decomposition between T12 and T13 (Figure 4.9 - *Pay with cash* and *Pay with VISA* with two Data-Based Exclusive Gateways). The shared phenomena on interface *o* between *Clerk* and *Consumer* are presented by means of BPMN message edges and Send and Receive Tasks.

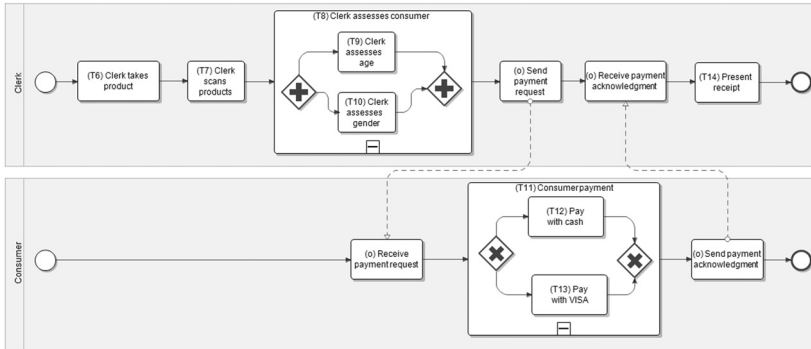


Figure 4.9: BPMN Model Resulting From Problem Diagram C

#### 4.4.4.2 BPMN Model Resulting From Problem Diagram D

In problem diagram D, the top node in requirement diagram RD (T15 *SEJ organises decision support for stores*) defines the business process in scope. The resulting BPMN model is given by Figure 4.10. From the seven domains of interest in DD, corresponding pools and events are generated (Figure 4.10 omits the *SEJ* pool as the only corresponding requirement T15 has already been resolved to the BPMN process itself). In order to represent the complex structure of shared phenomena in DD, several message flow edges with corresponding Send and Receive tasks are added to Figure 4.10. The *Point of Sale* pool sends consumer purchase and profile data to the *Franchise Store Computer* pool, which is forwarded to the *SEJ Host Computer* pool. The *SEJ Host Computer* then requests and receives weather data from the *Weather Service* pool, such that the consumer demand can be forecasted. The Forecast Consumer Demand subprocess at the *SEJ Host Computer* pool correlates the gathered data by means of one or more algorithms at the same time (subprocess *HC correlates data* with parallel AND decomposition) and updates the predictive model. Based on these insights, the *SEJ Host Computer* generates stock order recommendations and sends the updates in required stock to the *Franchise Store Computer* pool. Back in a *Franchise Store*, the *Clerk* can query reports and recommendations by means of the *Graphical Order Terminal*, which are forwarded to and answered by the *Franchise Store Computer*. Then the *Graphical Order Terminal* displays reports and *SEJ* recommendations (subprocess *GOT generates recommendations* with sequential AND decomposition) and makes the *Clerk* aware of new updates in stock. When the *Clerk* examines the recommendations, he/she accepts or refutes these recom-

mendations (subprocess *Clerk* takes decision with OR decomposition). The *Clerk* uses the *Graphical Order Terminal* to confirm his decision, which sends these updates to the *Franchise Store Computer* and to the *SEJ Host Computer*.

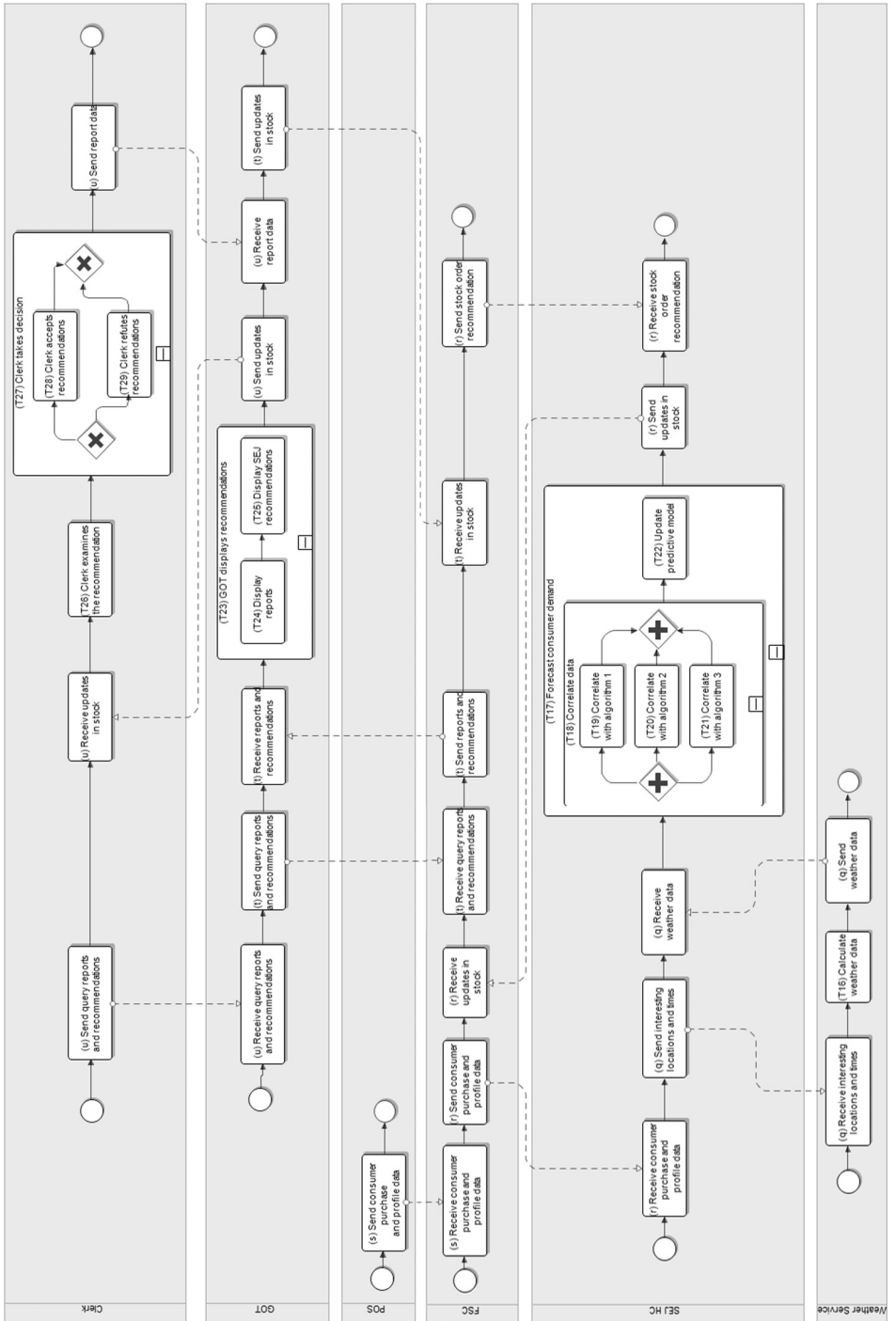


Figure 4.10: BPMN Model Resulting From Problem Diagram D

## 4.5 Evaluation studies

In this section, we will report on our case study research, conducted to understand what added value is experienced by business users of our approach, compared to a situation where business processes are directly modeled using BPMN. Section 4.5.1 gives an overview of the case study design. Section 4.5.2 introduces the strategic requirements and business processes in scope during the two case studies. Section 4.5.3 presents the case study findings that relate to the added value that can be attributed to the first solution property. Section 4.5.4 does the same for the second and third solution properties.

### 4.5.1 Case Study Design

A small, but representative, number of business experts (i.e., what Silver [Sil09] would call business users of BPMN) in both involved companies were asked to employ a graphical editor to create business process models in a scenario with real-life complexity, which we define as a scenario that would result in one or more business process diagrams with *over two hundred* nodes and relations in total. The scenarios employed also required participants to reconsider and design existing business process models in the selected companies. The first company is one of the world's major players in logistics and shipping, delivering to over 230 countries worldwide. The second company is one of the leading global suppliers of rail equipment and services, with 58 production and engineering sites as well as 20 service centres in 25 countries.

In the first case study, two business experts (financial planning managers) were offered the B-SCP editor that we developed. The B-SCP diagram created was then transformed into BPMN business process diagrams using the BSCP2BPMN model transformations that we developed. After the transformation, a business analyst (functional analyst at the IT department) was asked to complete the diagrams. The role of the business analyst was needed to gain feedback about the BPMN diagrams that resulted from the BSCP2BPMN model transformations. In the second case study, participants (a senior business process manager and a functional analyst) used the *itp-commerce Process Modeller for Microsoft Visio* editor, which is recommended by Silver [Sil09] for business process modelling by business users (i.e., at BPMN Level 1). These participants were thus asked to directly create BPMN business process diagrams.

#### 4.5.1.1 Measures

Several conceptual frameworks have been proposed to help understanding quality in the context of conceptual modelling [MP07, Moo05]. Bandara et al. [SRG02, SRD03, SGRS04, BGR05, BR05, BGR06] developed a process modelling success model using both qualitative and quantitative research methods. Three success factors (i.e., modelling aids, project-specific factors and modeller expertise) were found that explain process modelling success, which is measured by means of four outcome measures (i.e., model quality, user satisfaction, process impacts and project efficiency). The success factor that our method can contribute

to is *modelling aids*, which can be defined for our purposes as the combination of the (existing) B-SCP framework, the (new) graphical B-SCP editor, the (adapted) control flow annotations, and the (new) mapping rules for the BSCP2BPMN model transformation. These modelling aids aim at better supporting business users of BPMN (i.e., at BPMN Level 1), so we wish to find out what the effect of the support we developed is on the outcome of business process modelling, which we measure in terms of *model quality* and *user satisfaction*. We selected these two outcome measures, and excluded process impacts and project efficiency as outcome measures, because we are interested in the underlying reasons *why* our approach is adding value, being less concerned with quantifying the exact extent to which our approach adds value.

We will use the following measures [SRG02] to structure our data collection and data analysis. Firstly, *model quality* refers to the extent to which all essential features of a model are present, and contains six components: (a) *semantic correctness* measures how well the models describe the real-world, (b) *syntactic correctness* measures whether the models correctly instantiate the metamodel that defines the modelling language (i.e., in our case, whether the business process diagrams are correct BPMN models), (c) *economic efficiency* measures how feasible models are in terms of how easy it was to construct these models, (d) *clarity* measures how understandable the models are, (e) *comparability* measures how comparable the models are to one another, and (f) *systemic design* measures the degree to which models allow integrating information from different views. Secondly, *user satisfaction* refers to the feelings users have towards the derived models, and contains four components [DT88]: (a) *content* measures the information needs and the information that was captured to represent the underlying information, (b) *accuracy* measures the degree to which models are correct, exact and without any mistakes, (c) *format* measures the usefulness of the output format, and (d) *ease of use* measures the degree to which the modelling is user friendly.

Although *semantic correctness* and *content* look related, they differ in terms of stakeholders. The measure *semantic correctness* investigates the relation between a model and the real-world setting in an organisation that is represented by that model (including all stakeholders, corporate culture, corporate strategy, and operational priorities), while the measure *content* deals with the need for information as perceived by the users that are involved in the modelling (which does not necessarily align with the corporate culture or corporate strategy).

#### 4.5.1.2 Unit of Analysis

Eisenhardt [Eis89,EG07] recommends that the selection of cases should focus on theoretically useful cases (theoretical sampling), i.e., those that replicate or extend theory by filling conceptual categories. For our case study, we will look at business process-centred organisations that are present in the logistics and supply chain sector because this sector is known for its very strong and explicit focus on business processes [MJ00]. We targeted two different companies in the logistics sector, one in which we applied our approach (*literal replication*) and another in which we applied the process modelling approach for business

users of BPMN recommended by Silver [Sil09] (*theoretical replication*).

### 4.5.1.3 Data Collection and Analysis

We will triangulate [Eis89] four sources of evidence to collect our case study data [Yin09], i.e., *documentation*, *direct observation*, *participant-observation*, and *interviews*. In order to collect the necessary data, multiple rounds of data gathering were needed. The first round of data collection consisted of studying existing *documentation* (such as Powerpoint files) and identifying interesting business processes with a sufficient level of real-world complexity. In the second round, *direct observation* (taking a passive role as observer) and *interviewing* was used during the creation of the deliverables by the business users. The third round consisted of *participant-observation* (combining a passive observation role with possible active involvements) during the model transformation from B-SCP to BPMN to guarantee that the business analysts interpret and complete the generated business process skeletons in a way that is consistent to Section 4.3.4. Finally, we took *interviews* with both the business users and the business analysts, in order to receive more feedback on the modelling process and the model quality of the resulting models.

The collected data need to be analyzed, interpreted, and summarized in function of the research question. To this end, we employed the qualitative analysis methods of Miles and Huberman [MH94]. For each of the two cases, a checklist matrix was used to analyse model quality and another checklist matrix to measure user satisfaction (Appendix B - Figure B.1 provides the data format of this checklist matrix).

## 4.5.2 Case Studies

### 4.5.2.1 Case Study 1

In the first company, two perspectives on the current business were chosen, i.e., modelling the daily business and modelling the current initiatives to reduce cost. The resulting B-SCP model consists of approximately 175 nodes and 150 relationships, and can be found at [Dec10].

The daily business deals with helping customers to send a package, and to process the invoices to settle customer payments. Typically, a customer calls the call centre and determines the priority of the shipment. When the priority is normal, the customer drops off the package at an access point and the package is dispatched via several hubs to reach the delivery destination. When the priority is high, the dispatcher reviews the fastest shipment options and makes sure that the package is personally delivered in the shortest time possible. The other daily routine is to process invoices from customers, which starts by manually investigating the invoice, trying to find an appropriate manager to review the invoice, and then manually reconcile the invoices. In doing this reconciliation, often problems appear which could escalate to management levels.

The current initiatives to reduce costs have a severe impact on the organisation. The CEO wants to decrease the costs, which obliges the CFO to reduce the financial expenses without losing quality of service and without touching the core processes. To this end, three major goals

need to be achieved: the Shared Service Centre should reduce the unit price per activity, the Finance department should reduce the volume of needed Shared Service Centre activities, and Finance should have better insights in their spending. These goals are supported by several projects, such as the reorganisation of the procurement process, asking customers to do self-billing, offshore the generation of outgoing invoices, and outsource the printing and sending of these invoices.

#### 4.5.2.2 Case Study 2

In the second company, we selected the business processes related to the conformity and the procurement of products. The resulting BPMN model consists of approximately 190 nodes and 105 relationships, and can be found at [Dec10].

When the company receives a train order from a customer, it applies the *internal processes* to build such a train and to deliver the train to the customer. For this purpose, the product needs to be designed, *realized* and introduced into the customers' setting. The realization phase consists of looking for strategies to build, selecting technologies, scheduling manufacturing and control, developing specific methods and tools, the procurement of required materials, *controlling the product conformity* and managing the cost data. To control the product conformity, the conformity requirements and corresponding control plan are defined. When the product is not conform with these requirements, a chain reaction is started to identify the non-conformity, *document the non-conformity*, decide on the repair process, and repair the non-conformity. In order to document the non-conformity, quality control needs to open software screen XYZ, to fill out all the required fields, and to confirm the input.

To give another example, during the 'manage materials' process, a 'request for material' can result in the need to *procure* an external product. To this end, engineering will provide a 'technical specification', the project manager will determine the budget, a shortlist of possible suppliers is made, a 'request for quotation' is sent to all the shortlisted suppliers, a supplier is chosen, the material is ordered, which finally results in receiving and accepting the material. During this procurement process, departments Engineering and Procurement execute the core tasks of the company, that is to "deliver the right product, at the right price, on the right time", and other departments such as IT or Finance are in support of these core processes. Put differently, the entire company relies on the quality of its (sub-)products and timely delivery in respect of the timing in which these products are needed in manufacturing ("just in time"). The cost at which the product is purchased stands in function of the desired quality and timing; for instance, it might be cheaper for manufacturing to rent a helicopter to deliver a product 'on time' instead of waiting for classic delivery that causes a delay.

#### 4.5.3 Findings for First Solution Property

The first solution property considers the strategic rationale of having certain business processes, and having them organised in certain ways. In order to understand why this property is of added value to a business

user, we selected the measures that gave the most conflicting findings between the literal replication (using our approach) and the theoretical replication (using BPMN Level 1). As a result, we selected measures *semantic correctness* (indicating added value for model quality) and *content* (indicating added value for user satisfaction). The findings for other measures converged, so for these measures no indications were found that our approach would score better than BPMN Level 1.

#### 4.5.3.1 Semantic Correctness

Semantic correctness measures how well the models describe the real world. At the first company, the real-world situation was strongly goal-oriented, as the business users wanted to reduce the unit price per activity and to reduce the volume of activities. All case study material was structured to support any of these goals, such as implementation of procurement software, organisational change projects, and outsourcing initiatives. The B-SCP model, based on OMG's BMM semantics, was very appropriate to describe this kind of real-world situation. For instance, the strategic context (CEO, Finance, Operations, Customer) relates to the strategic goal *CEO wants to reduce financial expenses*, while the operational context of the procurement project (Supplier, Procurement, Operations, Marketing, Finance) relates to the tactics that are needed to execute the project (e.g., create purchase order, manager reviews purchase order, and supplier processes order). Furthermore, the original documentation of the business users did not contain any explicit control flow links, but loosely structured blocks such as requisition, approval, purchase order, deliver, etc.

At the second company, the real-world situation was also strongly goal-oriented, which was reflected by the company's mission: deliver the right product, at the right price, on the right time. These strategic requirements are important, especially in the field of train equipment, as these goods are highly capital intensive, require a good planning and should comply to strict quality demands. In order to achieve these strategic requirements, the company's business processes are organised in an optimal way to support these strategic requirements. For instance, the company uses just-in-time principles to manage its supply, has strong control mechanisms to prevent delay, and employs several full-time equivalents to follow up on the timing of suppliers. We observed that the BPMN models did not capture this strategic business process context, and only documented the execution of the business processes as understood by the business users.

#### 4.5.3.2 Content

The content measures the information needs and the information that was captured to represent the underlying information. At the first company, the information needs were the documentation of organisational change and to understand the interactions between programmes, projects, and departments. The B-SCP model represented all strategic requirements and operational business processes, so the business users were positive about the content of the resulting B-SCP models. Furthermore, the documentation that we used contained the objectives of



each project and process, as well as the clear overview of the roles and responsibilities on each company level.

At the second company, the need for information is mostly related to company organisation, scheduling and quality control. As a result, the business user was positive about the BPMN models as they reflected the way that the business organises its way of working, but stressed that these BPMN models should not be intended for technical developers (due to a lack of detail). The business user explained: “it’s like business people talk in Spanish, the technical people talk in French, which causes communication by means of body gestures”. A few years ago, the decision was made to implement a SAP information system, which struggled strongly in supporting the strategic requirements of the company, as SAP lacks a detailed planning module and offers insufficient support for quality management. Having strategic requirements linked to their business process models was perceived as very useful, as this would allow the technical developers to understand the priorities of the business users, instead of having technical developers hinder the work of the business users for technical reasons.

#### 4.5.4 Findings for Second and Third Solution Properties

The second solution property supports business users to describe business processes in familiar terms, and the third property is the explicit linking of business processes and strategic requirements. Analogue to the previous section, we selected the measures that gave the most conflicting findings between the *literal replication* (using our approach) and the *theoretical replication* (using BPMN Level 1) in the context of this property. As a result, we selected the measures *systemic design* (indicating added value for model quality) and *format* (indicating added value for user satisfaction). The other measures did not clearly indicate differences between the two approaches.

##### 4.5.4.1 Systemic Design

Systemic design measures the degree to which models allow integration of information from different views (including different abstraction levels). At the first company, the final B-SCP diagram included ten problem diagrams which ranged from high-level, strategic ones to low-level, implementation-oriented ones. We found good support of our approach for Moody’s principle of complexity management, as B-SCP offers explicit support for modularization and hierarchical structuring. For instance, the final B-SCP diagram shows the overview of the different modules that support the strategic requirements, such as problem diagrams with daily procedures (Handle shipments, Normal invoice processing), problem diagrams that represent projects (such as the project that improves the daily procurement process), and problem diagrams that represent outsourced companies (such as the outsourcing of invoice printing). Furthermore, the *i\** decomposition relations support Moody’s hierarchical structuring, which offers the business users visual support to understand the hierarchy of the B-SCP model.

At the second company, the business user was challenged in understanding the hierarchical relations of all BPMN models. The BPMN modelling tool that was used, allows users to collapse or expand by means of right-clicking, but this causes a sudden jump through the business process hierarchy, which makes it difficult to reposition the users' thoughts. For instance, when the business user finished the BPMN model *Schedule manufacturing and control process*, the collapsing of that BPMN model took the business user back to the high-level BPMN model *Processes for realization of products*. This high-level BPMN model is only one step higher in the business process hierarchy, while the Visio editor represented the BPMN model in a totally other location. Furthermore, the business users were challenged to insert or change models in the middle of the hierarchy, because it was unclear what the effects of the changes in one BPMN model are in relation to other BPMN models in the process hierarchy. This is due to redundancy in BPMN modelling of Silver [Sil09], which adds black-box pools for external participants on different levels, and requires the modeller to remember the connections between these black-box pools. For instance, when the black-box pool Customer was changed to Project Manager in one BPMN model, this change had to be done in five related BPMN models, which had to be discovered by visual inspection of all BPMN models.

#### 4.5.4.2 Format

The format measures the usefulness of the output format. At the first company, the B-SCP model was found very useful, especially as the discussion of the strategic requirements and business processes took place on a different abstraction level than the discussion about the control flow annotations. The modelling sessions were not only perceived as business process modelling, but as modelling the insights of the business users. For instance, one business user referred to the B-SCP model as 'a mindmap', which is a diagram used to represent ideas arranged around a central theme. The lack of typical BPMN Level 1 concepts, such as message flows, control flows, events and gateways, was left unnoticed, as the business users were not focused on the exact control flow details of the business processes.

At the second company, the BPMN format was less useful to express the company's business process hierarchy. The company distinguished five levels of process detail, i.e., process (level 1), subprocess (level 2), activity (level 3), task (level 4), action / transaction (level 5). Although these five process levels have clearly distinct semantics, the available BPMN symbols (subprocess and activity) are quite similar, and no symbol for *process* exists in BPMN (that is, a top-level BPMN diagram represents a *process*, but different top-level BPMN diagrams cannot be related to each other). Furthermore, the visual representation of the gateway and event symbols was not always clear. For instance, the visual difference between the exclusive and parallel gateways is very small, which confused the business user to understand the resulting control flow in the BPMN diagrams.

## 4.6 Conclusion

Our work addresses the issues that arise when providing business users with the same BPMN modelling aids as business analysts and technical developers. Although we support the intent of the BPMN specification to bridge the gap between business and IT, our case-study research shows that business users need more explicit support for BPMN modelling than offering a BPMN language that is common to business users and more IT-oriented users. As a solution we present a goal modelling based approach that allows business users to design complex business processes in terms of and in correspondence with strategic requirements (answering RQ2.2). Further on, we demonstrated the feasibility of our approach by means of the SEJ case exemplar (answering RQ3.3). Then, we motivate the solution properties of this approach by the following claims (answering RQ3.4):

- The consideration of the strategic rationale of having certain business processes, and having them organized in certain ways, is important for a business user of business process modelling methods. Previous research has shown that business users are highly goal-oriented and are used to consider the strategic rationale in organisations [LL90,KN08]. Our case study findings confirm this, and suggest that using our approach leads to models with high *semantic correctness* and that the information needs of business users are satisfied by the *content* that our approach tries to model.
- Support for describing business processes in terms familiar to business users and being able to explicitly link these business processes to strategic requirements, is important for a business user of business process modelling methods. Moody's principle of complexity management [Moo09] recommends to have explicit mechanisms for dealing with complexity during modelling. As our approach offers explicit support for complexity management, our case study findings are aligned with these theoretic insights. More specifically, we found that our approach improves the *systemic design* of an organization's business processes and that our approach offers a useful *format* to business users.

An underlying motivation of our research is to encourage the technology transfer between requirements engineering research and industry by lowering the barriers of current GORE research for real-world business users [KBBJ<sup>+</sup>02]. As a result, our approach tries to find a balance between *rigour* and *relevance* [HMPR04]. Firstly, the restrictions of the B-SCP metamodel and graphical B-SCP editor forces a certain degree of *rigour* in our approach, as the business user must understand and comply to our tool-supported method. Secondly, our approach offers *relevance* to business users by not overloading the business user with technical semantics or formal business rules, as B-SCP is based on strategic management techniques [Son99] and OMG's general accepted Business Motivation Model (BMM) [OMG09b].

Originally, B-SCP was proposed as a requirements engineering framework for validating strategic alignment of organisational IT, based on

manual interpretation of traceability links between strategy and technology. In this chapter, we reused the work on B-SCP for a different purpose, that is as modelling aid for BPMN Level 1, which we interpret as the business process modelling level where business users design complex business processes in terms of and in correspondence with strategic requirements. We believe that B-SCP offers a well-documented and scalable alternative to the currently available modelling methods that combine strategic goals and business processes, which are often  $i^*$ -based modelling languages (see chapter 2). Few published studies exist on applying the  $i^*$  goal language into practice, and indications exist that practitioners of large-scale industrial projects are unable to understand  $i^*$  models well enough to validate the requirements of the system they were building [MJM<sup>+</sup>04]. As the B-SCP framework was proposed to address the known shortcomings of  $i^*$  and to leverage the existing knowledge of Jackson's Problem Frames, we considered the B-SCP framework as the starting point of our work.

The differentiation between a goal-oriented requirements language and a business process language is the result of a deliberate design choice. As a modelling language is always conceived with a certain purpose in mind [My198], we believe it is easier to represent goals and business processes using different languages, and to provide model-based translations between these languages, instead of choosing one modelling language to represent both goals and business process concepts. With low modelling complexity (e.g., modelling a clearly understood business process), creating a requirements model could be seen as an overhead cost. But, as real-world business process modelling projects often quickly grow in terms of complexity, business users can use a requirements model as an overview (or one could say, an overarching strategically aligned business process architecture), and generate the required business process models from the requirements model.

The limitations of our approach are the absence of the reverse transformation (from BPMN to B-SCP) and the lack of larger-scale quantitative research study following-up on our case-study research. Firstly, our work presents a top-down modelling method, which enables business users to transform parts of B-SCP models into BPMN business process diagram skeletons, but the reverse transformation (from BPMN to B-SCP) is currently not supported. To this end, our future work will investigate how the PRiM method of Grau et al. [GFM08] could be reused to generate B-SCP requirements from BPMN diagrams. Secondly, a large-scale follow-up study is now needed to see whether the current results are representative for a larger population of business users. Our future work will use the business research methods of Cooper and Schindler [CS03] to organise a quantitative follow-up study.

# 5

## Conclusions

In the last decade, the Requirements Engineering (RE) and Business Process Management (BPM) research fields have slowly but steadily grown towards each other. The overall objective of our PhD research was to further bridge the gap between RE and BPM by focusing on the common notion of *goal* in both areas of research. As business processes are meant to fulfill through their execution certain organizational goals, it is worth investigating whether goal models that document and analyse organizational strategies, could play an active role in the design of business processes. Our research focused on Goal-Oriented Requirements Engineering for Business Processes (GORE for BP) methods that support business users to design business processes based on goals and other strategic requirements.

In the first part of this dissertation, the *problem analysis*, we wanted to acquire knowledge about existing GORE for BP methods. In the second part of this dissertation, the *solution design*, we aimed to improve the currently proposed GORE for BP methods to address the problems with respect to their use that we discovered in the first part. In the third part of the dissertation, the *solution validation*, we wanted to understand the feasibility and added value of the GORE for BP methods that we improved during solution design.

As the three parts of this PhD dissertation were written in parallel, the different parts interacted with each other as given by Figure 5.1. Initially, the work of chapter 3 was started, which provided first insights in the problem analysis of chapter 2. Later on, the research of chapter 3 was finalized, and a final draft was created for chapter 2. Based on the insights of chapter 2 and chapter 3, the work of chapter 4 was started. Finally, after finishing chapter 4, the final revision of chapter 2 was made.

As a result of the interactions between the different chapters, we structured this PhD by means of research questions as defined by Figure 1.7 (in Chapter 1). In this chapter, we will briefly summarize our contributions made in answering these research questions (Section 5.1), stress the implications of our research for practitioners and for researchers

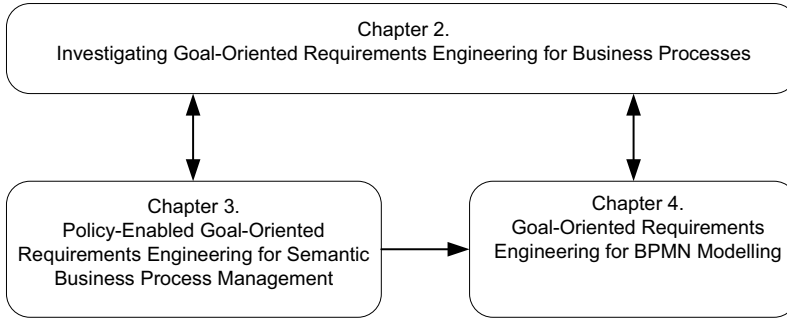


Figure 5.1: Interactions between chapters

(Section 5.2), and discuss the limitations and future work (Section 5.3).

## 5.1 Answering the Research Questions

- **RQ1.1 What is the unifying framework of a GORE for BP method?**

We discovered that the unifying framework of GORE for BP methods is constituted by seven states and their transitions, i.e. the Elicited Requirements state, the Specified Requirements state, the Validated Requirements state, the Process Designed state, the System Configured state, the Process Enacted state, and the Process Diagnosed state (Figure 2.2, in Chapter 2). Based on these seven states, we extended Kavakli's [KL99a, Kav02] unified framework for GORE methods for deriving a unified framework for GORE for BP methods, that allows us to model and compare individual GORE for BP methods.

- **RQ1.2 Which are the currently available GORE for BP methods that are sufficiently *complete* in terms of documentation, to allow for comparison with other methods and investigation of strengths and weaknesses?**

We conducted a systematic literature review (based on the guidelines of Kitchenham [Kit04]) in which GORE for BP methods were identified, considering all possible types of goal models and business process models. Typically, a GORE for BP method started by capturing high-level strategic requirements and by transforming them into low-level business process details (top-down), but we also included methods that start from business process details and extract the higher-level goals from these business process details. While performing the systematic literature review, we discovered large differences in completeness of method descriptions, ranging from vague descriptions to detailed user manuals. As we were interested in comparing these GORE for BP methods, we needed to reduce the subjective interpretation of the identified methods as much as possible. As a result, we identified seven GORE for BP methods that are sufficiently *complete* in terms of documentation to enable an objective evaluation. This list of seven methods includes Aburub et al. [AOB07, CdPLdMSN01], Bleistein et al.

[BCVP06a,BACR04,BCV04c,BCV04a,BCV06,BCVP06b,BCV04b,BCV05,CPBV05,CP03], Frankova et al. [FMS07b,FYS07,FMS07a,SHF08,GMMZ06,MMZ08], Grau et al. [GFM05,GFA06,GFM08,JM04], Koliadis et al. [KVB<sup>+</sup>06b,KVB<sup>+</sup>06a,FLM<sup>+</sup>04], Lapouchnian et al. [LL06a,LYM07,SGM04], and Soffer and Rolland [SR05,RPB99].

- **RQ1.3 What are the similarities and differences between these *complete* GORE for BP methods in relationship to the unifying framework addressed by RQ1.1?**

Once we understood the unifying framework of a GORE for BP method, and once we identified the currently existing GORE for BP methods that are sufficiently complete in terms of documentation, we investigated the similarities and differences of these GORE for BP methods. The phenomena of this knowledge problem were the GORE for BP methods that have been identified by research question RQ1.2. The variables we introduced to investigate these phenomena are given by the following list:

- Completeness in terms of REBPM lifecycle
- Kind of conceptual modelling languages
- Validation in practical context
- Applicability restrictions
- Practical guidelines offered to modeller
- Maturity of transformation activity

We discovered that GORE for BP methods cover most of the typical RE lifecycle but are quite incomplete with respect to their coverage of the typical BPM lifecycle. So, GORE for BP methods generally focus on how to *specify* requirements and *design* processes, and how these requirements specifications and process designs can be *mapped* onto each other. Less attention is given to *elicit* or *validate* such requirements/designs, and little focus is given to the implementation of process designs by means of a BPMS system. Furthermore, we observed that no ways-of-working feed back into the RE states Elicited Requirements, Specified Requirements, Validated Requirements once the System Configured state is reached. As a result, the connection from process design to high-level requirements is lost, which might obstruct a correct diagnosis of an enacted process. For instance, the business process ‘Manage Account Receivables and Account Payables’ could put pressure on the payment terms of the suppliers, in order to achieve the goal ‘Shorten Cash Cycle’. Once parts of the business process ‘Manage Account Receivables and Account Payables’ are implemented by means of an BPMS system, the diagnosis of the enacted processes might benefit from understanding its strategic importance (which is to support the business goal ‘Shorten Cash Cycle’). Current GORE for BP methods fall short of providing backward traceability from process execution results to the goal-oriented business process requirements.

When we compare the GORE for BP methods in terms of the other variables, we discover little common ground in the different conceptual modelling languages used by GORE for BP methods. An exception to this are the different *i*\*-variants (used by Bleistein et al, Frankova et al, Grau et al, Koliadis et al, and Lapouchnian et al), which are comparable in terms of visual syntax, but differ in terms of semantics. Next, most GORE for BP methods have been illustrated by means of an example, but lack further validation in practical context. Related to the lack of detailed validation, is the low priority for offering practical guidelines to the modeller, which might block industry adoption of GORE for BP methods. Furthermore, the maturity of the transformation activity is overall sufficient, but again misses concrete details about concept mappings or about practical guidelines for step-by-step transformation. Finally, the majority of the GORE for BP methods were created for generic business settings with typical strategic requirements and operational business processes, but some GORE for BP methods were restricted for application to secure business processes, non-functional requirements for business processes and high-variability processes.

- **RQ2.1 How can business users design semantic business processes in terms of and in correspondence with strategic requirements and policies?**

We proposed an approach to model policies as part of the requirements for business processes and to generate business process design skeletons that respond to these requirements, and hence to take the formulated policies into account. The differentiation between a requirement language and a business process language is based upon the belief that it adds value for a domain expert to create a requirement model in order to generate a business process design skeleton instead of the direct creation of business process models. In particular with respect to policy modelling and compliance management, our working hypothesis is that it is useful and valuable to first model new or changed policies in a requirement model and next to generate business process design skeletons out of this requirement model. Hence, the changes to business process designs, needed to comply with the new/changed policies, can be done more easily/effectively, compared to directly changing existing business process models.

In order to allow business users to model requirements in a specific, yet intuitive way, we needed to have an easy-to-use graphical tool and an accompanying method to build the requirements space. Our first contribution is taking an existing early requirements specification language, i.e. Formal Tropos [FLM<sup>+</sup>04], and extending this language to incorporate policies, called Policy-extended Formal Tropos. Furthermore, we created a graphical editor to allow business users to visually develop requirement models and generate Policy-extended Formal Tropos instances from this. Our second contribution is offering model transformations to create business process design skeletons (using Business Process Mod-



elling Ontology) out of the Policy-extended Formal Tropos models.

- **RQ2.2 How can business users design complex business processes in terms of and in correspondence with strategic requirements?**

We proposed a new BPMN approach to business process modelling targeted specifically at business users (i.e., BPMN level 1 as referred to by Silver [Sil09]). In our approach, the main solution properties are:

- consideration of the strategic rationale of having certain business processes, and having them organized in certain ways
- support for describing business processes in terms familiar to business users
- linking these business processes to strategic requirements

This approach relies on previous Goal-Oriented Requirements Engineering (GORE) research, such as the B-SCP framework of Bleistein et al. [BCVP06b] and the work of Lapouchnian et al. [LYM07]. In [DP10] we elaborated on the basis of these methods an initial outline of a *GORE for modelling with BPMN* approach. The first contribution in Chapter 4 was the introduction of the completed approach with all implementation details. We adopted the B-SCP framework and operationalized it for our purposes (instead of using B-SCP for business-IT alignment purposes) by creating a *B-SCP metamodel* to define the abstract syntax of the B-SCP language and a *B-SCP editor* for the visual creation of problem diagrams. Our approach thus offers an explicit serialization of the B-SCP model via the B-SCP editor and provides a tool-supported business process modelling method that is adapted to the needs of business users. In addition, we implemented the *control flow annotations* of Lapouchnian et al. [LYM07] in our B-SCP editor, and constructed *model transformations* from B-SCP (modelling the strategic requirements to which business process must correspond) to BPMN (modelling the business processes that correspond to strategic requirements) to support the business analyst in digesting the output of the business user.

The properties of our approach are supported by our implementation as follows. The consideration of the strategic rationale of having certain business processes, and having them organized in certain ways, is given by the new *B-SCP metamodel* and corresponding *B-SCP editor* that we developed to support business users in the visual creation of problem diagrams. The support for describing business processes in terms familiar to business users is given by the *adoption of the B-SCP framework* for goal modelling as well as the business priorities driven *annotation of control flow* in problem diagrams. The explicit linkage of business processes with strategic requirements is ensured by the *BSCP2BPMN transformation* of problem diagrams into business process diagrams.

- **RQ3.1 Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.1?**

We conducted three iterations of a pilot study that we used as empirical test of our first approach, and based on the lessons learned from each iteration, we adjusted the necessary elements in the subsequent iteration. Initially, the feasibility of our approach was challenged by problems related to the policy vocabulary we used, the lack of explicit method step descriptions, the demand for example-based introduction and a short preparation phase. After tackling each of these issues in the last pilot study iterations, we did not find any further problems related to the feasibility of our approach.

- **RQ3.2 What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.1?**

The pilot studies discussed in RQ3.1 also provided insights in the added value of our approach. We experienced that regular business process modelling (i.e., without our approach) resulted in semantically overloaded models. For instance, activities in the business process model were used to represent real activities performed in the course of a business process, but also to represent the verification of policies. Using Policy-extended Formal Tropos, no semantically overloaded concepts were found. Next, the overall feedback of the participants, when comparing the models they created with our method and the ‘standard’ approach and discussing their experiences, pointed at the potential value of separately modelling business processes and the policies that these process must adhere to. In their daily work, the participants focus on control-flow aspects of the business process, which frequently coincide with the ‘best case’ scenario of the business process. As a result of the pilot study, they discovered how they implicitly implemented policy activities in their daily work. For instance, several telephone calls and emails are sent to remind actors in the process to speed up their work (as the flight ordering should be done within two days), although this is not represented in the regular business process model. Overall, the user feedback was positive on separate thinking about business processes and about policies, and having tool support to link both concepts.

- **RQ3.3 Can we demonstrate the feasibility of the approach that incorporates the answer to RQ2.2?**

We demonstrated the feasibility of our approach by means of applying our approach to the SEJ case exemplar [NW04, BUMN97, Kun97, Li03]. More specifically, we used the specifics of SEJ to illustrate how the B-SCP metamodel and B-SCP editor contribute to consider the strategic rationale of having certain business processes (and having them organized in certain ways) and how the control flow annotations and BSCP2BPMN model transformations support the description of business processes in terms familiar to business users (and linking these business processes to strategic requirements). All details of this demonstration can be found in section 4.4 (Chapter 4).

- **RQ3.4 What is the added value experienced by business users when they use the approach that incorporates the answer to RQ2.2?**

We conducted two case studies based on the principles of Yin [Yin09], Eisenhardt [Eis89] and Miles and Huberman [MH94]. Firstly, we discovered that the consideration of the strategic rationale of having certain business processes, and having them organized in certain ways, is important for a business user of business process modelling methods. Previous research has shown that business users are highly goal-oriented and are used to consider the strategic rationale in organisations [LL90,KN08]. Our case study findings confirm this, and suggest that using our approach leads to models with high *semantic correctness* and that the information needs of business users are satisfied by the *content* that our approach tries to model. Secondly, we found that support for describing business processes in terms familiar to business users and being able to explicitly link these business processes to strategic requirements, is important for a business user of business process modelling methods. Moody's principle of complexity management [Moo09] recommends to have explicit mechanisms for dealing with complexity during modelling. As our approach offers explicit support for complexity management, our case study findings are aligned with these theoretic insights. More specifically, we found that our approach improves the *systemic design* of an organization's business processes and that our approach offers a useful *format* to business users.

## 5.2 Implications

Following Smith and Fingar [SF04], we consider BPM as a tool-supported management approach for the design, manufacture and maintenance of business processes. This vision on BPM shows a direct path from process design to a software system that implements the process. It does not relate to 'rapid application development', it simply *removes* application development from the business cycle. BPM is then a platform for sharing end-to-end business processes analogous to the use of a database management system as a platform for sharing business data, both between applications and among business partners. As such, BPM becomes the platform upon which the next generation of business applications will be constructed.

In this context, Silver [Sil09] explains the importance of BPMN 2.0 as it provides a common language for describing process behaviour, shareable by business and IT. Silver [Sil09] distinguishes different types of BPMN modelling, depending on the user category. Firstly, BPMN Level 1, or *descriptive* modelling, is geared towards the business user and offers a basic set of BPMN elements. Secondly, BPMN Level 2, or *analytical* modelling, supports the business analyst in using the complete BPMN notation to describe the activity flow precisely, including the exception paths. These models should be complete and consistent, but not yet contain technical details to make them executable. Thirdly, BPMN Level 3, or *executable* modelling, allows the technical developers

to add process data, service interfaces and human task assignment that are needed to execute the BPMN models using BPMS technology.

### 5.2.1 Implications for Practitioners

The work of Smith and Fingar [SF04] and Silver [Sil09] summarizes how business users are supposed to deal with business processes in BPMS software. From this perspective, the *assumption* is taken that *business users do not need to understand the requirements of business processes*. We do not agree with this assumption, and motivate the necessity of offering explicit support to business users for designing business processes based on requirements.

In the field of RE, it is well-known that business analysts and IT developers should express their requirements during the software development process [vL09]. So, when a BPMS allows business users to actively participate in the software development process, why should these business users be left out of the requirements engineering process? The importance of requirements engineering support for business users was made clear in 1967, when Ackoff [Ack67] wrote that most software engineers determine what information is needed by *asking* managers what information they would like to have, which is based on the incorrect assumption that *managers know what information they need*. While written in 1967, Ackoff's [Ack67] paper remains one of the most-cited papers up until today in the field of Information Systems [WL00], indicating the present value of his statements.



Figure 5.2: *Scylla and Charybdis* [Jac00]

One of the leading researchers in RE, Michael Jackson, explains that problems in software development have some precise and some imprecise aspects, all competing for our attention [Jac00]. Like Odysseus on his ship coming home from Troy, we are sailing between Scylla and Charybdis, and must try to steer a middle course (Figure 5.2). If we are attracted by the arguments of people who consider software problems as too formal and narrow, we may be dragged off to the right, into the world of purely human problems. This is an imprecise world of sociology and ethnography, where nothing is ever completely certain or completely exact. However, if we find that problems are less exciting than their software solutions, we are more likely to veer off to the left, towards the much more precise world of programming. This is a precise world of bits and bytes, where boolean values are always True or False, and every value between True or False is treated as useless information.

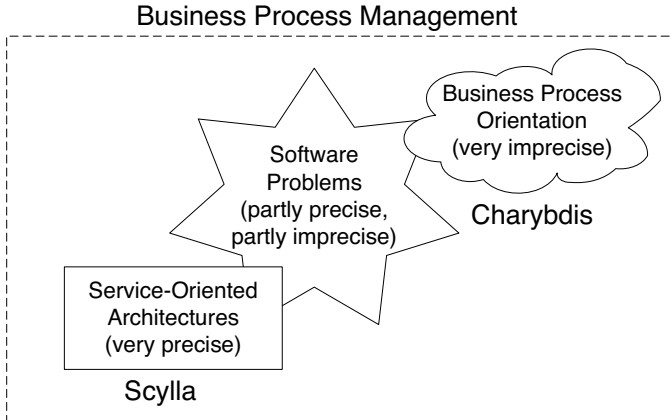


Figure 5.3: Positioning BPM to Scylla and Charybdis

When we apply Jackson’s ideas to the current view on BPM [SF04, Sil09], we obtain Figure 5.3. This view tries to unify the precise and imprecise aspects of business processes, by combining programming with Web Services and Service-Oriented Architectures (very precise) and sociologic aspects of business process orientation (very imprecise). The business user is to be situated in the field of sociology, the business analyst has to solve the software problems and the IT developer is responsible for programming with Service-Oriented Architectures.

As Figure 5.3 shows, the current view on BPM does not eliminate the danger of Scylla and Charybdis; in contrast, it brings Scylla and Charybdis very close to each other. This is a dangerous undertaking, as experienced by James Martin’s *Information Engineering* [Mar89] in the 1980s. Information Engineering was defined as translating a strategic plan into an Information Systems Architecture that can be transformed into data, software applications and geographic architectures [HK88, Kar88]. Back then, the revolutionary proposition of Information Engineering was that *information requirements* steer the development of *business process models*, and that *business process models* steer the development of *software applications*. Although the current RE community still accepts this proposition [RSB05], James Martin’s Information Engineering approach quickly lost popularity in the 1990s and is barely known in the current world of business and research. Based on a conversation with Basil Kritis, a former colleague of James Martin, we discovered that Information Engineering was ahead of its time, and that there was an enormous resistance of the IT developers that were made redundant by means of the Information Engineering platform. More specifically, the IT developers believed that Information Engineering “killed the art of programming”, which is in effect a *social* reason that hinder the introduction of a *software* development approach. More general, when we look at the business-IT alignment literature [CR07], it is well known that intellectual, structural, social and cultural reasons hinder the cooperation of business users and more technically-oriented employees.

Overall, we want to stress the importance for practitioners of *align-*

ing the (strategic) requirements of a business user with the (technical) requirements of the business analysts and IT developers. When all business process models are centralized in a common BPMS repository, the changes of one user might directly impact the interests of another user (e.g., what is the impact on the company's strategic goals when an IT developer changes the implementation of a business process?). This dissertation offers business users two tool-supported methods to design business processes based on strategic requirements, which generates graphical traces of the requirements of business processes. The usage of these methods has been demonstrated and validated in Chapter 3 and Chapter 4. All tool support is open source and freely available at <http://www.managementinformation.ugent.be/PFT2BPMS/> (first approach) or <http://code.google.com/p/bscp2bpms/> (second approach).

## 5.2.2 Implications for Researchers

During the problem analysis, we investigated the currently existing GORE for BP methods, and discovered a total of 19 different methods. Firstly, we believe that GORE for BP research should redirect attention from the requirements specification activity to further investigate the requirements elicitation and validation activities. Secondly, we would like to recommend researchers investigating GORE for BP to abstract their work from the individual goal and business process modelling languages, such that the methodological aspects of their work get separated from syntax aspects and a common methodological ground for GORE for BP can emerge. Thirdly, we would like to motivate future GORE for BP method researchers to support the business user in a more explicit way by providing clear and practical guidelines. Fourthly, we would like to motivate current and future research on GORE for BP to focus on the transformation activity, and reuse work from related research areas such as model driven development or model-to-model transformations.

Overall, the research in GORE for BP methods is fragmented. In terms of the conceptual framework of Wieringa and Heerkens [WH06], most researchers change the state of the world by introducing new GORE for BP artefacts, without motivating which *world problem* they are trying to solve, or without thoroughly validating the degree to which GORE for BP artefacts solves their world problem. The same phenomenon occurs at the RE conferences, where “authors find it less fun to do the research than to invent the techniques” ([WH06], p305). It seems that the GORE for BP research is driven by the urge to express personal creativity, and is less driven by the relevance to other researchers or businesses. We would like to recommend GORE for BP researchers to focus on solving *knowledge problems* (instead of *world problems*), to gain insights in assumptions, properties, and effects of GORE for BP methods. These insights might then inspire GORE for BP researchers to solve *shared and recognized problems*, which could help to diminish the fragmentation in GORE for BP research.

We would like to end this section by means of a critical discussion about the role of the *business user* (or more generally, the *end user*). Since the 1960s, the role of the end user in software development was often limited to express some requirements in the beginning of the soft-

ware development project, and to test the resulting software applications for bug fixing. Recently, we are experiencing a stronger focus on the end user aspects during software development, with the upcoming of *agile requirements engineering* [RCB07, CR08] and *end-user development* [FG04, PRdRW09]. First, agile requirements engineering [RCB07, CR08] advocates moving into coding without a centralized requirements analysis phase, and letting the requirements of the system emerge throughout the development process, while heavily relying on feedback from the customer.

Second, *end-user development* [FG04, PRdRW09] describes techniques that allow people who are not professional developers to create or modify a software artefact. The basic assumption is that domain experts, who see software development as a means to an end, will design tools and create contents of a higher quality than professional software designers, for whom software is both a means and an ends. If the tool created by the developer does not satisfy the needs or the tastes of the user (who knows best), then the user should be able to adapt the system without always requiring the assistance of the developer. For instance, when an end user wants to install a kitchen in his house, the user lacks the expertise to build the kitchen, so traditionally a kitchen-making company listens to this user's kitchen requirements and delivers the custom-made kitchen. Nowadays, IKEA offers an end-user development approach to install kitchens, allowing the end user to use IKEA modelling software to design his dream kitchen with drag-and-drop 3D functionality, after which the end user can visit an IKEA store to buy all needed prefabricated kitchen components and install the kitchen by himself.

This dissertation offers business users tool-supported methods to design business processes based on strategic requirements. We consider the *business user* as a domain expert who knows best the business processes (but with little experience in process design), and the *business analyst* as an expert in process design (but with little knowledge about the real-world execution of business processes). From this perspective, our work offers an end-user development approach that allows business users to express requirements of business processes, and -without requiring the business user to have process design expertise- generates business process skeletons out of these business process requirements. As BPMN 2.0 and BPMS technology assumes an active participation of business users, we consider the trend of end-user development in BPM of paramount importance. As Gerhard Fisher explained in his keynote speech of the 2nd International Symposium End-User Development [PRdRW09], end users will be empowered to create software artefacts without the help of IT experts, just like literacy empowered the common people to write text without the help of monks. The empowerment of the end user in BPM is confirmed by researchers at SAP [SSFM08, SSS08, SSFM09], which is the current world leader in business applications with more than one billion users worldwide.

### 5.3 Limitations and Future Work

The underlying motivation of our research is to encourage the technology transfer between RE research and industry by lowering the barriers of current GORE for BP research for real-world business users [KBBJ<sup>+</sup>02]. As a result, this dissertation tries to find a balance between *rigour* and *relevance* [HMPR04]. So from the point of view of *rigour*, we can further formalize our metamodels, constraints, semantics, and model transformations, and from the point of view of *relevance*, we can still improve the guidelines and methods for the business user. For instance, other common constructs of goal models could still be exploited, such as obstacles, anti-goals or goal mitigation. Furthermore, we used qualitative analysis during our solution validation, as we were interested in *why* business users would benefit from designing business processes based on strategic requirements. In order to extrapolate these findings into a more general setting, a large-scale follow-up study is now needed to see whether the current results are representative for a larger population of business users. To this end, the business research methods of Cooper and Schindler [CS03] might be used to organise large-scale quantitative research.

This dissertation presents top-down modelling methods, which enable business users to transform requirements into business process designs, while the reverse transformation is currently not supported. This reverse transformation, extracting requirements from business process designs, is an important step during GORE for BP methods. For instance, the business process ‘Manage Account Receivables and Account Payables’ could put pressure on the payment terms of the suppliers, in order to achieve the goal ‘Shorten Cash Cycle’. Once parts of the business process ‘Manage Account Receivables and Account Payables’ are implemented by means of an BPMS system, the diagnosis of the enacted processes might benefit from understanding its strategic importance (which is to support the business goal ‘Shorten Cash Cycle’). During our problem analysis, we discovered that the PRiM method of Grau et al. [GFM08] is one of the best GORE for BP methods that extract requirements from business process design. Future work might investigate the PRiM method [GFM08] to complement the work that we presented during our solution design.



# References

- [ACC<sup>+</sup>05] Claudia P. Ayala, Carlos Cares, Juan P. Carvallo, Gemma Grau, Mariela Haya, Guadalupe Salazar, Xavier Franch, Enric Mayol, and Carmen Quer. A Comparative Analysis of i\*-Based Agent-Oriented Modeling Languages. In *International Workshop on Agent-Oriented Software Development Methodology - SEKE Conference, Taipei, Taiwan*, 2005.
- [Ack67] Russel L. Ackoff. Management Misinformation Systems. *Management Science*, 14(4):147–156, 1967.
- [AEC03] Annie Anton, Julia B. Earp, and Ryan A. Carter. Precluding incongruous behavior by aligning software requirements with security and privacy policies. *Information and Software Technology*, 45(14):967–977, 2003.
- [AEH<sup>+</sup>04] Annie Anton, Julia B. Earp, Qingfeng He, William Stufflebeam, Davide Bolchini, and Carlos Jensen. Financial Privacy Policies and the Need for Standardization. *IEEE Security and Privacy*, 2(2):36–45, 2004.
- [Amy03] Daniel Amyot. Introduction to the User Requirements Notation: learning by example. *Computer Networks*, 42(3):285–301, 2003.
- [AOB07] Faisal Aburub, Mohammed Odeh, and Ian Beeson. Modelling non-functional requirements of business processes. *Information and Software Technology*, 49(11-12):1162–1171, 2007.
- [AP03] Marcus Alanen and Ivan Porres. A Relation Between Context-Free Grammars and Meta Object Facility Metamodels. *TUCS Technical Report No 606, ISSN = 1239-1891*, March, 2003.
- [BA05] Travis D. Breaux and Annie Anton. Analyzing Goal Semantics for Rights, Permissions, and Obligations. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering*. IEEE Computer Society, 2005. 177-188.
- [BA08] Travis Breaux and Annie Anton. Analyzing Regulatory Rules for Privacy and Security Requirements. *IEEE Transactions on Software Engineering*, 34(1):5–20, 2008.

- [BACR04] S. J. Bleistein, A. Aurum, K. Cox, and P. Ray. Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach. *Journal of Research and Practice in Information Technology*, 36:259–276, 2004.
- [BAD08] Travis D. Breaux, Annie Anton, and Jon Doyle. Semantic parameterization: A process for modeling domain descriptions. *ACM Transactions on Software Engineering and Methodology*, 18(2):1–27, 2008.
- [BAS09] Travis D. Breaux, Annie. Anton, and Eugene H. Spafford. A distributed requirements management framework for legal compliance and accountability. *Computers and Security*, 28(1-2):8–17, 2009.
- [BCV04a] S. J. Bleistein, K. Cox, and J. Verner. Problem frames approach for e-business systems. *IEEE Seminar Digests*, 2004(904):7–15, 2004.
- [BCV04b] Steven Bleistein, Karl Cox, and June Verner. Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagrams with Goal Modeling and BPM. In *Proceedings of the 11th Asia-Pacific Software Engineering Conference*. IEEE Computer Society, 2004. 410-417.
- [BCV04c] Steven J. Bleistein, Karl Cox, and June Verner. Modeling Business Strategy in E-Business Systems Requirements Engineering. In *Conceptual Modeling for Advanced Application Domains*, pages 617–628. 2004.
- [BCV05] Steven J. Bleistein, Karl Cox, and June Verner. Strategic alignment in requirements analysis for organizational IT: an integrated approach. In *Proceedings of the 2005 ACM symposium on Applied computing*, Santa Fe, New Mexico, 2005. ACM. 1300-1307.
- [BCV06] Steven J. Bleistein, Karl Cox, and June Verner. Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. *JSS*, 79(3):362–378, 2006.
- [BCVP06a] Steven Bleistein, Karl Cox, June Verner, and Keith Phalp. Requirements engineering for e-business advantage. *REJ*, 11(1):4–16, 2006.
- [BCVP06b] Steven J. Bleistein, Karl Cox, June Verner, and Keith T. Phalp. B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. *Information and Software Technology*, 48(9):846–868, 2006.

- [BGR05] Wasana Bandara, Guy G. Gable, and Michael Rosemann. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, 14(4):347–360, 2005.
- [BGR06] W. Bandara, G. Gable, and M. Rosemann. Business Process Modeling success: an empirically tested measurement model. In *ICIS*, Milwaukee, USA, 2006.
- [BKB<sup>+</sup>07] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571–583, 2007.
- [BLMR04] Arosha Bandara, Emil Lupu, Jonathan Moffett, and Alessandra Russo. A Goal-based Approach to Policy Refinement. In *Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, 2004.
- [BN04] Judith Barrios and Selmin Nurcan. Model Driven Architectures for Enterprise Information Systems. In *Advanced Information Systems Engineering*, pages 3–19. 2004.
- [BPS01] Janis Bubenko, Anne Persson, and Janis Stirna. Appendix B - EKD User Guide. *HyperKnowledge Deliverable IST-2000-28401*, 2001.
- [BR05] W. Bandara and M. Rosemann. What Are the Secrets of Successful Process Modeling? Insights From an Australian Case Study. *Systemes d'Information et Management*, 10(3):47–68, 2005.
- [Bri96] Sjaak Brinkkemper. Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 38:275–280, 1996.
- [BSH99] Sjaak Brinkkemper, Motoshi Saeki, and Frank Harmen. Meta-modelling based assembly techniques for situational method engineering. *Information Systems*, 24(3):209–228, 1999.
- [BUMN97] Ben M. Bensaou, H. Uchino, K. Mitani, and M. Noishiki. Seven-Eleven Japan: Managing a Networked Organisation. *INSEAD Euro-Asia Centre - Case Study 05/97-4690*, 1997.
- [BVA06] Travis D. Breaux, Matthew W. Vail, and Annie Anton. Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. In *14th IEEE International Requirements Engineering Conference*, pages 49–58. IEEE Computer Society, 2006.

- [CBRT06] K. Cox, S.J. Bleistein, P. Reynolds, and A. Thorogood. A contingency view of organizational infrastructure requirements engineering. In *ACM SAC OE, France*, 2006.
- [CdPLdMSN01] Luiz Marcio Cysneiros, Julio Cesar Sampaio do Prado Leite, and Jaime de Melo Sabat Neto. A Framework for Integrating Non-Functional Requirements into Conceptual Models. *Requirements Engineering*, 6(2):97–115, 2001.
- [CG08] Monique Calisti and Dominic Greenwood. Goal-Oriented Autonomic Process Modeling and Execution for Next Generation Networks. In *Modelling Autonomic Communications Environments*, pages 38–49. 2008.
- [Cib96] Claudio U. Ciborra. The Platform Organization: Recombining Strategies, Structures, and Surprises. *Organization Science*, 7(2):103–118, 1996.
- [CKM02] Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards requirements-driven information systems engineering: the Tropos project. *Information Systems*, 27(6):365–389, 2002.
- [CKO92] Bill Curtis, Marc I. Kellner, and Jim Over. Process modeling. *Communications of the ACM*, 35(9):75–90, 1992.
- [CP03] K. Cox and Keith Phalp. From process model to problem frame - a position paper. In *REFSQ*, 2003.
- [CPBV05] Karl Cox, Keith T. Phalp, Steven J. Bleistein, and June M. Verner. Deriving requirements from process models via the problem frames approach. *Information and Software Technology*, 47(5):319–337, 2005.
- [CR07] Yolande E. Chan and Blaize Horner Reich. IT alignment: what have we learned? *Journal of Information Technology*, 22(4):297–315, 2007.
- [CR08] Lan Cao and Balasubramaniam Ramesh. Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, 25(1):60–67, 2008.
- [CS03] D. Cooper and P. Schindler. *Business Research Methods*, volume 7th Edition. McGraw-Hill / Irwin, 2003.
- [Dav93] Tom Davenport. *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, 1993.
- [Dec10] Ken Decreus. Google Project Site - <http://code.google.com/p/bscp2bpnm/>, 2010.

- [DGR<sup>+</sup>06] Islay Davies, Peter Green, Michael Rosemann, Marta Indulska, and Stan Gallo. How do practitioners use conceptual modeling in practice? *Data and Knowledge Engineering*, 58(3):358–380, 2006.
- [DKPP09] Ken Decreus, M. El Kharbili, Geert Poels, and Elke Pulvermueller. Bridging Requirements Engineering and Business Process Management. In *Requirements Engineering for Business Process Management Workshop, at Software Engineering 2009*, Kaiserslautern, 2009.
- [DL06] Robert Darimont and Michel Lemoine. Goal-oriented Analysis of Regulations. In *International Workshop on Regulations Modelling and their Verification and Validation*, Luxemburg, 2006.
- [DLF93] Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.
- [dIVS07] Jose de la Vara and Juan Sanchez. Business process-driven requirements engineering: a goal-based approach. In *8th workshop on BPMDS*, 2007.
- [dIVS08] Jose Luis de la Vara and Juan Snchez. Improving Requirements Analysis through Business Process Modelling: A Participative Approach. In *Business Information Systems*, pages 165–176. 2008.
- [dIVSP08] Jose de la Vara, Juan Snchez, and scar Pastor. Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems. In *AISE*, pages 213–227. 2008.
- [DP09] Ken Decreus and Geert Poels. Mapping Semantically Enriched Formal Tropos to Business Process Models. *Proceedings of the ACM SAC 2009 (Hawaii) - Track RE*, 2009.
- [DP10] Ken Decreus and Geert Poels. A Goal-Oriented Requirements Engineering Method for Business Processes. In *CAiSE Forum*, Tunesia, 2010.
- [Dru88] Peter F. Drucker. Coming of the New Organization. *Harvard Business Review*, JanuaryFebruary:45–53, 1988.
- [DSP09] Ken Decreus, Monique Snoeck, and Geert Poels. Practical Challenges for Methods Transforming i\* Goal Models into Business Process Models. In *RE '09: Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE*, pages 15–23, Washington, DC, USA, 2009. IEEE Computer Society.

- [DT88] William J. Doll and Gholamreza Torkzadeh. The Measurement of End-User Computing Satisfaction. *MIS Quarterly*, 12(2):259–274, 1988.
- [Ecl10a] Eclipse. ATL - ATLAS Transformation Language - <http://www.eclipse.org/m2m/atl/>, 2010.
- [Ecl10b] Eclipse. EMF - Eclipse Modelling Framework - <http://www.eclipse.org/modeling/emf/>, 2010.
- [Ecl10c] Eclipse. GMF - Graphical Modeling Framework - <http://www.eclipse.org/modeling/gmf/>, 2010.
- [EG07] Kathleen M. Eisenhardt and Melissa E. Graebner. Theory Building From Cases: Opportunities and Challenges. *Academy of Management Journal*, 50(1):25–32, 2007.
- [Eis89] Kathleen M. Eisenhardt. Building Theories from Case Study Research. *The Academy of Management Review*, 14(4):532–550, 1989.
- [EKSMP08a] M. El Kharbili, S. Stein, I. Markovic, and E. Pulvermiller. Towards a Framework for Semantic Business Process Compliance Management. In *Shazia Sadiq, M. z. M. (ed.), Proceedings of the workshop on Governance, Risk and Compliance for Information Systems (GRCIS 2008), CEUR Workshop Proceedings, Montpellier, France, pages 1–15, 2008.*
- [EKSMP08b] M. El Kharbili, S. Stein, I. Markovic, and E. Pulvermiller. Towards Policy-Powered Semantic Enterprise Compliance Management. In *3rd International Workshop on Semantic Business Process Management (SBPM), CEUR Workshop Proceedings, 16-21., 2008.*
- [FFFVL97] Martin Feather, Stephen Fickas, Anthony Finkelstein, and Axel Van Lamsweerde. Requirements and Specification Exemplars. *Automated Software Engineering*, 4:419438, 1997.
- [FG04] Gerhard Fisher and Elisa Giaccardi. Meta-design: a framework for the future of end-user development. In H Lieberman, F Paterno, and Wulf V, editors, *End User Development - Empowering people to flexibly employ advanced ICT*. Kluwer Academic Publishers, 2004.
- [FLM<sup>+</sup>04] Ariel Fuxman, Lin Liu, John Mylopoulos, Marco Pistore, Marco Roveri, and Paolo Traverso. Specifying and analyzing early requirements in Tropos. *Requirements Engineering*, 9:132–150, 2004.
- [FMS07a] Ganna Frankova, Fabio Massacci, and Magali Seguran. From Early Requirements Analysis towards Secure Workflows (Short paper). In *IFIPTM*, 2007.

- [FMS07b] Ganna Frankova, Fabio Massacci, and Magali Seguran. Technical Report DIT-07-036 - From Early Requirements Analysis Towards Secure Workflows. *University of Trento*, 2007.
- [FPGGD<sup>+</sup>10] H. Fernandez Fernandez, Elias Palacios-Gonzalez, Vicente Garcia-Diaz, B. Cristina Pelayo G-Bustelo, Oscar Sanjuan Martinez, and Juan Manuel Cueva Lovelle. SBPMN - An easier business process modeling notation for business users. *Computer Standards and Interfaces*, 32(1-2):18–28, 2010.
- [Fux01] Ariel Fuxman. *Formal analysis of early requirements specifications (MSc thesis)*. PhD thesis, 2001.
- [FYS07] Ganna Frankova, Artsiom Yautsiukhin, and Magali Seguran. Technical Report DIT-07-037 - From Early Requirements to Business Processes With Service Level Agreements. *University of Trento*, 2007.
- [GAP07] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Towards a Framework for Tracking Legal Compliance in Healthcare. In *Advanced Information Systems Engineering*, pages 218–232. 2007.
- [GAP09] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Compliance Analysis Based on a Goal-oriented Requirement Language Evaluation Methodology. In *17th IEEE International Requirements Engineering Conference (RE09)*, pages 133–142, 2009.
- [GFA06] Gemma Grau, Xavier Franch, and Sebastian Avila. J-PRiM: A Java Tool for a Process Reengineering i\* Methodology. In *Proceedings of the 14th IEEE International Requirements Engineering Conference*. IEEE Computer Society, 2006. 352-353.
- [GFM05] G. Grau, X. Franch, and N.A.M. Maiden. A goal based round-trip method for system development. In *11th International Workshop on Requirements Engineering: Foundation For Software Quality (REFSQ)*, pages 71–86, 2005.
- [GFM08] Gemma Grau, Xavier Franch, and Neil A. M. Maiden. PRiM: An i\*-based process reengineering method for information systems specification. *Information and Software Technology*, 50(1-2):76–100, 2008.
- [Gla97] Robert L. Glass. Pilot studies: What, why and how. *Journal of Systems and Software*, 36(1):85–97, 1997. 0164-1212 doi: DOI: 10.1016/0164-1212(95)00197-2.
- [GMMZ06] Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Requirements engineering for trust management: model, methodology, and reasoning. *International Journal of Information Security*, 5(4):257–274, 2006.

- [GR07] Dominic Greenwood and Giovanni Rimassa. Autonomic Goal-Oriented Business Process Management. In *Proceedings of the Third International Conference on Autonomic and Autonomous Systems*. IEEE Computer Society, 2007.
- [Gre08] Dominic Greenwood. Goal-Oriented Autonomic Business Process Modeling and Execution: Engineering Change Management Demonstration. In *Business Process Management*, pages 390–393. 2008.
- [GSP<sup>+</sup>09] Sepideh Ghanavati, Alberto Siena, Anna Perini, Daniel Amyot, Liam Peyton, and Angelo Susi. A Legal Perspective on Business: Modeling the Impact of Law. In *E-Technologies: Innovation in an Open World*, pages 267–278. 2009.
- [Gui07] Giancarlo Guizzardi. On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In *Proceeding of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference*. IOS Press, 2007. 18-39.
- [Har91] H. James Harrington. *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness*. McGraw-Hill, 1991.
- [Hav06] Michael Havey. Keeping BPM simple for business users: power users beware. *BPTrends*, January, 2006.
- [HC94] Michael Hammer and James Champy. *Reengineering the corporation : a manifesto for business revolution*. HarperBusiness, New York, 1994.
- [HDK93] P. Hsia, A. Davis, and D. Kung. Status-Report - Requirements Engineering. *IEEE Software*, 10(6):75–79, 1993.
- [HK88] Richard D. Hackathorn and Jahangir Karimi. A Framework for Comparing Information Engineering Methods. *MIS Quarterly*, June:203–220, 1988.
- [HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105, 2004.
- [HR07] Martin Hepp and Dumitru Roman. An ontology framework for semantic business process management. *Proceedings of Wirtschaftsinformatik*, 2007.
- [HSFM06] Janelle Hill, Jim Sinur, David Flint, and Michael Melenovsky. Gartner’s Position on Business Process Management. *Gartner Research*, (G00136533), 2006.
- [Ima86] Masaaki Imai. *Kaizen: The Key To Japan’s Competitive Success*. McGraw-Hill/Irwin, New York, 1986.



- [Jac00] Michael Jackson. *Problem Frames: Analysing and Structuring Software Development Problems*. Addison-Wesley, New York, 2000.
- [JM04] S. Jones and N.A.M. Maiden. RESCUE: An integrated method for specifying requirements for complex socio-technical systems. In *Requirements Engineering for Sociotechnical Systems*. Idea Group Inc, 2004.
- [Kag04] L. Kagal. A Policy-Based Approach to Governing Autonomous Behaviour in Distributed Environments. In *Faculty of the Graduate School of the University of Maryland*, 2004.
- [Kar88] Jahangir Karimi. Strategic planning for information systems: requirements and information. *Journal of Management Information Systems*, 4(4):5–24, 1988.
- [Kav02] Evangelia Kavakli. Goal-Oriented Requirements Engineering: A Unifying Framework. *Requirements Engineering*, 6(4):237–251, 2002.
- [Kav04] Evangelia Kavakli. Modeling organizational goals: analysis of current methods. In *Proceedings of the 2004 ACM symposium on Applied computing*, Nicosia, Cyprus, 2004. ACM. 1339-1343.
- [KBBJ<sup>+</sup>02] Hermann Kaindl, Sjaak Brinkkemper, Janis A. Bubenko Jr, Barbara Farbey, Sol J. Greenspan, Constance L. Heitmeyer, Julio Cesar Sampaio do Prado Leite, Nancy R. Mead, John Mylopoulos, and Jawed Siddiqi. Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. *Requirements Engineering*, 7(3):113–123, 2002.
- [KDJ04] Barbara A. Kitchenham, Tore Dyba, and Magne Jorgensen. Evidence-Based Software Engineering. In *Proceedings of the 26th International Conference on Software Engineering*. IEEE Computer Society, 2004.
- [KG06] George Koliadis and Aditya Ghose. Relating Business Process Models to Goal-Oriented Requirements Models in KAOS. In *Advances in Knowledge Acquisition and Management*, pages 25–39. 2006.
- [Kit04] B. Kitchenham. Procedures for Performing Systematic Reviews. *Technical Report TR/SE0401, Keele University, and Technical Report 0400011T.1, National ICT Australia*, 2004.
- [KL99a] Evangelia Kavakli and P. Loucopoulos. Focus issue on legacy information systems and business process engineering: modelling of organisational change using the EKD framework. *Communications of the AIS*, 2(1):1–29, 1999.

- [KL99b] Evangelia Kavakli and Pericles Loucopoulos. Goal-driven business process analysis application in electricity deregulation. *Information Systems*, 24(3):187–207, 1999.
- [KL06] Birgit Korherr and Beate List. Extending the UML 2 Activity Diagram with Business Process Goals and Performance Measures and the Mapping to BPEL. In *Advances in Conceptual Modeling - Theory and Practice*, pages 7–18. 2006.
- [KL07] Birgit Korherr and Beate List. Extending the EPC and the BPMN with Business Process Goals and Performance Measures. In *ECEIS*, pages 287 – 294, 2007.
- [KN08] Robert S. Kaplan and David P. Norton. Execution Premium: Linking Strategy to Operations for Competitive Advantage. *Harvard Business School Press Books*, 2008.
- [KP99a] M. Koubarakis and D. Plexousakis. Business Process Modelling and Design A Formal Model and Methodology. *BT Technology Journal*, 17(4):23–35, 1999.
- [KP99b] Manolis Koubarakis and Dimitris Plexousakis. Business Process Modeling and Design: AI Models and Methodology. In *IJCAI-99 Workshop on Intelligent Workflow and Process Management: the New Frontier for AI in Business*, 1999.
- [KP00] Manolis Koubarakis and Dimitris Plexousakis. A Formal Model for Business Process Modeling and Design. In *Proceedings of the 12th International Conference on Advanced Information Systems Engineering*. Springer-Verlag, 2000. 142-156.
- [KP02] Manolis Koubarakis and Dimitris Plexousakis. A formal framework for business process modelling and design. *Information Systems*, 27(5):299–319, 2002.
- [KPBB<sup>+</sup>09] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology*, 51(1):7–15, 2009.
- [KPR04a] R. Kazhamiakin, M. Pistore, and M. Roveri. Formal Verification of requirements using SPIN: A Case Study on Web Services. In *3rd IEEE International Conference on Software Engineering and Formal Methods (SEFM)*, 2004.
- [KPR04b] Raman Kazhamiakin, Marco Pistore, and Marco Roveri. A Framework for Integrating Business Processes and Business Requirements. In *Proceedings of*

- the Enterprise Distributed Object Computing Conference, Eighth IEEE International*. IEEE Computer Society, 2004. 9-20.
- [Kun97] Ryuichi Kunitomo. Seven-Eleven is revolutionising grocery distribution in Japan. *Long Range Planning*, 30(6):877–889, 1997.
- [KVB<sup>+</sup>06a] G. Koliadis, A. Vranesevic, M. Bhuiyan, A. Krishna, and A. K. Ghose. A Combined Approach for Supporting the Business Process Model Lifecycle. In *Asia-Pacific Conference on Information Systems*, 2006.
- [KVB<sup>+</sup>06b] George Koliadis, Aleksandar Vranesevic, Moshir Bhuiyan, Aneesh Krishna, and Aditya Ghose. Combining i\* and BPMN for Business Process Model Lifecycle Management. In *BPM Workshops*, pages 416–427. 2006.
- [KZB<sup>+</sup>08] Nadzeya Kiyavitskaya, Nicola Zeni, Travis Breaux, Annie Anton, James Cordy, Luisa Mich, and John Mylopoulos. Automating the Extraction of Rights and Obligations for Regulatory Compliance. In *Conceptual Modeling - ER 2008*, pages 154–168. 2008.
- [Li03] Linyang Li. Seven-Eleven Japan - [http://iir.uwaterloo.ca/case\\_studies/e-business\\_2003/Seven-Eleven\\_Japan.pdf](http://iir.uwaterloo.ca/case_studies/e-business_2003/Seven-Eleven_Japan.pdf). *Technical Report - Centre for Business Education, St. Andrews University*, 2003.
- [LK95] P. Loucopoulos and E. Kavakli. Enterprise Modelling and the Teleological Approach to Requirements Engineering. *International Journal of Intelligent and Cooperative Information Systems*, 4(1):45–79, 1995.
- [LK05] Beate List and Birgit Korherr. A UML 2 Profile for Business Process Modelling. In *Perspectives in Conceptual Modeling*, pages 85–96. 2005.
- [LK06] Beate List and Birgit Korherr. An evaluation of conceptual business process modelling languages. In *Proceedings of the 2006 ACM symposium on Applied computing*, Dijon, France, 2006. ACM. 1532-1539.
- [LL90] Edwin A. Locke and Gary P. Latham. *A theory of goal setting and task performance*. Prentice Hall, NY, 1990.
- [LL06a] Alexei Lapouchnian and Yves Lesperance. Modeling Mental States in Agent-Oriented Requirements Engineering. In *Advanced Information Systems Engineering*, pages 480–494. 2006.
- [LL06b] Edwin A. Locke and Gary P. Latham. New Directions in Goal-Setting Theory. *Current Directions in Psychological Science (Wiley-Blackwell)*, 15(5):265–268, 2006.

- [LY04] Lin Liu and Eric Yu. Designing information systems in social context: a goal and scenario modelling approach. *Information Systems*, 29(2):187–203, 2004.
- [LY08] Amy Lo and Eric Yu. From Business Models to Service-Oriented Design: A Reference Catalog Approach. In *Conceptual Modeling - ER 2007*, pages 87–101. 2008.
- [LYM07] Alexei Lapouchnian, Yijun Yu, and John Mylopoulos. Requirements-Driven Design and Configuration Management of Business Processes. In *Business Process Management*, pages 246–261. 2007.
- [Mar89] James Martin. Information Engineering (3 Volumes). *Prentice-Hall Inc*, 1989.
- [MCY99] John Mylopoulos, Lawrence Chung, and Eric Yu. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1):31–37, 1999.
- [MH94] Matthew Miles and Michael Huberman. *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE, 1994.
- [MJ00] Kevin P. McCormack and William C. Johnson. *Business Process Orientation: Gaining the E-Business Competitive Advantage*. Saint Lucie Press, 2000.
- [MJM<sup>+</sup>04] Neil A. M. Maiden, Sara V. Jones, Sharon Manning, John Greenwood, and L. Renou. Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study. In *Advanced Information Systems Engineering*, pages 3–21. 2004.
- [MK08] Ivan Markovic and Marek Kowalkiewicz. Linking Business Goals to Process Models in Semantic Business Process Modeling. In *EDOC Proceedings*, 2008.
- [ML08] Michela Montesi and Patricia Lago. Software engineering article types: An analysis of the literature. *Journal of Systems and Software*, 81(10):1694–1714, 2008.
- [MM03] Richard E. Mayer and Roxana Moreno. Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist*, 38(1):43 – 52, 2003.
- [MMZ08] Fabio Massacci, John Mylopoulos, and Nicola Zannone. An Ontology for Secure Socio-Technical Systems. In Peter Rittgen, editor, *Handbook of Ontologies for Business Interaction*. Information Science Reference, 2008.
- [Moo05] Daniel L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current

- state and future directions. *Data and Knowledge Engineering*, 55(3):243–276, 2005.
- [Moo09] Daniel L. Moody. The Physics of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.
- [MP07] Ann Maes and Geert Poels. Evaluating quality of conceptual modelling scripts based on user perceptions. *Data and Knowledge Engineering*, 63(3):701–724, 2007.
- [MRM04] Donella Meadows, Jorgen Randers, and Dennis Meadows. *Limits to Growth: The 30-Year Update*. Earthscan, London, 2004.
- [Myl98] John Mylopoulos. Information modeling in the time of the revolution. *Information Systems*, 23(3-4):127–155, 1998.
- [NC03] Dina Neiger and Leonid Churilov. Structuring Business Objectives: A Business Process Modeling Perspective. In *Business Process Management*. 2003.
- [NC04a] Dina Neiger and Leonid Churilov. Goal-Oriented Business Process Engineering Revisited: a Unifying Perspective. In *1st International Workshop on Computer Supported Activity Coordination 2004*, 2004.
- [NC04b] Dina Neiger and Leonid Churilov. Goal-Oriented Business Process Modeling with EPCs and Value-Focused Thinking. In *Business Process Management*, pages 98–115. 2004.
- [NC06] Dina Neiger and Leonid Churilov. Intelligent Decision Support through Synchronized Decomposition of Process and Objectives Structures. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences - Volume 02*. IEEE Computer Society, 2006.
- [NE05] Selmin Nurcan and Marc-Henri Edme. Intention-driven modeling for flexible workflow applications. *Software Process: Improvement and Practice*, 10(4):363–377, 2005.
- [NEK<sup>+</sup>05] Selmin Nurcan, Anne Etien, Rim Kaabi, Iyad Zoukar, and Colette Rolland. A strategy driven business process modelling approach. *Business Process Management Journal*, 11:628–649, 2005.
- [Nur04a] Selmin Nurcan. Business Process Modeling for developing Process Oriented IT Systems. In *The Business Process Management Tools and Technologies track of*

- the 2004 Information Resources Management Association International Conference*, New Orleans, USA, 2004.
- [Nur04b] Selmin Nurcan. A Conceptual Framework for Intention Driven Flexible Workflow Modeling. In *BPMDs*, pages 192–199, 2004.
- [NW04] Kei Nagayama and Peter Weill. Seven Eleven Japan: Reinventing the Retail Business Model. In *CISR Working Paper 338 - MIT Sloan WP 4485-04*, 2004.
- [OAS07] OASIS. Web Services Business Process Execution Language Version 2.0 - <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>. 2007.
- [OMG09a] OMG. BPMN 1.2 specification - <http://www.omg.org/spec/BPMN/1.2/>. 2009.
- [OMG09b] OMG. Business Motivation Model - <http://www.omg.org/spec/BMM/>. 2009.
- [OMG09c] OMG. Model Driven Architecture - <http://www.omg.org/mda/>. 2009.
- [Oul95] M.A. Ould. *Business Processes: Modelling and Analysis for Reengineering and Improvement*. Wiley, Chichester, New York, 1995.
- [PM85] Michael Porter and Victor Millar. How information gives you competitive advantage. *Harvard Business Review*, 63(4):149–60, 1985.
- [PRB04] Marco Pistore, Marco Roveri, and Paolo Busetta. Requirements-Driven Verification of Web Services. *Electronic Notes in Theoretical Computer Science*, 105:95–108, 2004.
- [PRdRW09] Volkmar Pipek, Mary Beth Rosson, Boris de Ruyter, and Volker Wulf. *End-User Development - 2nd International Symposium, IS-EUD 2009, Siegen, Germany, March 2-4, 2009. Proceedings*. Springer Berlin / Heidelberg, 2009.
- [PS89] Patricia A. Prescott and Karen L. Soeken. The Potential Uses of Pilot Work. *Nursing Research*, 38(1):60, 1989.
- [RB10] Pascal Ravesteyn and Ronald Batenburg. Surveying the critical success factors of BPM-systems implementation. *Business Process Management Journal*, 16(3):492–507, 2010.
- [RCB07] Balasubramaniam Ramesh, Lan Cao, and Richard Baskerville. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, pages 1–32, 2007.

- [RD08] Andre Rifaut and Eric Dubois. Using Goal-Oriented Requirements Engineering for Improving the Quality of ISO/IEC 15504 based Compliance Assessment Frameworks. In *Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*. IEEE Computer Society, 2008. 33-42.
- [Rec10] Jan Recker. Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal*, 16(1):181–201, 2010.
- [RNG99] Colette Rolland, Selmin Nurcan, and Georges Grosz. Enterprise knowledge development: the process view. *Information and Management*, 36(3):165–184, 1999.
- [RPB99] C. Rolland, N. Prakash, and A. Benjamen. A Multi-Model View of Process Modelling. *Requirements Engineering*, 4(4):169–187, 1999.
- [RSB05] Gil Regev, Pnina Soffer, and Ilia Bider. Coordinated development of business processes and their support systems. *Requirements Engineering*, 10(3):173–174, 2005. 10.1007/s00766-005-0006-5.
- [RSM95] Colette Rolland, Carine Souveyet, and Mario Moreno. An approach for defining ways-of-working. *Information Systems*, 20(4):337–359, 1995.
- [SF04] Howard Smith and Peter Fingar. *Business Process Management (BPM): The Third Wave*. 2004.
- [SGM04] Roberto Sebastiani, Paolo Giorgini, and John Mylopoulos. Simple and Minimum-Cost Satisfiability for Goal Models. In *Advanced Information Systems Engineering*, pages 20–35. 2004.
- [SGRS04] Wasana Sedera, Guy G. Gable, Michael Rosemann, and Robert W. Smyth. A success model for business process modeling: findings from a multiple case study. In *PACIS*, 2004.
- [SHF08] Magali Seguran, Cedric Hebert, and Ganna Frankova. Secure Workflow Development from Early Requirements Analysis. In *The 6th IEEE European Conference on Web Services*, 2008.
- [SHK<sup>+</sup>07] Duncan R. Shaw, Christopher P. Holland, Peter Kawalek, Bob Snowdon, and Brian Warboys. Elements of a business process management system: theory and practice. *BPMJ*, 13:91–107, 2007. doi:10.1108/14637150710721140.
- [Sil09] Bruce Silver. *BPMN Method and Style*. Cody-Cassidy Press, 2009.

- [SML<sup>+</sup>08] Alberto Siena, Neil Maiden, James Lockerbie, Kristine Karlsen, Anna Perini, and Angelo Susi. Exploring the Effectiveness of Normative i \* Modelling: Results from a Case Study on Food Chain Traceability. In *Advanced Information Systems Engineering*, pages 182–196. 2008.
- [Som04] Ian Sommerville. *Software Engineering - 7th edition*. Addison Wesley, 2004.
- [Son99] R Sondhi. *Total Strategy*. Airworthy Publications International Ltd, 1999.
- [SR05] Pnina Soffer and Colette Rolland. Combining Intention-Oriented and State-Based Process Modeling. In *Conceptual Modeling ER 2005*, pages 47–62. 2005.
- [SR07] Keng Siau and Matti Rossi. Evaluation techniques for systems analysis and design modelling methods - a review and comparative analysis. *Information Systems Journal*, pages 1–20, 2007.
- [SRD03] Wasana Sedera, Michael Rosemann, and Gabriella Doebeli. A Process Modelling Success Model: Insights from a Case Study. In *ECIS*, 2003.
- [SRG02] Wasana Sedera, Michael Rosemann, and Guy G. Gable. Measuring Process Modelling Success. In *ECIS*. University of Gdansk, 2002.
- [SSFM08] Todor Stoitsev, Stefan Scheidl, Felix Flentge, and Max Mhlhuser. From Personal Task Management to End-User Driven Business Process Modeling. In *Business Process Management*, pages 84–99. 2008.
- [SSFM09] Todor Stoitsev, Stefan Scheidl, Felix Flentge, and Max Mhlhuser. Enabling End Users to Proactively Tailor Underspecified, Human-Centric Business Processes: Programming by Example of Weakly-Structured Process Models. In *Enterprise Information Systems*, pages 307–320. 2009.
- [SSS08] Todor Stoitsev, Michael Spahn, and Stefan Scheidl. EUD for enterprise process and information management. In *Proceedings of the 4th international workshop on End-user software engineering*, 2008.
- [SUP07] SUPER. D5.1: Semantic Process Modelling Environment - [www.super-ip.org](http://www.super-ip.org). 2007.
- [SUP09] SUPER. Business Process Modelling Ontology v2 - <http://www.ip-super.org/ontologies/process/bpmo/v2.0.1>. 2009.



- [SW04] Pnina Soffer and Yair Wand. Goal-Driven Analysis of Process Model Validity. In *Advanced Information Systems Engineering*, pages 521–535. 2004.
- [SW05] Pnina Soffer and Yair Wand. On the notion of soft-goals in business process modeling. *Business Process Management Journal*, 11:663–679, 2005.
- [Swe94] John Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312, 1994.
- [TPR<sup>+</sup>04] P. Traverso, M. Pistore, M. Roveri, A. Marconi, R. Kazhamiakin, P. Lucchese, P. Busetta, and P. Bertoli. Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development. In *2nd International Conference on Service Oriented Computing (ICSOC)*, 2004.
- [Tur05] Rodney Turner. The role of pilot studies in reducing risk on projects and programmes. *International Journal of Project Management*, 23(1):1–6, 2005.
- [Tur08] Graham M. Turner. A comparison of The Limits to Growth with 30 years of reality. *Global Environmental Change*, 18:397–411, 2008.
- [TWC08] Jose M. Tribolet, Robert Winter, and Artur Caetano. Special track on Organizational Engineering: editorial message. In *ACM SAC*, pages 516–517, 2008.
- [VCN<sup>+</sup>01] Andre Vasconcelos, Artur Caetano, Joao Neves, Pedro Sinogas, Ricardo Mendes, and Jose Tribolet. A Framework for Modeling Strategy, Business Processes and Information Systems. In *Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*. IEEE Computer Society, 2001.
- [vdA04] Wil M. P. van der Aalst. Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. In *Lectures on Concurrency and Petri Nets*, pages 21–58. 2004.
- [Ves91] I. Vessey. Cognitive Fit: A Theory-based Analysis of the Graphs versus Tables Literature. *Decision Sciences*, 22:219–240, 1991.
- [vL00] Axel van Lamsweerde. Requirements engineering in the year 00: a research perspective. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 5–19, New York, NY, USA, 2000. ACM.
- [vL09] Axel van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. John Wiley And Sons, Inc., 2009.

- [vTH03] E. van Teijlingen and V. Hundley. The importance of pilot studies. *Social Science Research Update, University of Surrey, UK*, 35:Available from <http://sru.soc.surrey.ac.uk/SRU35.html>, 2003.
- [vTRHG01] E. R. van Teijlingen, A. M. Rennie, V. Hundley, and W. Graham. The importance of conducting and reporting pilot studies: the example of the Scottish Births Survey. *Journal of Advanced Nursing*, 34:289–295, 2001.
- [Wac02] Mathis Wackernagel. Tracking the ecological overshoot of the human economy. *Proceedings of the Academy of Science*, 99(14):9266–9271, 2002.
- [Wal88] Mary Walton. *The Deming Management Method*. Perigee Books, New York, 1988.
- [WG06] Jeff Woods and Yvonne Genovese. Delivery of Commodity Business Applications in a BPMS Does Not Mean You Should Customize the Applications. *Gartner Research*, G00139084, 2006.
- [WH06] R. Wieringa and J. Heerkens. The methodological soundness of requirements engineering papers: a conceptual framework and two case studies. *Requirements Engineering*, 11(4):295–307, 2006.
- [Wie96] R. J. Wieringa. *Requirements engineering: frameworks for understanding*. John Wiley And Sons, Inc., 1996.
- [Wie05] Roel Wieringa. Requirements researchers: are we really doing research? *Requirements Engineering*, 10(4):304–306, 2005. 10.1007/s00766-005-0013-6.
- [WL00] Kent A. Walstrom and Lori N. K. Leonard. Citation classics from the information systems literature. *Information and Management*, 38(2):59–72, 2000.
- [WMMR06] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006.
- [WV01] P. Weill and M. Vitale. *Place to Space: Moving to eBusiness Models*. Harvard Business School Publishing Corporation, Boston, 2001.
- [WW02] Yair Wand and Ron Weber. Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, 13(4):363–376, 2002.
- [Yin09] Robert K. Yin. *Case Study Research: Design and Methods, Fourth Edition*, volume 5 of *Applied Social Research Method Series*. SAGE Publications, 2009.

- [Yu97] E. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, pages 226–235, 1997.
- [ZW98] Marvin V. Zelkowitz and Dolores R. Wallace. Experimental Models for Validating Technology. *Computer*, 31(5):23–31, 1998.





## Appendices for Chapter 3

```

ACTOR DT_NED Attribute localization : String, subcontractor : SUB_CONTRACTOR
ACTOR SC_GRE Attribute localization : String
ACTOR SUB_CONTRACTOR
ACTOR COMPANY_X
ACTOR US_CUSTOMER

ENTITY SHIPMENT
  Attribute actual_delivery_date : Date, planned_sending_date : Date

ENTITY HARBOR
  Attribute location : String

TASK TREAT_TOXIC_DEFECTS
  Mode Achieve
  Actor DT_NED
  Attribute reception_date : Date, finalization_date : Date

TASK DELIVER_SHIPMENT
  Mode Achieve
  Actor SC_GRE
  Attribute shipment : SHIPMENT

TASK SHIPPING_MINERAL
  Mode Achieve
  Actor SC_GRE
  Attribute stop_harbors : HARBOR

POLICY_SET NED_POLSET
  Actor DT_NED
  POLICY NED_POL1
    PolicyTarget TREAT_TOXIC_DEFECTS
    Deontic obligation
    PolicyConstraint Constraint Condition TREAT_TOXIC_DEFECTS.finalization_date -
      TREAT_TOXIC_DEFECTS.reception_date <= 90 days
  AND
  POLICY NED_POL2
    PolicyTarget TREAT_TOXIC_DEFECTS
    Deontic permission
    PolicyDependency ActiveDepender DT_NED
      PassiveDepender SUB_CONTRACTOR delegation NED_POL1

POLICY_SET GRE_POLSET
  Actor SC_GRE
  POLICY GRE_POL1
    PolicyTarget DELIVER_SHIPMENT
    Deontic obligation
    PolicyConstraint Constraint Condition SHIPMENT.actual_delivery_date
      - SHIPMENT.planned_sending_date < 21 days
  AND
  POLICY GRE_POL2
    PolicyTarget DELIVER_SHIPMENT
    Deontic obligation
    PolicyConstraint Constraint Condition SHIPMENT.actual_delivery_date
      - SHIPMENT.planned_sending_date > 7 days
  AND
  POLICY GRE_POL3
    PolicyTarget DELIVER_SHIPMENT
    Deontic permission
    PolicyConstraint Constraint Condition SHIPMENT.actual_delivery_date
      - SHIPMENT.planned_sending_date <= 14 days
  AND
  POLICY GRE_POL4
    PolicyTarget SHIPPING_MINERAL
    Deontic prohibition
    PolicyConstraint Constraint Condition ∃(sh:SHIPPING_MINERAL.stop_harbor)

```

Figure A.1: Policy-extended Formal Tropos Specification Generated by Visual Editor

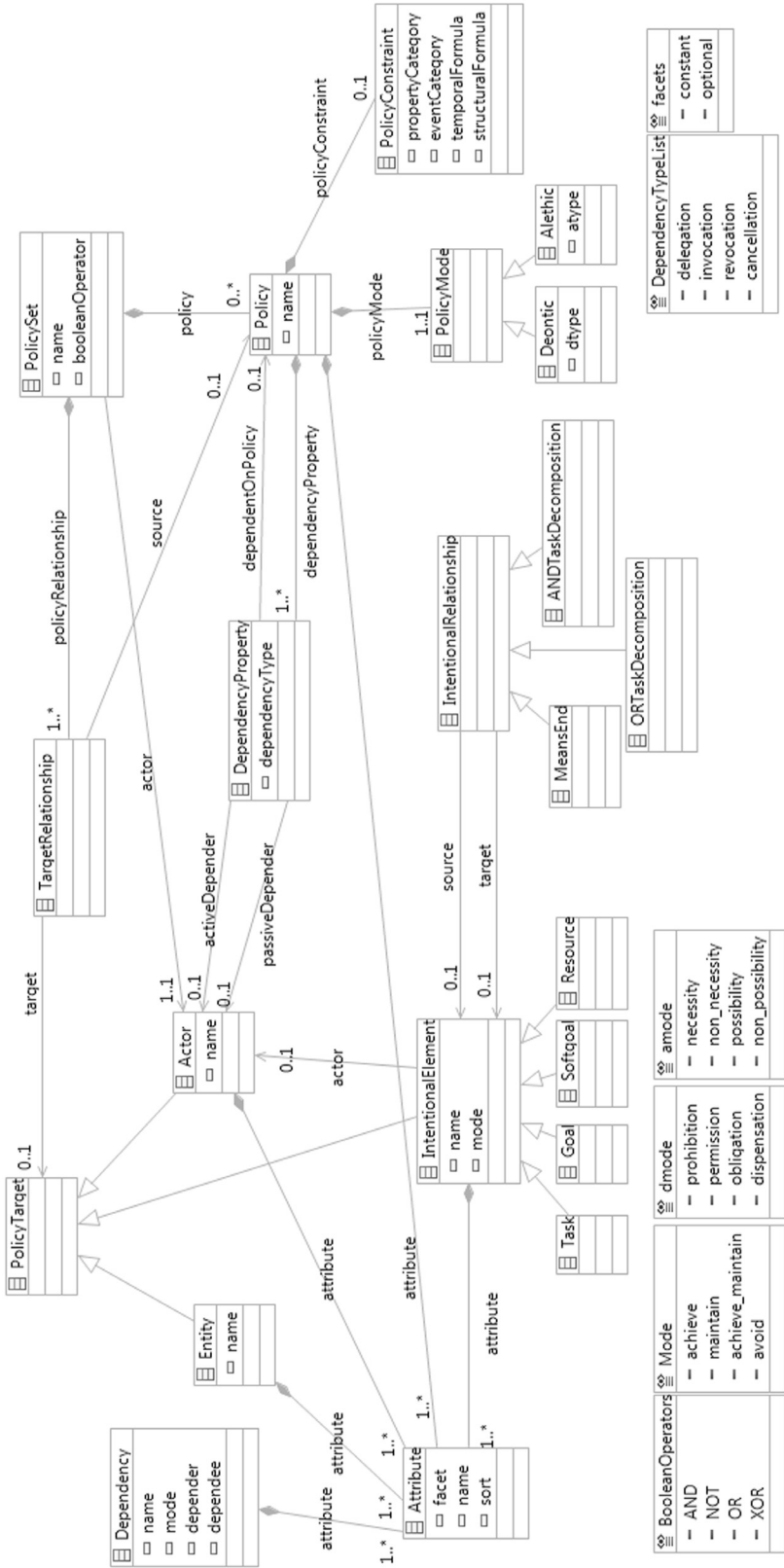


Figure A.2: Ecore metamodel of Policy-extended Formal Tronos

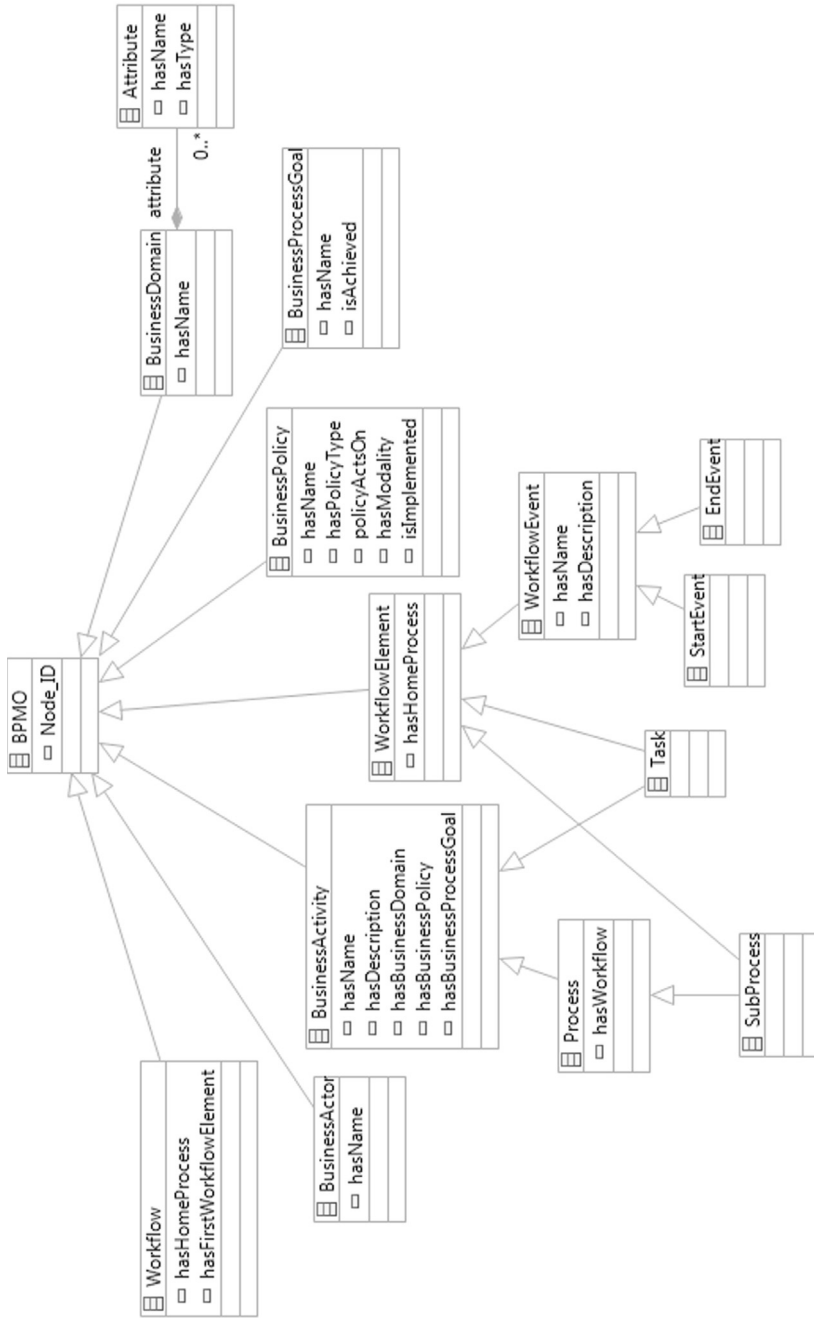


Figure A.3: Ecore metamodel of Business Process Modelling Ontology



```

helper context PFT!Task def: hasSource(taskNode : String) : Boolean =
  PFT!IntentionalRelationship.allInstances()->collect(f|f.source)-
  >exists(e|e.name = taskNode);

rule actor2lane {
  from
    a : PFT!Actor
  to
    bpmo_process : BPMO!Process(Node_ID<-a.name, hasName <- a.name,
      hasWorkflow<- 'Workflow_'+a.name),
    bpmo_startevent : BPMO!StartEvent(Node_ID<- 'StartEvent_ID',
      hasHomeProcess<- bpmo_process.Node_ID, hasName<- 'Start node'),
    bpmo_endevent : BPMO!EndEvent(Node_ID<- 'EndEvent_ID',
      hasHomeProcess<- bpmo_process.Node_ID, hasName<- 'End node'),
    bpmo_workflow : BPMO!Workflow(Node_ID<- 'Workflow_'+a.name,
      hasHomeProcess<- bpmo_process.Node_ID,
      hasWorkflowElement<- bpmo_startevent.Node_ID)
}

rule task2subprocess{
  from
    --When task node has an incoming source, it has children
    a : PFT!Task(a.hasSource(a.name))
  to
    bpmo_subprocess : BPMO!SubProcess(
      Node_ID<- 'SubProcess_'+a.name, hasName<- a.name,
      hasWorkflow<- 'Workflow_'+a.name,
      hasHomeProcess<- 'Process_'+a.name),
    bpmo_workflow : BPMO!Workflow(Node_ID<- 'Workflow_'+a.name,
      hasHomeProcess<- 'Process_'+a.name,
      hasWorkflowElement<- 'StartEvent_ID')
}

rule task2task {
  from
    --When task node has no incoming sources, it is a leaf node
    a : PFT!Task(not a.hasSource(a.name))
  to
    bpmo_task : BPMO!Task(Node_ID<- 'Task_ID', hasName<- a.name,
      hasHomeProcess<- 'Process_ID')
}

rule goal2businessprocessgoal{
  from
    a : PFT!Goal
  to
    b : BPMO!BusinessProcessGoal(
      Node_ID<-a.name, hasName <- a.name)
}

rule policy2businesspolicy{
  from
    a : PFT!Policy
  to
    b : BPMO!BusinessPolicy(Node_ID<-a.name, hasName <- a.name)
}

```

*Figure A.4: PFT2BPMO Model Transformations*

```

ontology Policy_Enriched_BP_Of_Company_X

instance Pool_DT_NED memberOf Process
  hasName hasValue "Pool of actor DT_NED"
  hasBusinessDomain hasValue {localization, subcontractor}

instance Pool_SC_GRE memberOf Process
  hasName hasValue "Pool of actor SC_GRE"
  hasBusinessDomain hasValue {localization}

instance TREAT_TOXIC_DEFECTS memberOf Task
  hasName hasValue "Treat toxic defects"
  hasWorkflow hasValue Pool_DT_NED_Workflow
  hasBusinessDomain hasValue {reception_date, finalization_date}
  hasBusinessPolicy hasValue {NED_POL1, NED_POL2}

instance DELIVER_SHIPMENT memberOf SubProcess
  hasName hasValue "Deliver Shipment"
  hasWorkflow hasValue Pool_SC_GRE_Workflow
  hasBusinessProcessGoal hasValue DELIVER_SHIPMENT
  hasBusinessDomain hasValue {stop_harbors}
  hasBusinessPolicy hasValue {GRE_POL1, GRE_POL2, GRE_POL3}

instance SHIPPING_MINERAL memberOf Task
  hasName hasValue "Ship Mineral"
  hasWorkflow hasValue Pool_SC_GRE_Workflow
  hasBusinessProcessGoal hasValue DELIVER_SHIPMENT
  hasBusinessDomain hasValue {stop_harbors}
  hasBusinessPolicy hasValue {GRE_POL4}

instance NED_POL1 memberOf BusinessPolicy
  hasName hasValue "NED_POL1"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue DT_NED
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Obligation"
  structuralConstraint hasValue "TREAT_TOXIC_DEFECTS.finalization_date -
                                TREAT_TOXIC_DEFECTS.reception_date ≤ 90 days"
  isImplemented hasValue TRUE

instance NED_POL2 memberOf BusinessPolicy
  hasName hasValue "NED_POL2"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue DT_NED
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Permission"
  isImplemented hasValue TRUE

instance GRE_POL1 memberOf BusinessPolicy
  hasName hasValue "GRE_POL1"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue SC_GRE
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Obligation"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "SHIPMENT.actual_delivery_date
                                - SHIPMENT.planned_sending_date < 21 days"

instance GRE_POL2 memberOf BusinessPolicy
  hasName hasValue "GRE_POL2"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue SC_GRE
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Obligation"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "SHIPMENT.actual_delivery_date
                                - SHIPMENT.planned_sending_date > 7 days"

instance GRE_POL3 memberOf BusinessPolicy
  hasName hasValue "GRE_POL3"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue SC_GRE
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Permission"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "SHIPMENT.actual_delivery_date
                                - SHIPMENT.planned_sending_date ≤ 14 days"

instance GRE_POL4 memberOf BusinessPolicy
  hasName hasValue "GRE_POL4"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue SC_GRE
  hasModality hasValue "http://www.ip-super.org/ontologies/BPRO/20070831#Prohibition"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "¬(sh:SHIPPING_MINERAL.stop_harbor)"

```

Figure A.5: Output of Atlas Transformation Language Mappings for Company X

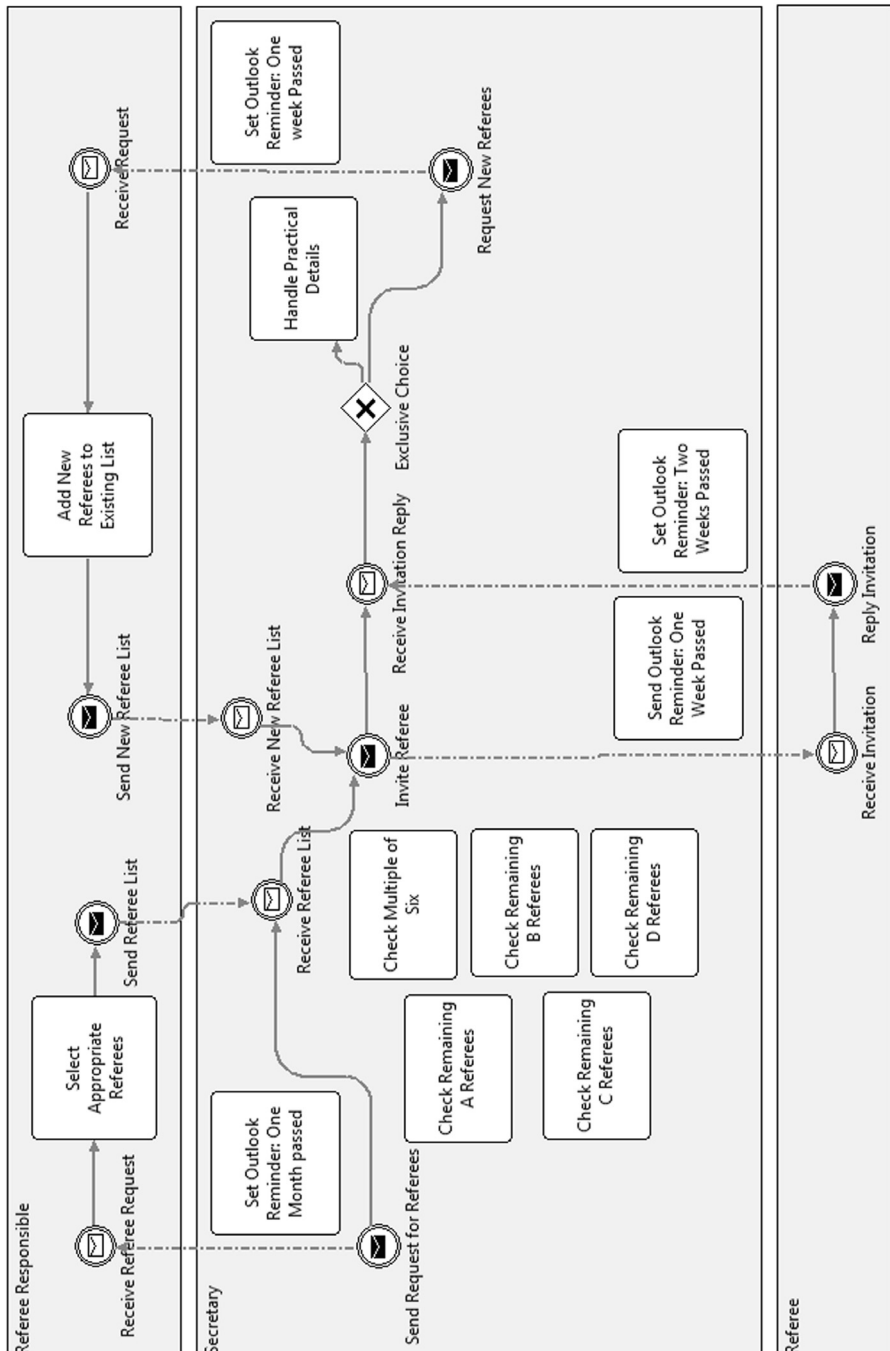


Figure A.6: Policy-enabled business process model (with regular business process modelling tool)

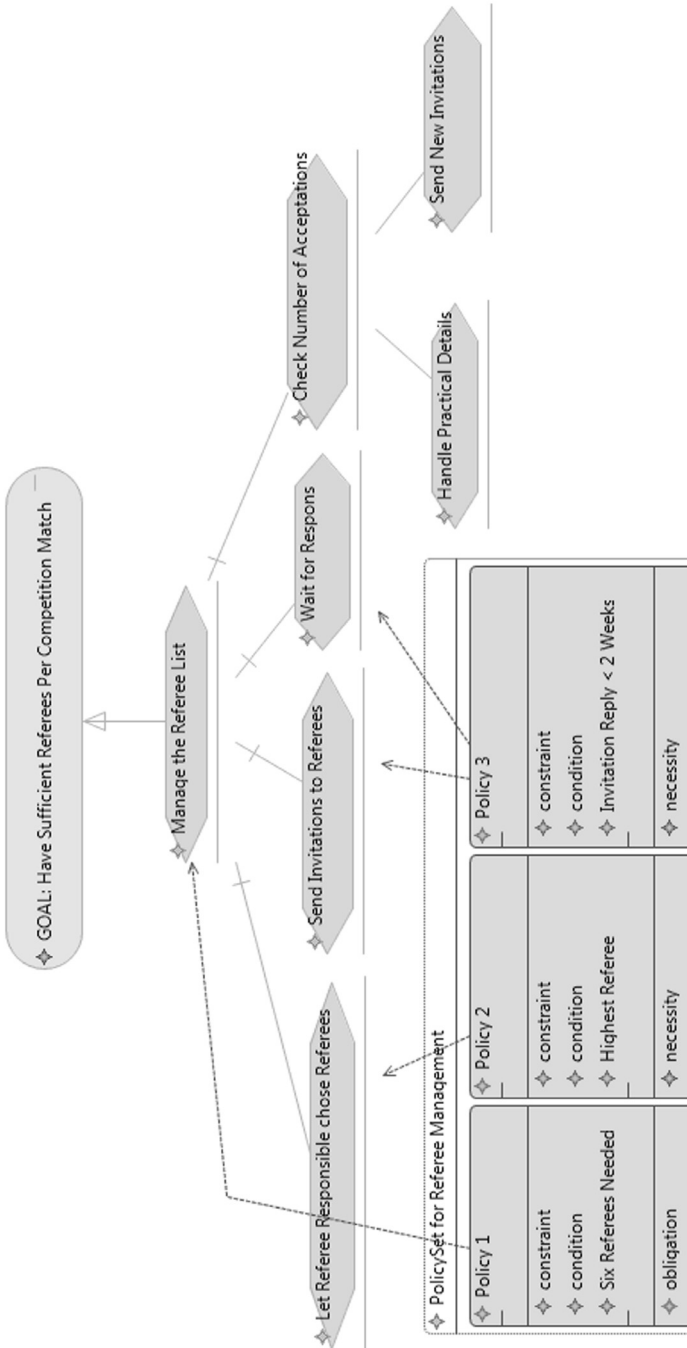


Figure A.7: Policy-extended Formal Tropos model



```
instance Policy_1 memberOf BusinessPolicy
  hasName hasValue "Policy 1"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue Secretary
  hasModality hasValue "ontologies/BPRO/20070831#Obligation"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "Six Referees Needed"
```

*Figure A.9: Subsection of Textual Part for Figure A.8*

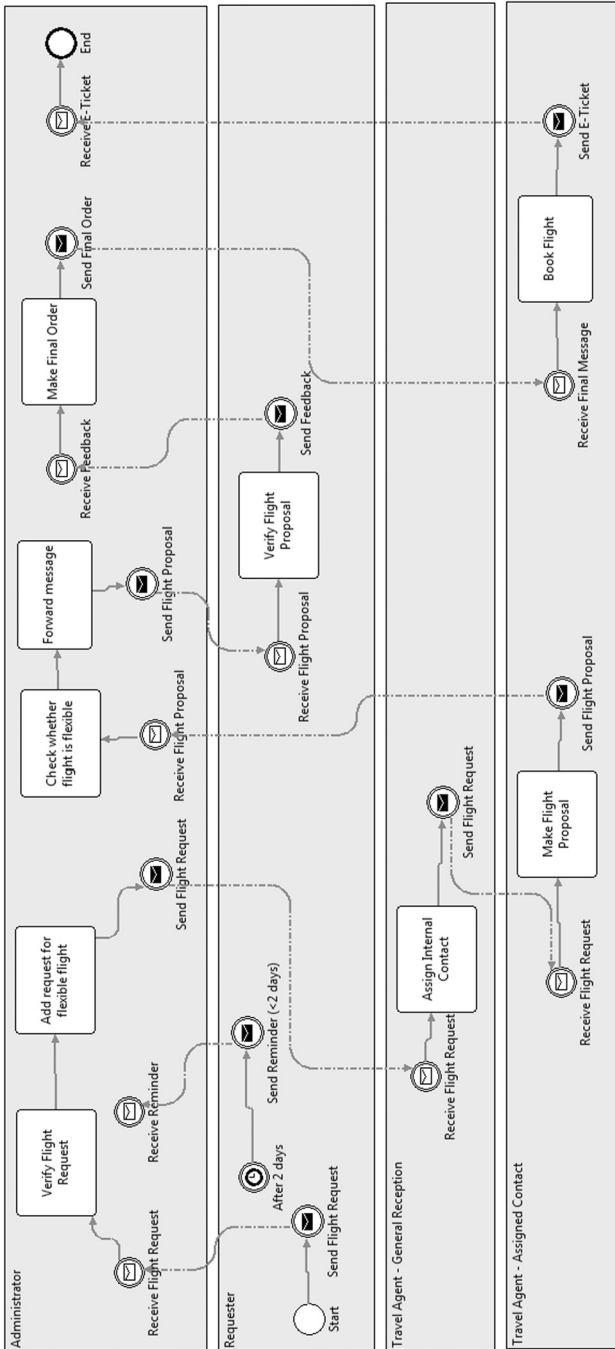


Figure A.10: Policy-enabled business process model (with regular business process modelling tool)

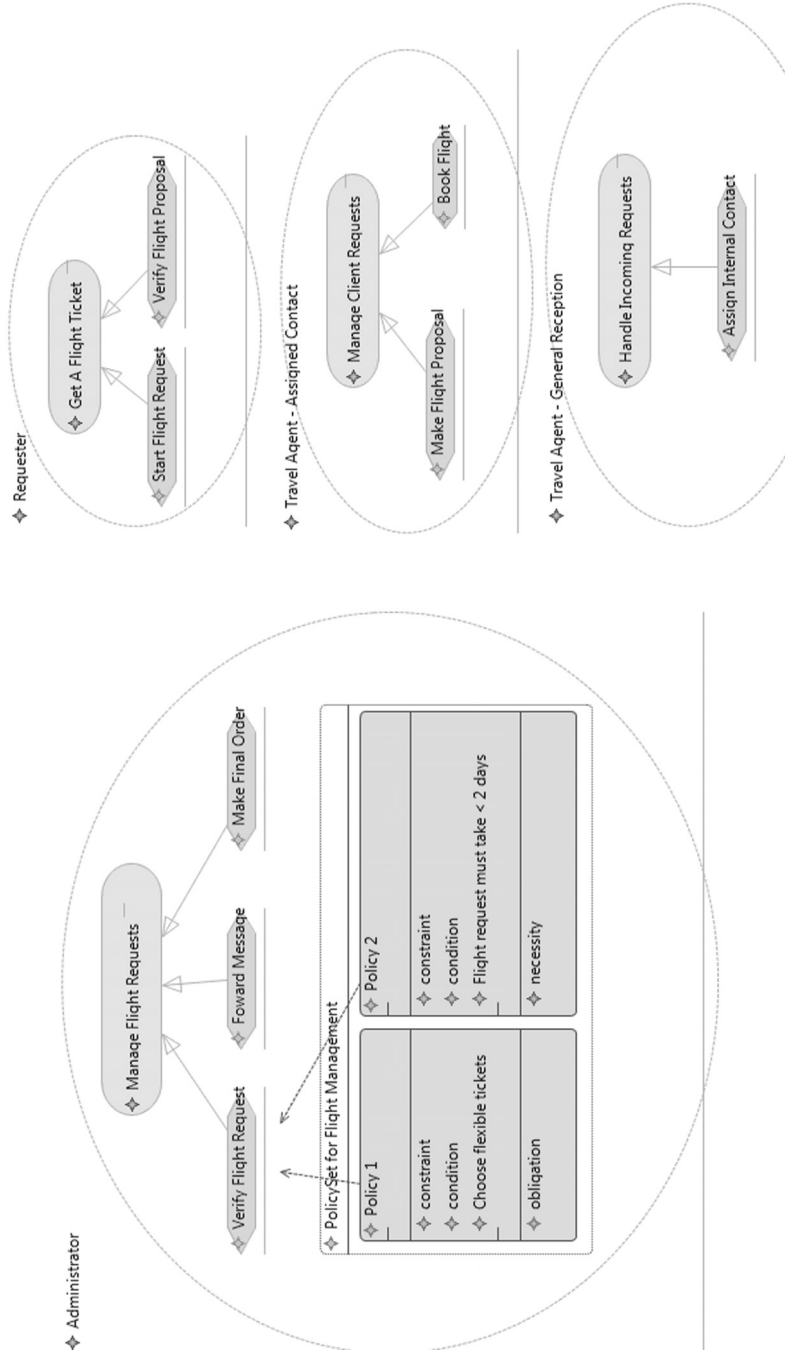


Figure A.11: Policy-extended Formal Tropos model



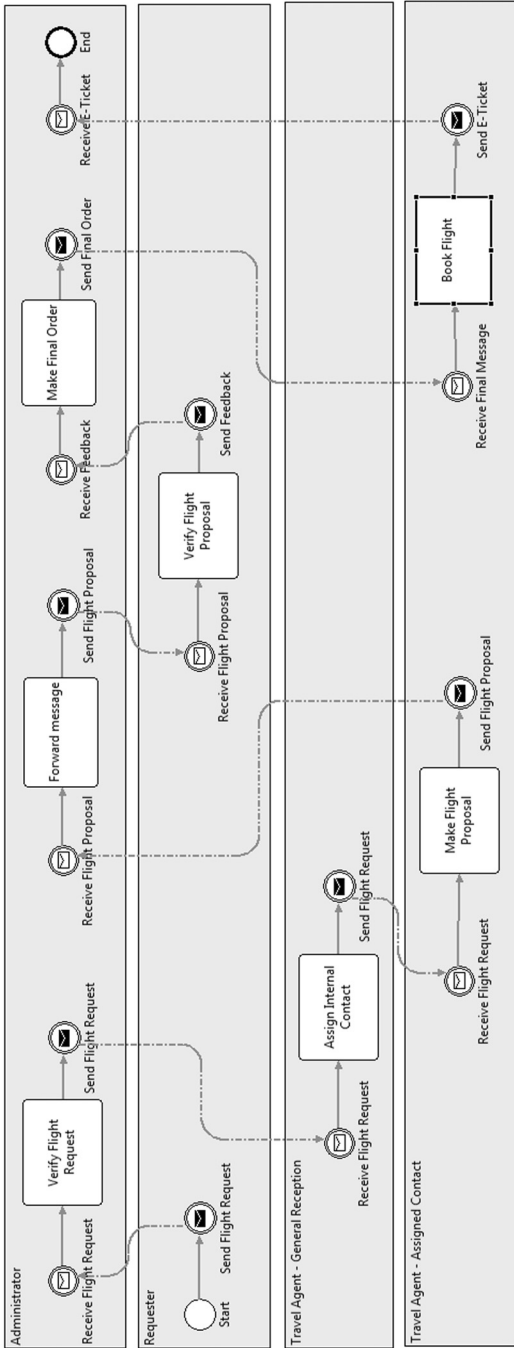


Figure A.12: Policy-enabled business process model (based on Policy-extended Formal Tropos model)

```
instance Policy_1 memberOf BusinessPolicy
  hasName hasValue "Policy 1"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue Administrator
  hasModality hasValue "ontologies/BPRO/20070831#Obligation"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "Choose flexible tickets"
```

*Figure A.13: Subsection of Textual Part for Figure A.12*

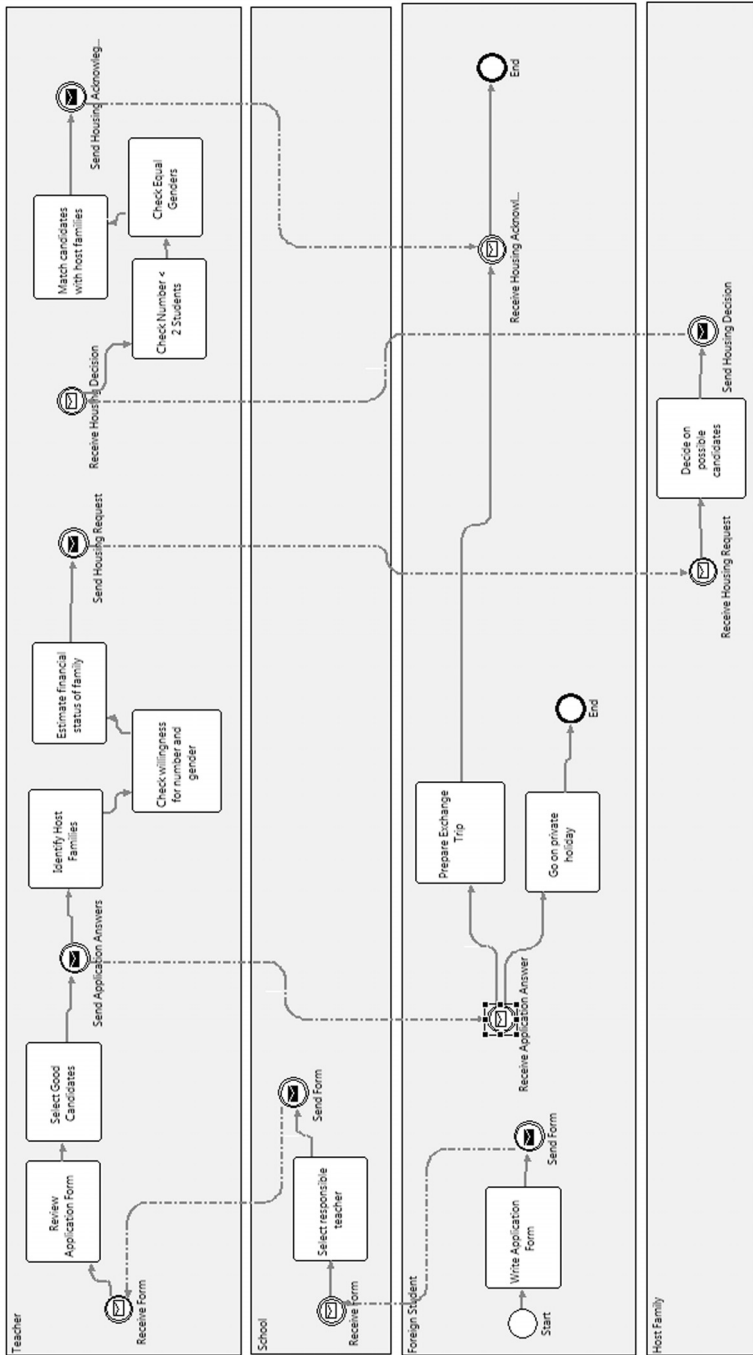


Figure A.14: Policy-enabled business process model (with regular business process modelling tool)

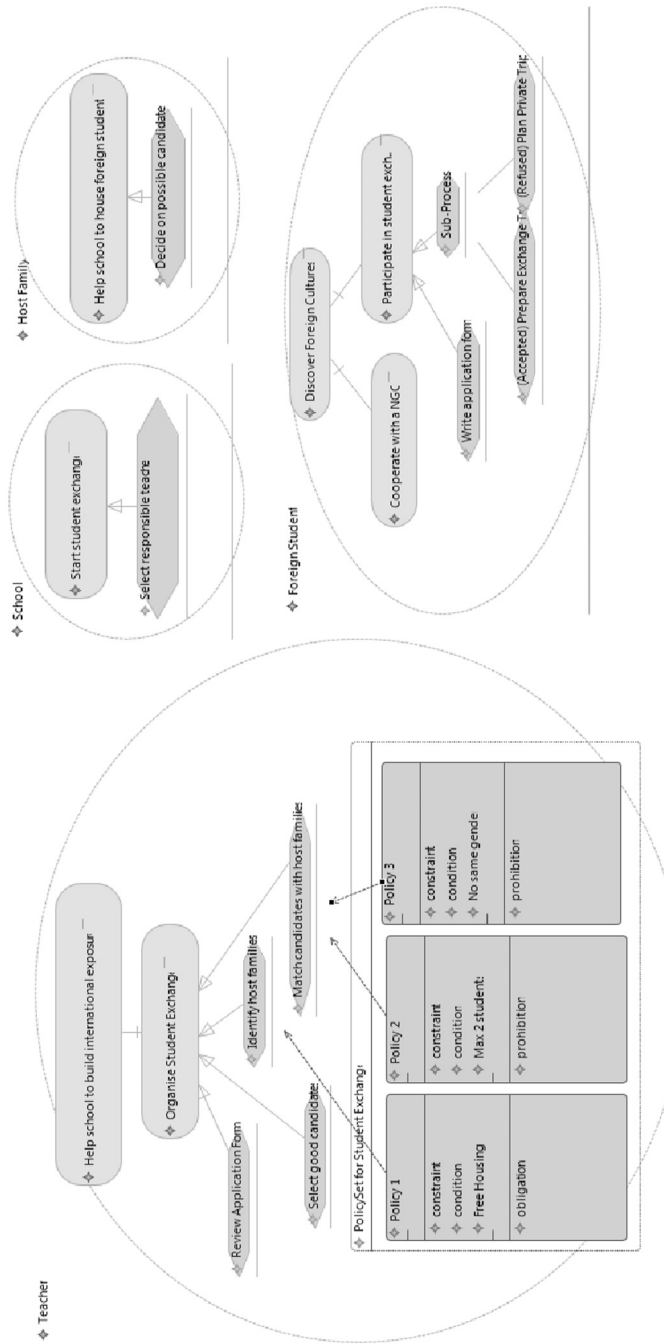


Figure A.15: Policy-extended Formal Tropos model

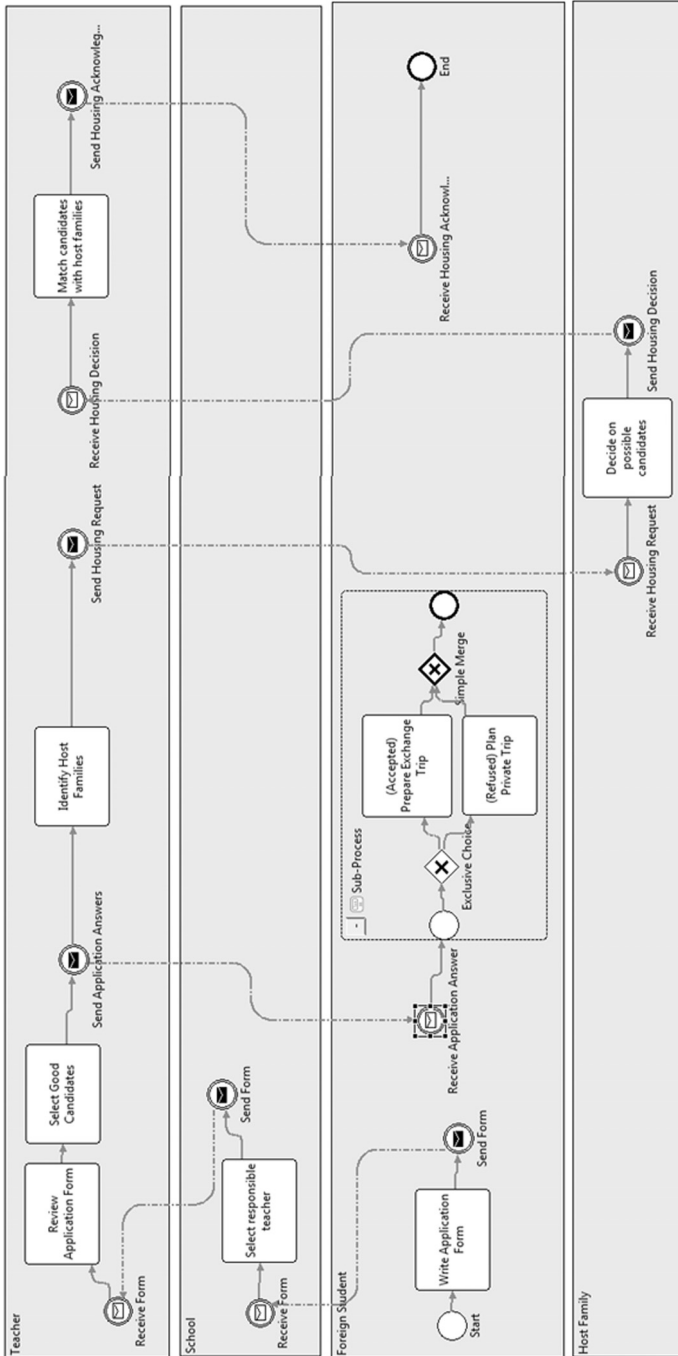


Figure A.16: Policy-enabled business process model (based on Policy-extended Formal Tropos model)

```
instance Policy_2 memberOf BusinessPolicy
  hasName hasValue "Policy 2"
  hasPolicyType hasValue "ConstraintPolicy"
  policyActsOn hasValue Administrator
  hasModality hasValue "ontologies/BPRO/20070831#Prohibition"
  isImplemented hasValue TRUE
  structuralConstraint hasValue "Max 2 Students"
```

*Figure A.17: Subsection of Textual Part for Figure A.16*

# B

## Appendices for Chapter 4

```

helper context BSCP!Tactic def: hasSource(nodeName : String) : Boolean =
  BSCP!Relationship.allInstances()->collect(f|f.hasSource)->exists(e|e.name = nodeName);
helper context BSCP!Tactic def: hasTarget(nodeName : String) : Boolean =
  BSCP!Relationship.allInstances()->collect(f|f.hasTarget)->exists(e|e.name = nodeName);

(1) rule TopNode2BPMNDiagram{
  from a : BSCP!Tactic (a.hasSource(a.name) and not a.hasTarget(a.name))
  to b : BPMN!BpmnDiagram(name <- a.name)}

(2) rule DomainOfInterest2Pool{
  from a : BSCP!DomainOfInterest
  to b : BPMN!Pool(name <- a.name),
    startevent : BPMN!Activity(activityType <- 'EventStartEmpty'),
    endevent : BPMN!Activity(activityType <- 'EventEndEmpty'),
    firstSequence : BPMN!SequenceEdge}

(3) rule MediumNode2SubProcess{
  from a : BSCP!Tactic (a.hasSource(a.name) and a.hasTarget(a.name))
  to b : BPMN!SubProcess(name <- a.name)}

(4) rule LeafNode2Task{
  from a : BSCP!Tactic (not a.hasSource(a.name) and a.hasTarget(a.name))
  to b : BPMN!Activity(activityType <- 'Task', name <- a.name)}

(5) rule ORDecomposition_FirstOccurrence{
  from a : BSCP!ORDecomposition(BSCP!ORDecomposition.allInstances()->first())
  to b : BPMN!Activity(activityType <- 'GatewayDataBasedExclusive'),
    c : BPMN!SequenceEdge(id <- 'Left Conditional Edge'),
    d : BPMN!SequenceEdge(id <- 'Right Conditional Edge'),
    e : BPMN!Activity(activityType <- 'GatewayDataBasedExclusive'),
    f : BPMN!SequenceEdge(id <- 'Edge Closing Conditional Construction')}}

(6) rule ORDecomposition_OtherOccurrences{
  from a : BSCP!ORDecomposition(not BSCP!ORDecomposition.allInstances()->first())
  to b : BPMN!SequenceEdge(id <- 'Left Conditional Edge'),
    c : BPMN!SequenceEdge(id <- 'Right Conditional Edge')}}

(7) rule ANDDecomposition_Sequence{
  from a : BSCP!ANDDecomposition(self.type = #SequentialOrder)
  to b : BPMN!SequenceEdge(id <- 'Sequence Edge')}}

(8) rule ANDDecomposition_Parallel_FirstOccurrence{
  from a : BSCP!ANDDecomposition(self.type = #ParallelOrder and
    BSCP!ANDDecomposition.allInstances()->first())
  to b : BPMN!Activity(activityType <- 'GatewayParallel'),
    c : BPMN!SequenceEdge(id <- 'Left Parallel Edge'),
    d : BPMN!SequenceEdge(id <- 'Right Parallel Edge'),
    e : BPMN!Activity(activityType <- 'GatewayParallel'),
    f : BPMN!SequenceEdge(id <- 'Edge Closing Parallel Construction')}}

(9) rule ANDDecomposition_Parallel_OtherOccurrences{
  from a : BSCP!ANDDecomposition(self.type = #ParallelOrder and not
    BSCP!ANDDecomposition.allInstances()->first())
  to b : BPMN!SequenceEdge(id <- 'Left Parallel Edge'),
    c : BPMN!SequenceEdge(id <- 'Right Parallel Edge')}}

(10) rule SharedPhenomenon2MessagingEdge{
  from a : BSCP!SharedPhenomenon
  to b : BPMN!Activity(activityType <- 'Task', name <- 'Send'),
    BPMN!Activity(activityType <- 'Task', name <- 'Receive'),
    BPMN!SequenceEdge(id <- 'Sequence Edge'),
    BPMN!SequenceEdge(id <- 'Sequence Edge'),
    BPMN!MessagingEdge(id <- 'Messaging Edge')}}

```

*Figure B.1: The ATL Implementation of the BSCP2BPMN Model Transformations*

(\*) In terms of semantic correctness, syntactic correctness, economic efficiency, clarity, comparability, systemic design

(\*\*) In terms of content, accuracy, format, ease of use



<b>Data Collection Method</b>	<b>Model Quality(*)</b>	<b>User Satisfaction(**)</b>
Documentation Direct Observation Participant-Observation Feedback from interviews		

*Table B.1: Data Format for Case Study Findings*



# C

## BSCP2BPMN Details

Chapter 4 introduces the BSCP2BPMN model transformations, but does not provide in-depth technical details about the transformation. In order to address this gap, this technical appendix elaborates on the missing details:

1. The business user employs the BSCP editor to design a BSCP diagram (e.g., Problem Diagram B in Figure C.2) that corresponds to our transformation criteria:

- The problem diagrams should solely consists of tactics.
- Each tactic should refer to or constrain one domain of interest.
- There should be at least one shared phenomenon on each interface between domains of interest.
- The top tactic of the requirement diagram (e.g., tactic *Encourage rental extensions* in Figure C.2) gives the name of the business process in scope of the transformation.
- Control flow annotations should be consistent per group of tactic decompositions. For instance, tactic *Encourage rental extensions* (Figure C.2) has two sequential AND decompositions, tactic *Persuade airport customers* (Figure C.2) has two parallel AND decompositions, and tactic *Handle rental extensions* (Figure C.2) has two OR decompositions.

2. The BSCP editor created an XML serialization of Problem Diagram B (Figure C.2), which is displayed by Figure C.3. Some BSCP concepts that are contained in the XML serialization, are not visualized by Figure C.2, such as:

- `<relationship xsi:type="BSCP:ANDDecomposition" ... type="Sequence"/>`
- `<relationship xsi:type="BSCP:ANDDecomposition" ... type="Parallel"/>`
- `<requirement xsi:type="BSCP:Tactic" ... refersTo="..."/>`
- `<sharedPhenomenon content="x" isControlledBy="..."/>`

3. The XML serialization of Problem Diagram B (Figure C.2), corresponding to the BSCP metamodel (Chapter 4 - Figure 4.2), should be transformed into another XML serialization, corresponding to the BPMN metamodel (Figure C.4). The concept mappings between the BSCP metamodel and BPMN metamodel are given in Table 4.1 (Chapter 4). The implementation of these concept mappings is done by means of the ATL language [Ecl10a], which is given by Figure B.1 in Appendix C. When we apply these ATL mappings to the XML serialization of Problem Diagram B (Figure C.2), we obtain the generated BPMN skeleton of Figure C.5.
4. The generated BPMN skeleton of Figure C.5 contains all BPMN concepts, but lacks the correct *hierarchical* correspondence to the BPMN metamodel. For instance, Figure C.5 has `<bpmn:Bpmn-Diagram...>` and `<pools...>` as child nodes of `<xmi:XML...>`, while the BPMN metamodel prescribes that `<bpmn:BpmnDiagram...>` is the parent node of `<pools...>`. Normally, the ATL project offers functionality for ‘understanding’ the hierarchical structure of a metamodel, by means of the *Resolve Algorithm*. Firstly, this algorithm creates a traceability link between source and target elements for each application of an ATL rule. Secondly, each time a target property is initialized with a source element, the corresponding target element is computed by the resolve algorithm. So using this algorithm, ATL should ‘understand’ that `<pools...>` is a child node of `<bpmn:BpmnDiagram...>`. Unfortunately, due to the highly experimental nature of the current ATL Resolve Algorithm (version 3.0.0.v200905260429), we did not succeed in utilizing this algorithm during our BSCP2BPMN model transformation (Figure B.1 in Appendix C). Hence, the hierarchical structure should be added manually to the BPMN skeleton, which results in Figure C.6.
5. In order to complete the BPMN diagram, the business analyst takes the BPMN skeleton of Figure C.6, and adds the *necessary* cross-references, which results in Figure C.7. Each line in the BPMN skeleton has a unique identifier (iD), which is used to reference to other BPMN lines. The following cross-references are *necessary*:
  - BPMN Activity and BPMN SubProcess can have *outgoingEdges*, *incomingEdges* or *both*. For instance, `<vertices xmi:type="bpmn:Activity" name="Offer extra flight miles" iD="7" outgoingEdges="12" incomingEdges="11"/>` states that BPMN Activity ‘Offer extra flight miles’ has an incoming edge with iD = 11 and an outgoing edge with iD = 12.
  - BPMN SequenceEdge always interconnects BPMN Activities or BPMN SubProcesses, by means of a *source* and a *target* attribute. For instance, `<sequenceEdges xmi:type="bpmn:SequenceEdge" iD="12" source="7" target="10">` connects BPMN node with iD=7 (i.e., Offer extra flight miles) with BPMN node with iD=10 (i.e., the second GatewayParallel in SubProcess *Persuade airport customers*).

- BPMN MessagingEdge always interconnects BPMN Activities or BPMN SubProcesses from different BPMN Pools, by means of a *source* and a *target* attribute. Because MessagingEdges interconnect different Pools, extra annotations *outgoingMessages* and *incomingMessages* are used to send and receive the messages. An example of a full BPMN MessagingEdge structure is given by Figure C.1.

```
- <bpmn:BpmnDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
  xmlns:bpmn="http://stp.eclipse.org/bpmn" id="1">
- <pools xmi:type="bpmn:Pool" name="EU-Rent" id="2">
  <!-- Hidden details -->
  - <vertices xmi:type="bpmn:Activity" name="Send" id="16" outgoingEdges="20" incomingEdges="15">
    <outgoingMessages xmi:type="bpmn:MessagingEdge" href="#_X" />
  </vertices>
  <!-- Hidden details -->
</pools>
<messages xmi:type="bpmn:MessagingEdge" id="22" source="16" target="36" />
- <pools xmi:type="bpmn:Pool" name="EU-Rent Airport Centre" id="23">
  <!-- Hidden details -->
  - <vertices xmi:type="bpmn:Activity" name="Receive" id="36" outgoingEdges="40" incomingEdges="26">
    <incomingMessages xmi:type="bpmn:MessagingEdge" href="#_X" />
  </vertices>
  <!-- Hidden details -->
</pools>
</bpmn:BpmnDiagram>
```

Figure C.1: Example of BPMN MessagingEdge Structure (Part of Figure C.7)

6. Once the BPMN diagram is completed (Figure C.7), the visual information of the BPMN diagram still needs to be generated. Normally, one uses the visual BPMN editor to design BPMN diagrams, which results in two different but related files: a domain file that corresponds to the BPMN metamodel (e.g., FinalCopy.bpmn), and a layout file that contains the exact position of all BPMN concepts (e.g., FinalCopy.bpmn.diagram). Fortunately, the EMF project [Ecl10b] offers functionality to create a layout file based on a BPMN domain file, as demonstrated by Figure C.8.
7. The resulting BPMN diagram can be opened by double clicking on the layout file (Figure C.9). Using the visual BPMN editor, the BPMN diagram can be further refined or corrected by the business analyst.

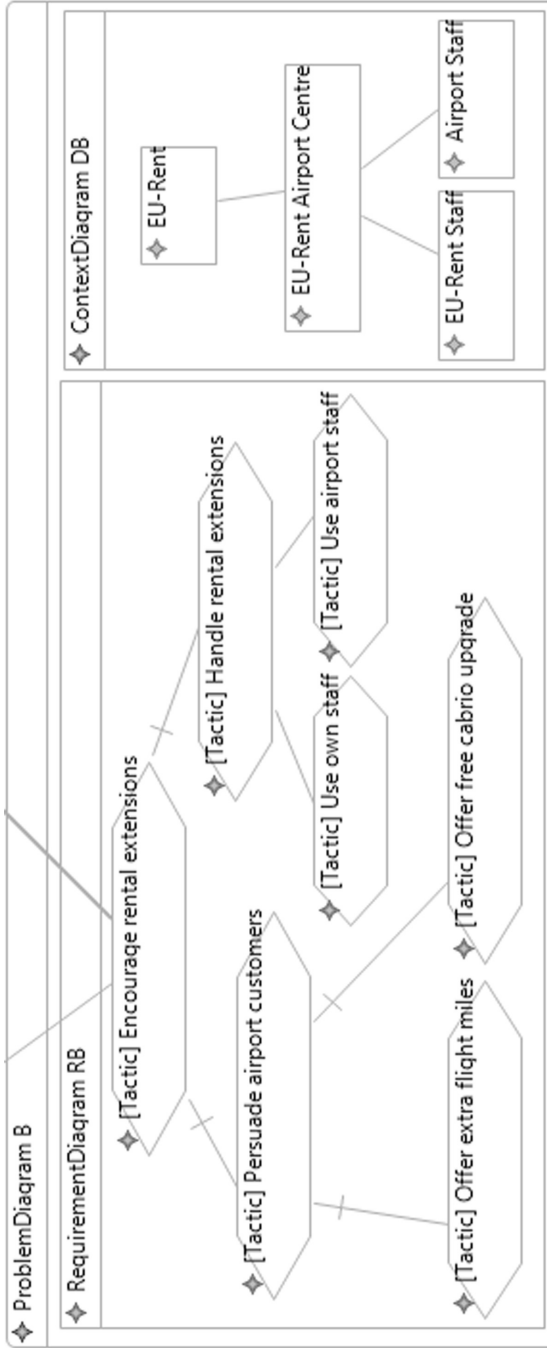


Figure C.2: Problem Diagram B

```

<?xml version="1.0" encoding="UTF-8" ?>
-<BSCP:BCSPDiagram xmi:version="2.0" xmi:ns:xmi="http://www.omg.org/XMI" xmi:ns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmins:BSCP="http://BSCP">
-<problemDiagram name="ProblemDiagram B">
-<requirementDiagram name="RequirementDiagram RB">
-<relationship xsi:type="BSCP:ANDDecomposition" hasSource="//problemDiagram.0/@requirementDiagram/@requirement.0" hasTarget="//problemDiagram.0/@requirementDiagram/@requirement.1" type="Sequence" />
-<relationship xsi:type="BSCP:ANDDecomposition" hasSource="//problemDiagram.0/@requirementDiagram/@requirement.1" hasTarget="//problemDiagram.0/@requirementDiagram/@requirement.4" type="Parallel" />
-<relationship xsi:type="BSCP:ANDDecomposition" hasSource="//problemDiagram.0/@requirementDiagram/@requirement.1" hasTarget="//problemDiagram.0/@requirementDiagram/@requirement.2" type="Parallel" />
-<relationship xsi:type="BSCP:ANDDecomposition" hasSource="//problemDiagram.0/@requirementDiagram/@requirement.4" hasTarget="//problemDiagram.0/@requirementDiagram/@requirement.5" type="Parallel" />
-<relationship xsi:type="BSCP:ORDecomposition" hasSource="//problemDiagram.0/@requirementDiagram/@requirement.4" hasTarget="//problemDiagram.0/@requirementDiagram/@requirement.6" type="Sequence" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Encourage rental extensions" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Persuade airport customers" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.0" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Offer extra flight miles" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.0" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Offer free cabrio upgrade" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.0" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Handle rental extensions" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.1" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Use own staff" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.1" />
-<requirement xsi:type="BSCP:Tactic" name="[[Tactic] Use airport staff" refersTo="//problemDiagram.0/contextDiagram/@domainOfInterest.1" />
-</requirementDiagram>
-<contextDiagram name="ContextDiagram DB">
-<interface hasSource="//problemDiagram.0/contextDiagram/@domainOfInterest.0" hasTarget="//problemDiagram.0/contextDiagram/@domainOfInterest.1" name="a" />
-<sharedPhenomenon content="x" isControlledBy="//problemDiagram.0/contextDiagram/@domainOfInterest.0" />
-<sharedPhenomenon content="y" isControlledBy="//problemDiagram.0/contextDiagram/@domainOfInterest.1" />
</interface>
<interface hasSource="//problemDiagram.0/contextDiagram/@domainOfInterest.1" hasTarget="//problemDiagram.0/contextDiagram/@domainOfInterest.2" name="b" />
<interface hasSource="//problemDiagram.0/contextDiagram/@domainOfInterest.1" hasTarget="//problemDiagram.0/contextDiagram/@domainOfInterest.3" name="c" />
<domainOfInterest name="EU-Rent" />
<domainOfInterest name="EU-Rent Airport Centre" />
<domainOfInterest name="EU-Rent Staff" />
<domainOfInterest name="Airport Staff" />
</contextDiagram>
</problemDiagram>
</BSCP:BCSPDiagram>

```

Figure C.3: XML serialization of Problem Diagram B

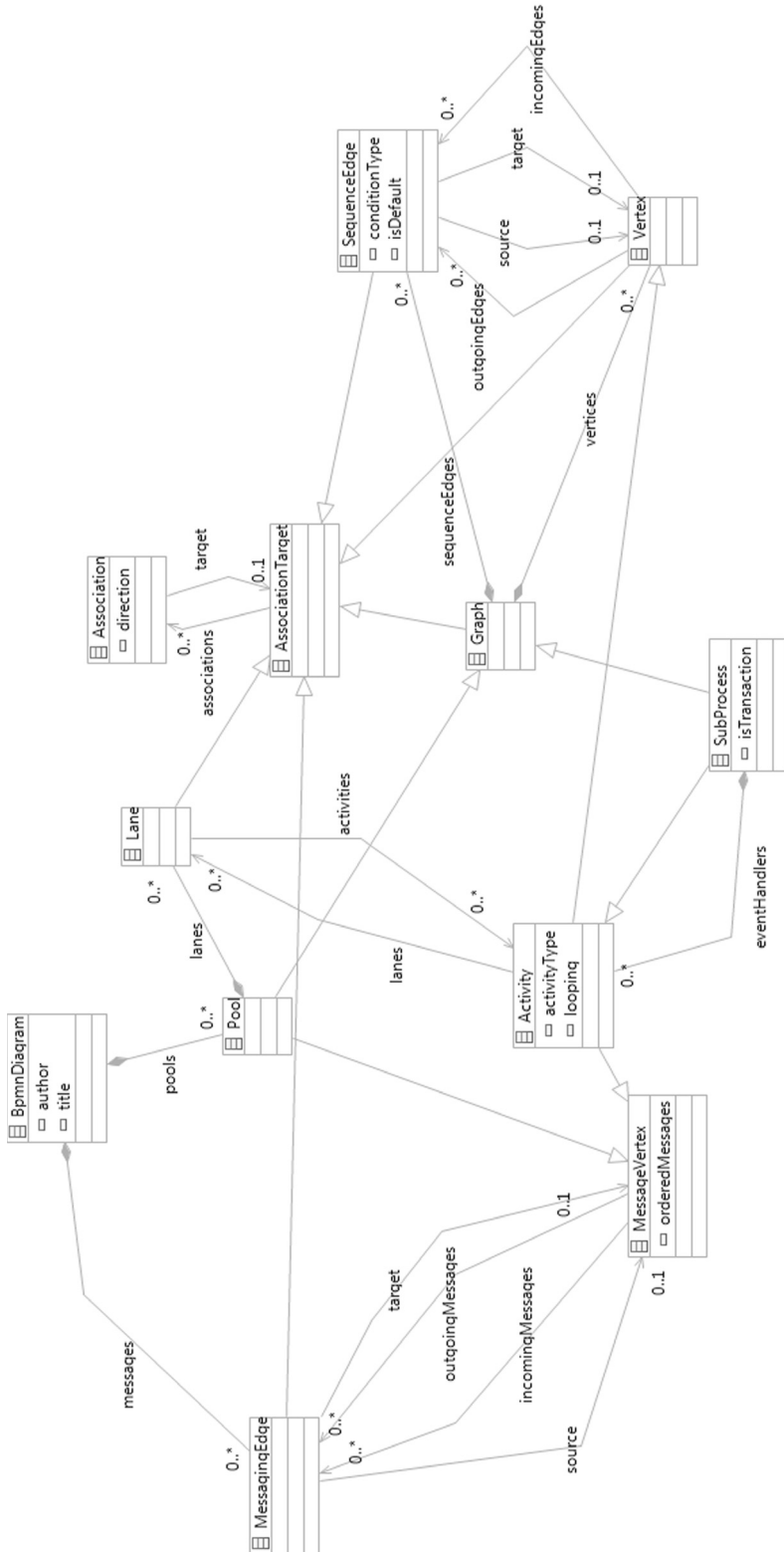


Figure C.4: The BPMN metamodel



```

<?xml version="1.0" encoding="UTF-8" ?>
- <xml:XML xml:version="2.0" xmlns:xml="http://www.omg.org/XMI" xmlns:bpmn="http://stp.eclipse.org/bpmn">
  <!-- Rule 1 -->
  <bpmn:BpmnDiagram xml:version="2.0" xmlns:xml="http://www.omg.org/XMI" xmlns:bpmn="http://stp.eclipse.org/bpmn" id="1" />
  <!-- Rule 2 -->
  <pools xml:type="bpmn:Pool" name="EU-Rent" id="2" />
  <!-- Rule 2 -->
  <vertices xml:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="3" />
  <!-- Rule 2 -->
  <vertices xml:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="4" />
  <!-- Rule 2 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="5" />
  <!-- Rule 3 -->
  <vertices xml:type="bpmn:SubProcess" name="Persuade airport customers" id="6" />
  <!-- Rule 4 -->
  <vertices xml:type="bpmn:Activity" name="Offer extra flight miles" id="7" />
  <!-- Rule 4 -->
  <vertices xml:type="bpmn:Activity" name="Offer free cabrio upgrades" id="8" />
  <!-- Rule 8 -->
  <vertices xml:type="bpmn:Activity" activityType="GatewayParallel" id="9" />
  <!-- Rule 8 -->
  <vertices xml:type="bpmn:Activity" activityType="GatewayParallel" id="10" />
  <!-- Rule 8 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="11" />
  <!-- Rule 8 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="12" />
  <!-- Rule 9 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="13" />
  <!-- Rule 9 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="14" />
  <!-- Rule 7 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="15" />
  <!-- Rule 10 -->
  <vertices xml:type="bpmn:Activity" name="Send" id="16" />
  <!-- Rule 10 -->
  <outgoingMessages xml:type="bpmn:MessagingEdge" id="17" />
  <!-- Rule 10 -->
  <vertices xml:type="bpmn:Activity" name="Receive" id="18" />
  <!-- Rule 10 -->
  <incomingMessages xml:type="bpmn:MessagingEdge" id="19" />
  <!-- Rule 10 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="20" />
  <!-- Rule 10 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="21" />
  <!-- Rule 10 -->
  <messages xml:type="bpmn:MessagingEdge" id="22" />
  <!-- Rule 2 -->
  <pools xml:type="bpmn:Pool" name="EU-Rent Airport Centre" id="23" />
  <!-- Rule 2 -->
  <vertices xml:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="24" />
  <!-- Rule 2 -->
  <vertices xml:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="25" />
  <!-- Rule 2 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="26" />
  <!-- Rule 3 -->
  <vertices xml:type="bpmn:SubProcess" name="Handle rental extensions" id="27" />
  <!-- Rule 4 -->
  <vertices xml:type="bpmn:Activity" name="Use our staff" id="28" />
  <!-- Rule 4 -->
  <vertices xml:type="bpmn:Activity" name="Use airport staff" id="29" />
  <!-- Rule 5 -->
  <vertices xml:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="30" />
  <!-- Rule 5 -->
  <vertices xml:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="31" />
  <!-- Rule 5 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="32" />
  <!-- Rule 5 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="33" />
  <!-- Rule 6 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="34" />
  <!-- Rule 6 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="35" />
  <!-- Rule 10 -->
  <vertices xml:type="bpmn:Activity" name="Receive" id="36" />
  <!-- Rule 10 -->
  <incomingMessages xml:type="bpmn:MessagingEdge" id="37" />
  <!-- Rule 10 -->
  <vertices xml:type="bpmn:Activity" name="Send" id="38" />
  <!-- Rule 10 -->
  <outgoingMessages xml:type="bpmn:MessagingEdge" id="39" />
  <!-- Rule 10 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="40" />
  <!-- Rule 10 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="41" />
  <!-- Rule 7 -->
  <sequenceEdges xml:type="bpmn:SequenceEdge" id="42" />
  <!-- Rule 10 -->
  <messages xml:type="bpmn:MessagingEdge" id="43" />
</xml:XML>

```

Figure C.5: Generated BPMN skeleton

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Rule 1 -->
- <bpmn:BpmnDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:bpmn="http://stp.eclipse.org/bpmn" id="1">
  <!-- Rule 2 -->
  - <pools xmi:type="bpmn:Pool" name="EU-Rent" id="2">
    <!-- Rule 3 -->
    <vertices xmi:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="3" />
    <!-- Rule 2 -->
    <vertices xmi:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="4" />
    <!-- Rule 2 -->
    <sequenceEdges xmi:type="bpmn:SequenceEdge" id="5" />
    <!-- Rule 3 -->
    - <vertices xmi:type="bpmn:SubProcess" name="Persuade airport customers" id="6">
      <!-- Rule 4 -->
      <vertices xmi:type="bpmn:Activity" name="Offer extra flight miles" id="7" />
      <!-- Rule 4 -->
      <vertices xmi:type="bpmn:Activity" name="Offer free cabrio upgrades" id="8" />
      <!-- Rule 8 -->
      <vertices xmi:type="bpmn:Activity" activityType="GatewayParallel" id="9" />
      <!-- Rule 8 -->
      <vertices xmi:type="bpmn:Activity" activityType="GatewayParallel" id="10" />
      <!-- Rule 8 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="11" />
      <!-- Rule 8 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="12" />
      <!-- Rule 9 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="13" />
      <!-- Rule 9 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="14" />
      <!-- Rule 3 -->
      </vertices>
      <!-- Rule 7 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="15" />
      <!-- Rule 10 -->
      - <vertices xmi:type="bpmn:Activity" name="Send" id="16">
        <!-- Rule 10 -->
        <outgoingMessages xmi:type="bpmn:MessagingEdge" id="17" />
        <!-- Rule 10 -->
        </vertices>
        <!-- Rule 10 -->
        - <vertices xmi:type="bpmn:Activity" name="Receive" id="18">
          <!-- Rule 10 -->
          <incomingMessages xmi:type="bpmn:MessagingEdge" id="19" />
          <!-- Rule 10 -->
          </vertices>
          <!-- Rule 10 -->
          <sequenceEdges xmi:type="bpmn:SequenceEdge" id="20" />
          <!-- Rule 10 -->
          <sequenceEdges xmi:type="bpmn:SequenceEdge" id="21" />
          <!-- Rule 2 -->
          </pools>
          <!-- Rule 10 -->
          <messages xmi:type="bpmn:MessagingEdge" id="22" />
          <!-- Rule 2 -->
          - <pools xmi:type="bpmn:Pool" name="EU-Rent Airport Centre" id="23">
            <!-- Rule 2 -->
            <vertices xmi:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="24" />
            <!-- Rule 2 -->
            <vertices xmi:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="25" />
            <!-- Rule 2 -->
            <sequenceEdges xmi:type="bpmn:SequenceEdge" id="26" />
            <!-- Rule 3 -->
            - <vertices xmi:type="bpmn:SubProcess" name="Handle rental extensions" id="27">
              <!-- Rule 4 -->
              <vertices xmi:type="bpmn:Activity" name="Use our staff" id="28" />
              <!-- Rule 4 -->
              <vertices xmi:type="bpmn:Activity" name="Use airport staff" id="29" />
              <!-- Rule 5 -->
              <vertices xmi:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="30" />
              <!-- Rule 5 -->
              <vertices xmi:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="31" />
              <!-- Rule 5 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="32" />
              <!-- Rule 5 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="33" />
              <!-- Rule 6 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="34" />
              <!-- Rule 6 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="35" />
              <!-- Rule 3 -->
              </vertices>
              <!-- Rule 10 -->
              - <vertices xmi:type="bpmn:Activity" name="Receive" id="36">
                <!-- Rule 10 -->
                <incomingMessages xmi:type="bpmn:MessagingEdge" id="37" />
                <!-- Rule 10 -->
                </vertices>
                <!-- Rule 10 -->
                <vertices xmi:type="bpmn:Activity" name="Send" id="38">
                  <!-- Rule 10 -->
                  <outgoingMessages xmi:type="bpmn:MessagingEdge" id="39" />
                  <!-- Rule 10 -->
                  </vertices>
                  <!-- Rule 10 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="40" />
                  <!-- Rule 10 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="41" />
                  <!-- Rule 7 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="42" />
                  <!-- Rule 2 -->
                  </pools>
                  <!-- Rule 10 -->
                  <messages xmi:type="bpmn:MessagingEdge" id="43" />
                  <!-- Rule 1 -->
                </bpmn:BpmnDiagram>

```

Figure C.6: Added Hierarchical Structure to BPMN skeleton

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Rule 1 -->
- <bpmn:BpmnDiagram xmlns:xmi="http://www.omg.org/XMI" xmlns:bpmn="http://stp.eclipse.org/bpmn" id="1">
  <!-- Rule 2 -->
  - <pools xmi:type="bpmn:Pool" name="EU-Rent" id="2">
    <!-- Rule 2 -->
    <vertices xmi:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="3" outgoingEdges="5" />
    <!-- Rule 2 -->
    <vertices xmi:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="4" incomingEdges="21" />
    <!-- Rule 2 -->
    <sequenceEdges xmi:type="bpmn:SequenceEdge" id="5" source="3" target="6" />
    <!-- Rule 3 -->
    - <vertices xmi:type="bpmn:SubProcess" name="Persuade airport customers" id="6" outgoingEdges="15" incomingEdges="5">
      <!-- Rule 4 -->
      <vertices xmi:type="bpmn:Activity" name="Offer extra flight miles" id="7" outgoingEdges="12" incomingEdges="11" />
      <!-- Rule 4 -->
      <vertices xmi:type="bpmn:Activity" name="Offer free cabrio upgrades" id="8" outgoingEdges="14" incomingEdges="13" />
      <!-- Rule 8 -->
      <vertices xmi:type="bpmn:Activity" activityType="GatewayParallel" id="9" outgoingEdges="11 13" />
      <!-- Rule 8 -->
      <vertices xmi:type="bpmn:Activity" activityType="GatewayParallel" id="10" incomingEdges="12 14" />
      <!-- Rule 8 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="11" source="9" target="7" />
      <!-- Rule 8 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="12" source="7" target="10" />
      <!-- Rule 9 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="13" source="9" target="8" />
      <!-- Rule 9 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="14" source="8" target="10" />
      <!-- Rule 3 -->
      </vertices>
      <!-- Rule 7 -->
      <sequenceEdges xmi:type="bpmn:SequenceEdge" id="15" source="6" target="16" />
      <!-- Rule 10 -->
      - <vertices xmi:type="bpmn:Activity" name="Send" id="16" outgoingEdges="20" incomingEdges="15">
        <!-- Rule 10 -->
        <outgoingMessages xmi:type="bpmn:MessagingEdge" href="#_X" />
        <!-- Rule 10 -->
        </vertices>
        <!-- Rule 10 -->
        - <vertices xmi:type="bpmn:Activity" name="Receive" id="18" outgoingEdges="21" incomingEdges="20">
          <!-- Rule 10 -->
          <incomingMessages xmi:type="bpmn:MessagingEdge" href="#_Y" />
          <!-- Rule 10 -->
          </vertices>
          <!-- Rule 10 -->
          <sequenceEdges xmi:type="bpmn:SequenceEdge" id="20" source="16" target="18" />
          <!-- Rule 10 -->
          <sequenceEdges xmi:type="bpmn:SequenceEdge" id="21" source="18" target="4" />
          <!-- Rule 2 -->
          </pools>
          <!-- Rule 10 -->
          <messages xmi:type="bpmn:MessagingEdge" id="22" source="16" target="36" />
          <!-- Rule 2 -->
          - <pools xmi:type="bpmn:Pool" name="EU-Rent Airport Centre" id="23">
            <!-- Rule 2 -->
            <vertices xmi:type="bpmn:Activity" name="StartEvent" activityType="EventStartEmpty" id="24" outgoingEdges="26" />
            <!-- Rule 2 -->
            <vertices xmi:type="bpmn:Activity" name="EndEvent" activityType="EventEndEmpty" id="25" incomingEdges="42" />
            <!-- Rule 2 -->
            <sequenceEdges xmi:type="bpmn:SequenceEdge" id="26" source="24" target="36" />
            <!-- Rule 3 -->
            - <vertices xmi:type="bpmn:SubProcess" name="Handle rental extensions" id="27" outgoingEdges="41" incomingEdges="40">
              <!-- Rule 4 -->
              <vertices xmi:type="bpmn:Activity" name="Use our staff" id="28" outgoingEdges="33" incomingEdges="32" />
              <!-- Rule 4 -->
              <vertices xmi:type="bpmn:Activity" name="Use airport staff" id="29" outgoingEdges="35" incomingEdges="34" />
              <!-- Rule 5 -->
              <vertices xmi:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="30" outgoingEdges="32 34" />
              <!-- Rule 5 -->
              <vertices xmi:type="bpmn:Activity" activityType="GatewayDataBasedExclusive" id="31" incomingEdges="33 35" />
              <!-- Rule 5 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="32" source="30" target="28" />
              <!-- Rule 5 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="33" source="28" target="31" />
              <!-- Rule 6 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="34" source="30" target="29" />
              <!-- Rule 6 -->
              <sequenceEdges xmi:type="bpmn:SequenceEdge" id="35" source="29" target="31" />
              <!-- Rule 3 -->
              </vertices>
              <!-- Rule 10 -->
              - <vertices xmi:type="bpmn:Activity" name="Receive" id="36" outgoingEdges="40" incomingEdges="26">
                <!-- Rule 10 -->
                <incomingMessages xmi:type="bpmn:MessagingEdge" href="#_X" />
                <!-- Rule 10 -->
                </vertices>
                <!-- Rule 10 -->
                - <vertices xmi:type="bpmn:Activity" name="Send" id="38" outgoingEdges="42" incomingEdges="41">
                  <!-- Rule 10 -->
                  <outgoingMessages xmi:type="bpmn:MessagingEdge" href="#_Y" />
                  <!-- Rule 10 -->
                  </vertices>
                  <!-- Rule 10 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="40" source="36" target="27" />
                  <!-- Rule 10 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="41" source="27" target="38" />
                  <!-- Rule 7 -->
                  <sequenceEdges xmi:type="bpmn:SequenceEdge" id="42" source="38" target="25" />
                  <!-- Rule 2 -->
                  </pools>
                  <!-- Rule 10 -->
                  <messages xmi:type="bpmn:MessagingEdge" id="43" source="38" target="18" />
                  <!-- Rule 1 -->
                  </bpmn:BpmnDiagram>
                
```

Figure C.7: Completed BPMN diagram

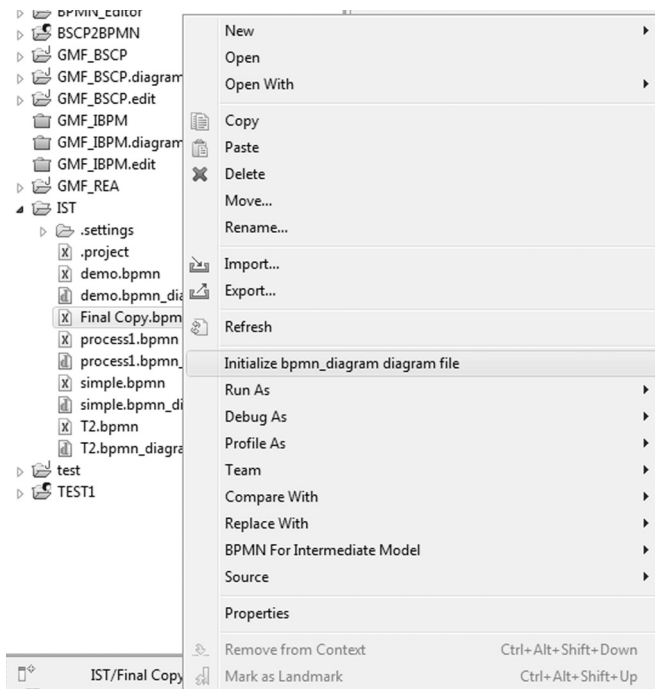


Figure C.8: Generate diagram file with EMF

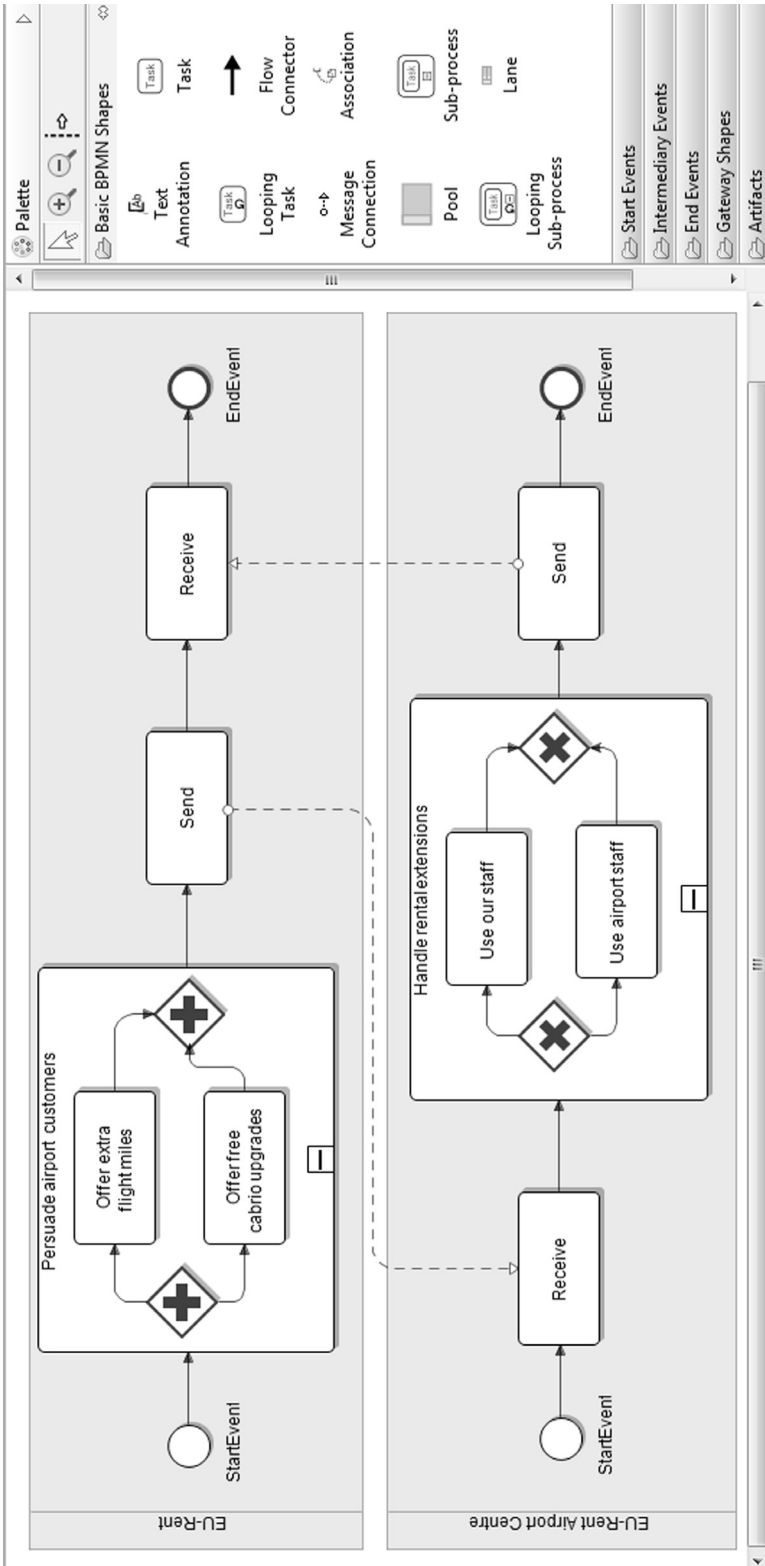


Figure C.9: Completed BPMN Diagram (visualized in BPMN Editor)



# D

## Graphical Editor Details

This appendix will provide more information about the usage of EMF and GMF to implement a graphical editor in Eclipse. In chapter 3, we used this technique to implement the PFT editor, and in chapter 4, we created the B-SCP editor. The source files of the PFT editor are available at <http://www.managementinformation.ugent.be/PFT2BPMO/>, and the source files of the B-SCP editor can be found at <http://code.google.com/p/bscp2bpnm/>. The process of creating this editor was exactly the same (based on the applicable metamodel), so for the purpose of illustration, we will select the B-SCP source files. Overall, the process for creating a graphical editor with EMF and GMF consists of the following steps:

1. *Create an Ecore metamodel*: employ the Ecore concepts to create a new metamodel (Figure D.1). First start with creating all EClasses, and then connect these EClasses by means of Associations, Aggregations and Generalizations. Make sure that each EClass has the required EAttributes, such as the EAttribute *name* of EClass *ProblemDiagram*.
2. *Create a new GMF project based on the Ecore metamodel*: Using the Eclipse creation wizard, you can select a new GMF project, which will provide you with a new GMF Dashboard (Figure D.2). On the GMF Dashboard, click on *select* to search for the Ecore metamodel that you want to use (e.g., the BSCP Ecore metamodel is selected in Figure D.3).
3. *Derive a domain gen model*: On the GMF Dashboard, select the option to *derive* a domain gen model from the specified domain model (e.g., the BSCP Ecore metamodel). In the popup window, choose *Ecore Model (CDO Native)*, then load the domain model (e.g., BSCP.ecore) and select the correct root package (e.g., root package BSCP from file BSCP.ecore). The GMF Dashboard will be updated to show the included domain gen model (Figure D.4). Then, right click on the domain gen top model (e.g., the BSCP top node), and select *Generate Model Code* and *Generate Edit Code*.

4. *Derive a graphical def model:* On the GMF Dashboard, select the option to *derive* the graphical def model (Figure D.5). Once the graphical def model is automatically derived, check whether the correct nodes, connections, compartments and diagram labels have been created.
5. *Derive a tooling def model:* On the GMF Dashboard, select the option to *derive* the tooling def model (Figure D.6). Make sure that there is a corresponding creation tool (e.g., Creation Tool for Problem Diagram) for every concept in the domain model (e.g., ProblemDiagram from BSCP Ecore metamodel).
6. *Create a mapping model:* On the GMF Dashboard, select the option to *combine* the domain model, graphical def model and tooling def model. This is an important step during the GMF process, so make sure that the correct mappings are applied (Figure D.7).
7. *Create a generator model:* From the Eclipse folder structure, right click on the mapping model to select *Create generator model* (Figure D.8). Give an appropriate name (e.g., BSCP.gmfgen), load the mapping model (e.g., BSCP.gmfmap), load the previous gen model (e.g., BSCP.genmodel), and do not change the default settings of the transformation options menu. As a result, a generator model is created (Figure D.9).
8. *Generate the editor code:* From the Eclipse folder structure, right click on the generator model to select *Generate diagram code* (Figure D.10). Wait for the editor code to be generated.
9. *Run the editor:* When the editor code is generated (e.g., adding a new Eclipse project BSCP.Editor.diagram in the Eclipse Project list), go to menu *Run*, and select *Run Configurations*. Click on *new* to add a new Eclipse application (Figure D.11), and click on *Run* to execute the editor. A new Eclipse session will start, which allows you to create a new BSCP model by selecting *file, new, example* (Figure D.12). Finally, you can use the editor to model BSCP concepts (Figure D.13).





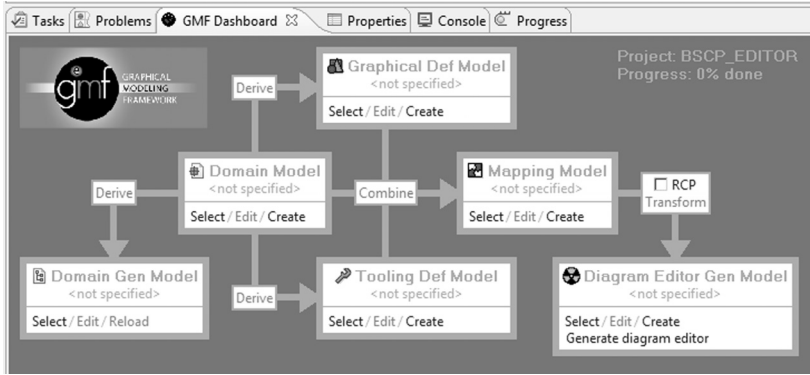


Figure D.2: New GMF Project

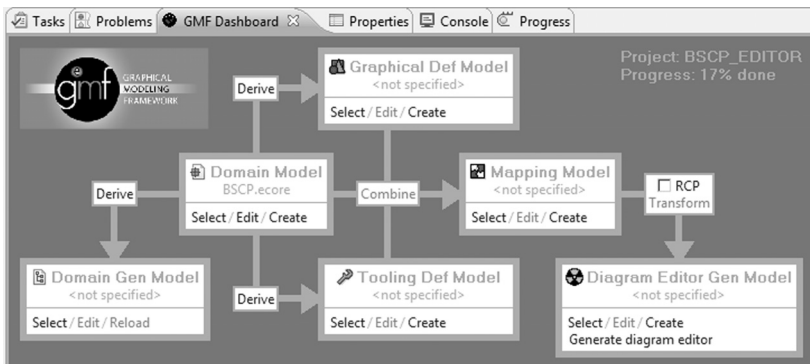


Figure D.3: BSCP Ecore Metamodel included in GMF Project

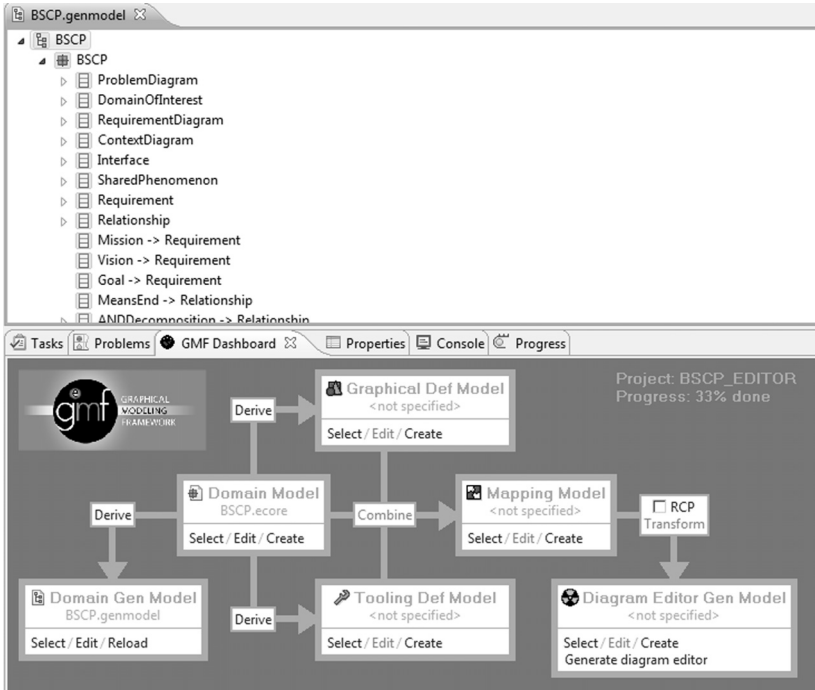


Figure D.4: BSCP Domain Gen Model included in GMF Project

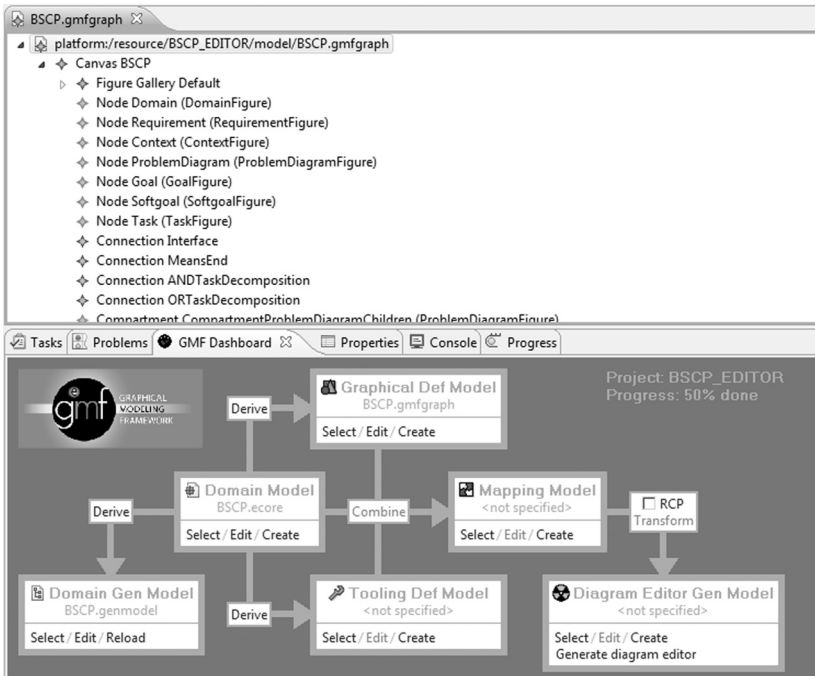


Figure D.5: BSCP Graphical Def Model included in GMF Project

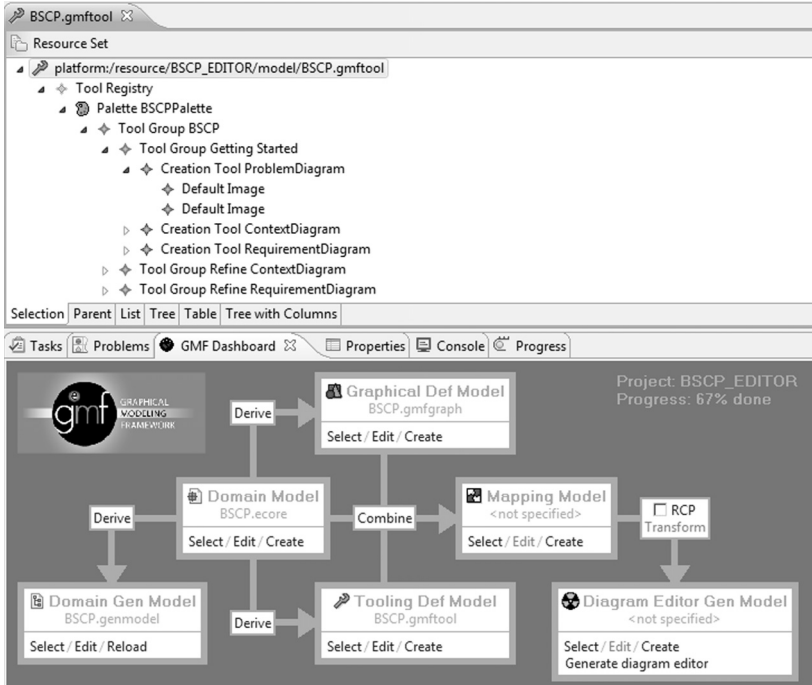


Figure D.6: BSCP Tooling Def Model included in GMF Project

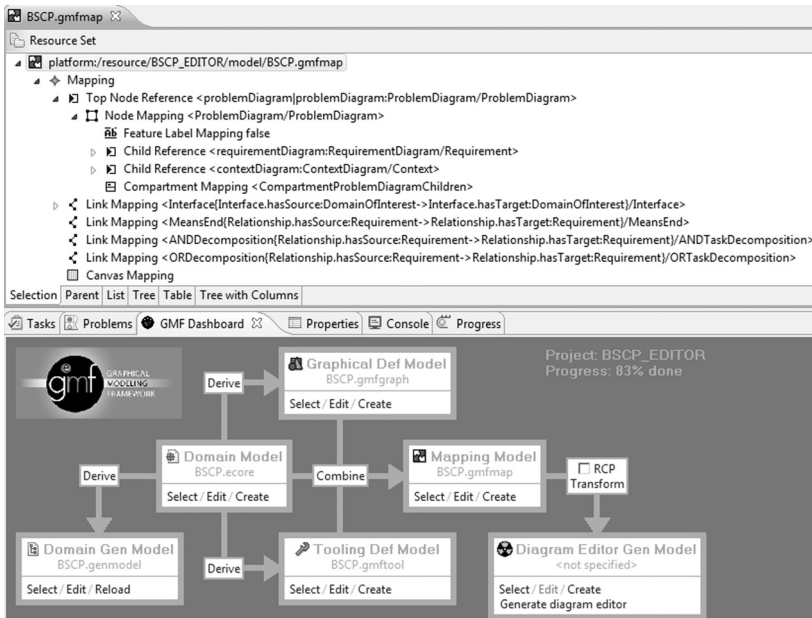


Figure D.7: BSCP Mapping Model included in GMF Project

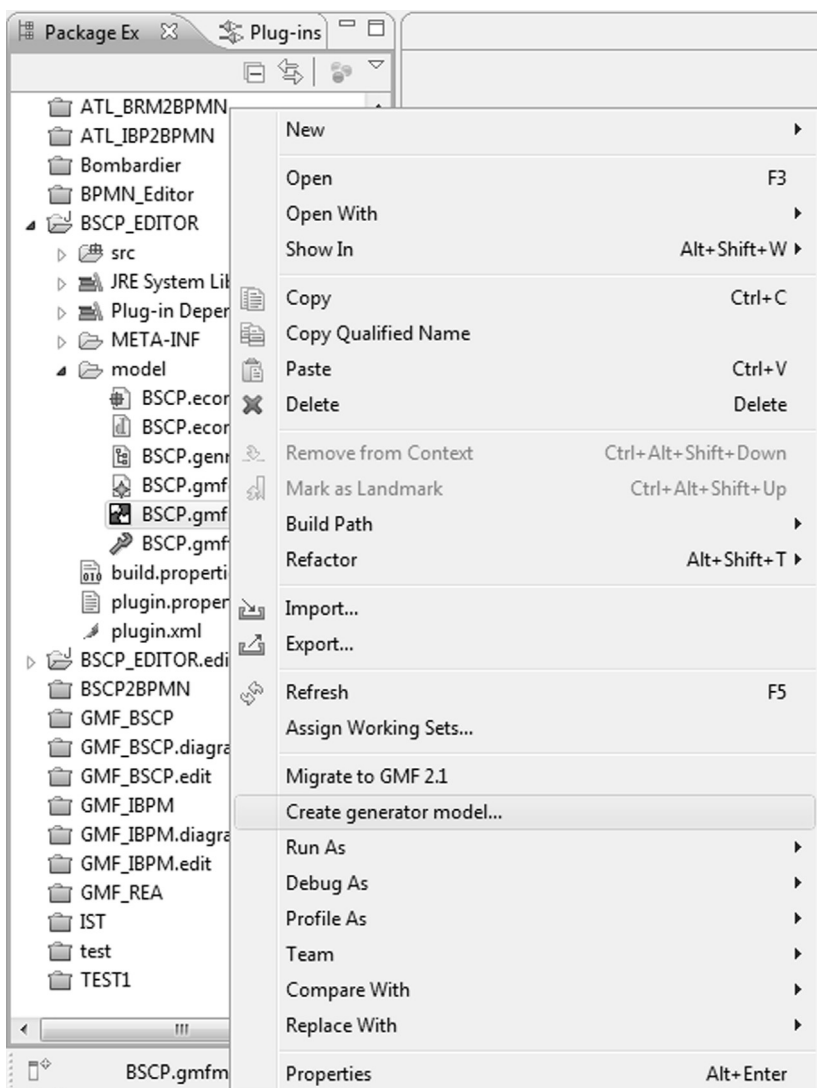


Figure D.8: Menu to Create BSCP Generator Model

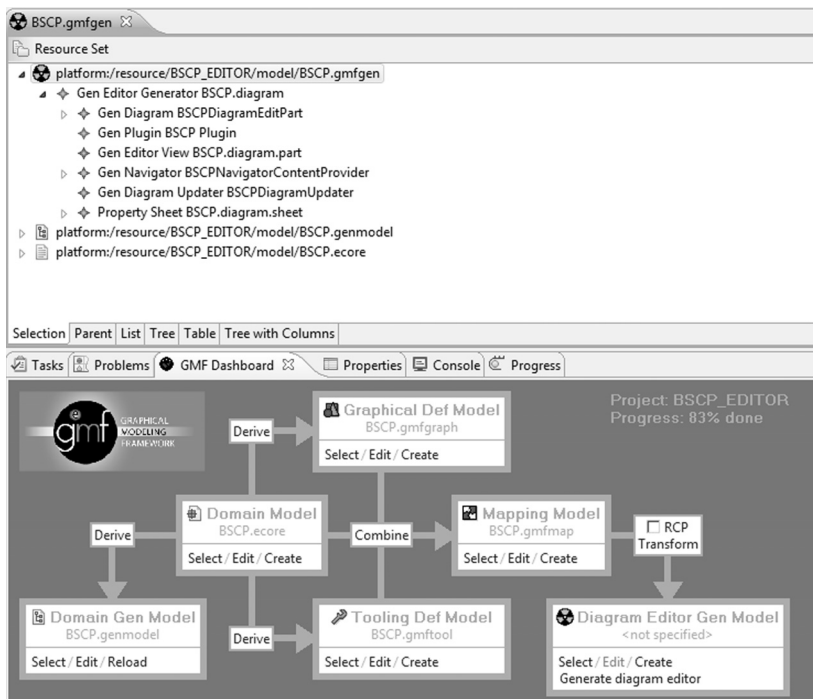


Figure D.9: BSCP Generator Model included in GMF Project



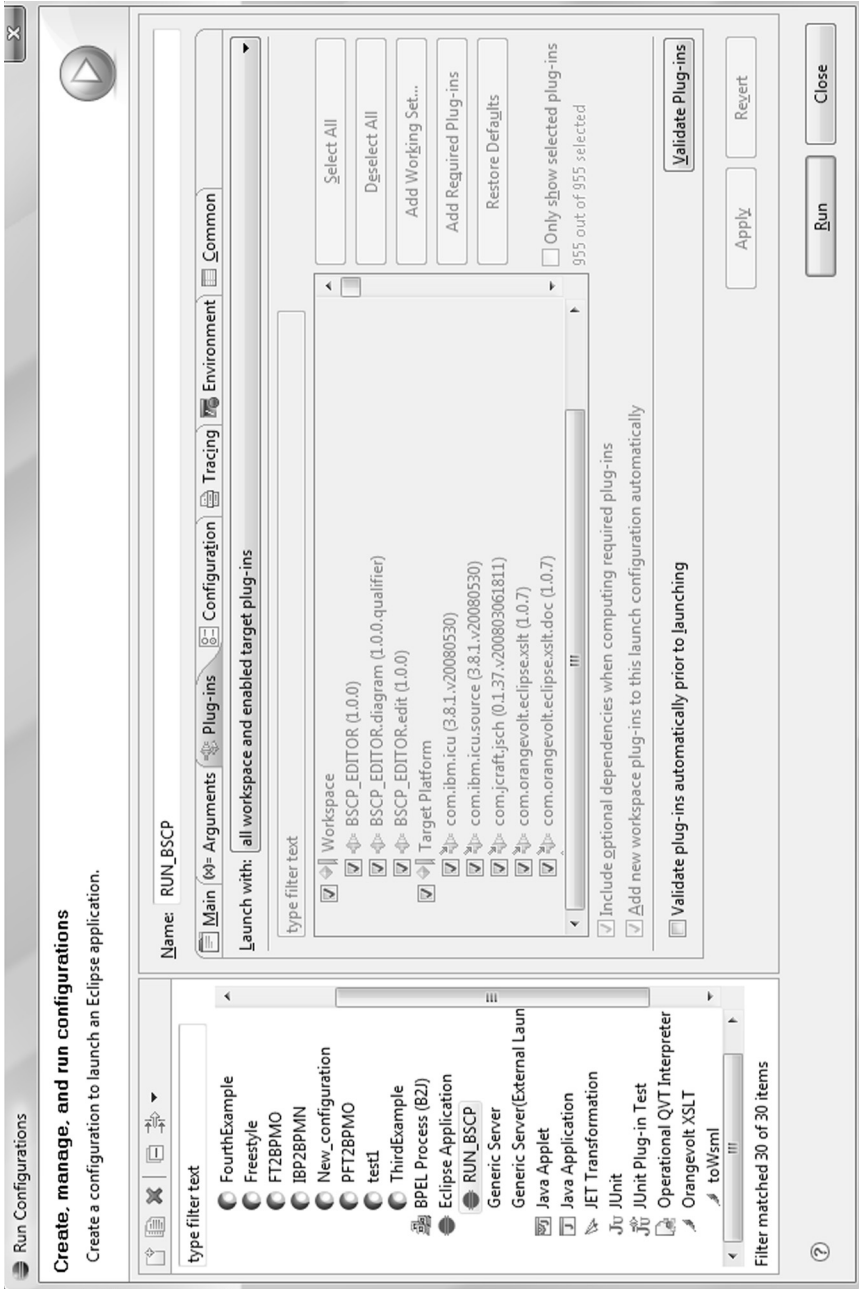


Figure D.11: Run Configuration for BSCP Editor



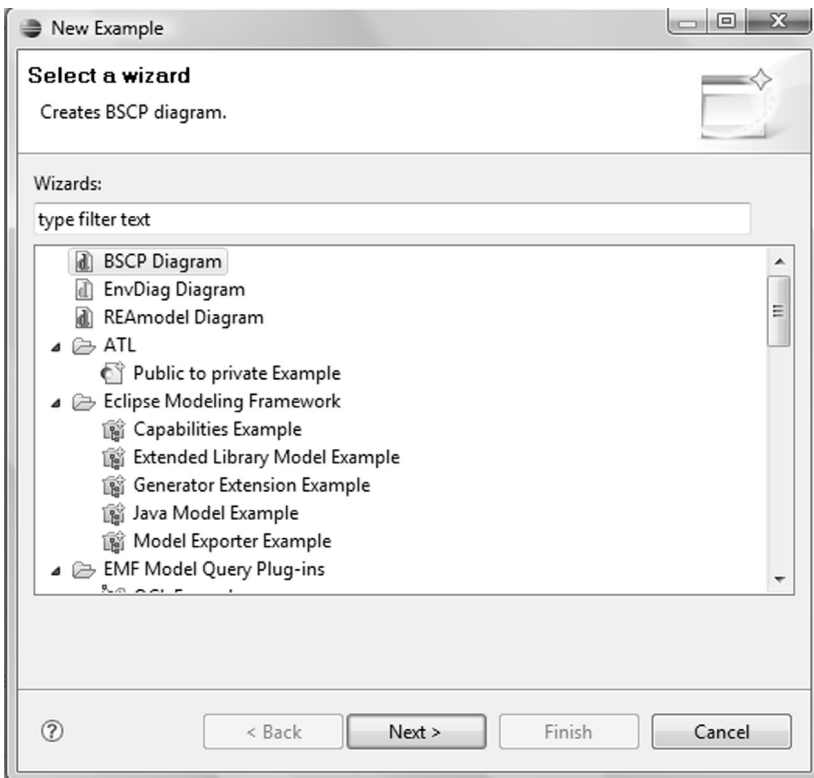


Figure D.12: Select BSCP diagram with Wizard

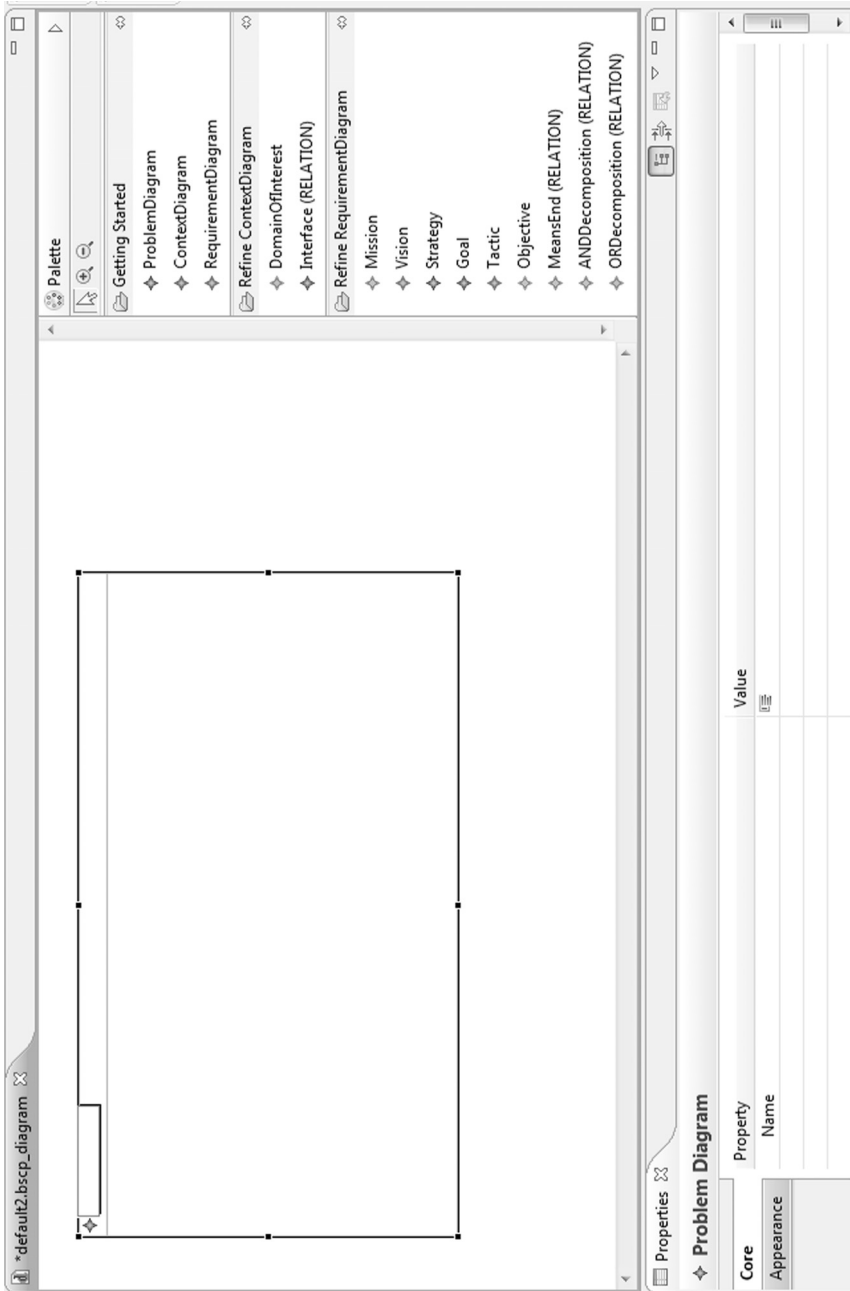


Figure D.13: Use the BSCP Editor



