The Evaluation of Business Process Modeling Techniques

Bart-Jan Hommes
STELLENINGEN

behorende bij het proefschrift
"The Evaluation of Business Process Modeling Techniques"
van Bart-Jan Hommes
26 januari 2004

I.

Het vergelijken van theorieën met een referentie-ontologie is een nuttig instrument om verschillen tussen die theorieën aan te tonen. Hoewel het voor dit doel zeer vaak wordt toegepast, is dit instrument niet geschikt om objectief de tekortkomingen of overbodigheden van een theorie aan te tonen (dit proefschrift).

II.

Technieken voor het modelleren van bedrijfsprocessen schrijven in het algemeen wel voor hoe (structuren van) elementaire transities binnen een systeem kunnen worden gemodeleerd, maar niet tot welke systeemcategorie deze transities dienen te behoren (dit proefschrift).

III.

Modelleertechnieken dienen de modelleur niet alleen in staat te stellen datgene dat in de werkelijkheid kan voorkomen te modelleren, ze dienen hem of haar er ook van te weerhouden datgene te modelleren dat niet in de werkelijkheid voor kan komen (dit proefschrift).

IV.

In het werk van Kuhn (1962) is het lastig verschil te maken tussen de begrippen paradigma en theorie. Dit leidt bij ontologische evaluaties (zoals besproken in hoofdstuk 3 van dit proefschrift) vaak tot verwarring. Een mogelijke oplossing hiervoor is het begrip paradigma als een rolbegrip te zien: Een theorie is een paradigma indien zij gebruikt wordt als kader waarbinnen andere theorieën met elkaar vergeleken kunnen worden.


V.

Het gebruik van non-deterministische modelleertechnieken in het onderwijs heeft als consequentie dat bij het tentamineren een objectieve maat voor het beoordelen van de resultaten ontbreekt.
VI.
Daar waar axiomata worden verheven tot onaantastbare waarheden, wordt wetenschappelijke vooruitgang belemmerd.

VII.
Het streven om computers als mensen te laten denken komt voor informatici binnen handbereik doordat zij zelf steeds meer als computers gaan denken.

VIII.
Het streven naar een alles verklarende theorie (bijvoorbeeld het streven naar een uniforme modelleertaal voor bedrijfsprocessen) hindert wetenschappelijke vooruitgang. Juist bij voldoende diversiteit kan groei van kennis door middel van evolutie plaatsvinden.

IX.
Door het sleutelen aan een oude auto leert men aan den lijve dat vooronderstellingen ten aanzien van de aard van een probleem een obstakel vormen voor het oplossen ervan. Derhalve zou een practicum sleutelen niet misstaan in het curriculum van een academische opleiding.

X.
Ten aanzien van de voorspellende waarde van inductieve theorieën geldt het voorbehoud dat reeds behaalde resultaten geen garantie zijn voor de toekomst.

Deze stellingen worden verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor, Prof.dr.ir. J.L.G. Dietz.
The Evaluation of Business Process Modeling Techniques
The Evaluation of Business Process Modeling Techniques

Evaluatie van technieken voor het modelleren van bedrijfsprocessen

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 26 januari 2004, om 15:30 uur

door

Lambertus Johannes HOMMES

informatica ingenieur

geboren te Oud-Beijerland
Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. J.L.G. Dietz

Samenstelling promotiecommissie:

Rector magnificus
Prof.dr.ir. J.L.G. Dietz
Prof.dr. V.E. van Reijswoud
Prof.dr. K.M. van Hee
Prof.dr.ir. W.M.P. van der Aalst
Prof.dr. R.W. Wagenaar
Prof.dr.ir. A. Verbraeck

voorzitter
Technische Universiteit Delft, promotor
Uganda Martyrs University
Technische Universiteit Eindhoven
Technische Universiteit Eindhoven
Technische Universiteit Delft
University of Maryland

Distributed by

Bart-Jan Hommes
W.Vrijlandtstraat 39
3262GN OUD-BEIJERLAND
The Netherlands

Cover design by

Bart Andeweg, Chimedia
www.chimedia.nl


© 2004 Bart-Jan Hommes

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopying, recording or otherwise, without the written permission of the author.
ACKNOWLEDGEMENTS

There is no process modeling technique capable of capturing the process of writing a Ph.D. thesis. Although colleagues presented me their descriptive models, more and more I realized that the aim of doing Ph.D research is not so much to execute these models as efficient and effective as possible, but more that it is the experience gained by carrying out each and every insecure step that at the end determines the quality of the result.

I am indebted to many people. I would like to thank Victor van Reijswoud for his support during the initiation of this research. Victor inspired me and taught me the principles of doing scientific research for which I am very grateful. Furthermore I would like to thank Jan Dietz for his guidance and support during the ‘execution and result phase’ of this research. Many thanks to my colleagues, in particular to Hans Mulder and Paul Mallens for their interest in my work. I would like to express my gratitude also for the technical support group, for always being at my back and call.

I owe gratitude to my family and friends. In particular I thank my parents for their unconditional help and support during difficult times and for putting up with my absence during busy times.

Oud-Beijerland, December 2003

Bart-Jan Hommes
CONTENTS

ACKNOWLEDGEMENTS ................................................................. I

CONTENTS .................................................................................. VII

PART I: Introduction

1 INTRODUCTION ......................................................................... 3
  1.1 Introduction .............................................................................. 3
  1.2 Research Objective and Questions ........................................... 5
  1.3 Philosophical Foundation and Research Methodology .............. 6
  1.4 Outline of this thesis ............................................................... 9

2 THE CONCEPTUAL MODELING OF BUSINESS PROCESSES ........... 12
  2.1 Introduction .............................................................................. 12
  2.2 Conceptual Modeling ............................................................ 12
  2.3 Conceptual Modeling Methods and Techniques ....................... 20
  2.4 Conceptual Business Processes Modeling ............................... 22
  2.5 Commonly Used Business Process Modeling Techniques .......... 23
  2.6 Conclusions ............................................................................ 26

3 EXISTING APPROACHES FOR EVALUATION ............................. 27
  3.1 Introduction .............................................................................. 27
  3.2 Quality Criteria and Frameworks ............................................ 27
  3.3 Perspectives in Modeling ........................................................ 37
  3.4 Analysis of Concepts and Conceptual Structures ................... 41
  3.5 Quality Measurement ............................................................. 52
  3.6 Conclusions ............................................................................ 57
# PART II: Proposition

## 4 A PROPOSAL FOR UNDERSTANDING THE QUALITY OF BUSINESS PROCESS MODELING TECHNIQUES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>63</td>
</tr>
<tr>
<td>4.2 Modeling Quality</td>
<td>64</td>
</tr>
<tr>
<td>4.3 Quality of Models</td>
<td>66</td>
</tr>
<tr>
<td>4.4 Quality of Modeling Techniques: Assuring Correctness</td>
<td>76</td>
</tr>
<tr>
<td>4.5 Quality of Modeling Techniques: Assuring Usefulness</td>
<td>82</td>
</tr>
<tr>
<td>4.6 Conclusions</td>
<td>91</td>
</tr>
</tbody>
</table>

## 5 A PROPOSAL FOR CAPTURING BUSINESS PROCESS MODELING TECHNIQUES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>93</td>
</tr>
<tr>
<td>5.2 Describing Modeling Techniques</td>
<td>93</td>
</tr>
<tr>
<td>5.3 Formalization of the Description of Modeling Techniques</td>
<td>105</td>
</tr>
<tr>
<td>5.4 Describing Modeling Methods</td>
<td>109</td>
</tr>
<tr>
<td>5.5 Formalization of the Description of Methods</td>
<td>113</td>
</tr>
<tr>
<td>5.6 A Complete Description of Modeling Techniques</td>
<td>114</td>
</tr>
<tr>
<td>5.7 Conclusions</td>
<td>115</td>
</tr>
</tbody>
</table>

## 6 A PROPOSAL FOR MEASURING THE QUALITY OF BUSINESS PROCESS MODELING TECHNIQUES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>117</td>
</tr>
<tr>
<td>6.2 Kinds of Metrics</td>
<td>117</td>
</tr>
<tr>
<td>6.3 Evaluating Modeling Techniques: Assuring Correctness</td>
<td>120</td>
</tr>
<tr>
<td>6.4 Evaluating Modeling Techniques: Assuring Usefulness</td>
<td>131</td>
</tr>
<tr>
<td>6.5 Conclusions</td>
<td>132</td>
</tr>
</tbody>
</table>
PART III: Application & Validation

7 APPLICATION AND VALIDATION OF THE
C-ME DESCRIPTION LANGUAGE ........................................ 137
7.1 Introduction ..................................................................... 137
7.2 Event Driven Process Chains ............................................. 138
7.3 Dynamic Essential Modeling of Organizations ................. 148
7.4 Petri-net Based Modeling ................................................ 161
7.5 Business Modeling With UML .......................................... 172
7.6 Conclusions ................................................................... 186

8 APPLICATION AND VALIDATION OF THE
Q-ME MEASUREMENT SCHEME .............................................. 189
8.1 Introduction ..................................................................... 189
8.2 Application of the Metrics Assuring Correctness ................. 190
8.3 Application of the Metrics Assuring Usefulness .................. 202
8.4 Empirical Study on the Interpretation of the Proposed Metrics 217
8.5 Conclusions ................................................................... 232

9 CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH .......... 235
9.1 Introduction ..................................................................... 235
9.2 Answers to Research Questions ........................................ 235
9.3 Q-Me: An Integrated Approach ....................................... 236
9.4 Application and Validation ............................................. 237
9.5 Guidelines for Qualitative Modeling Techniques ................. 238
9.6 Directions for Further Research ...................................... 242

LITERATURE REFERENCES .................................................. 245

APPENDIX A: DETAILED TECHNIQUE DESCRIPTIONS ..................... 255

SAMENVATTING ................................................................. 263

CURRICULUM VITAE ........................................................... 267
Part I

Introduction
1

INTRODUCTION

1.1 Introduction

Business Process Modeling (BPM) is becoming increasingly popular. Both experts in the field of Information and Communication Technology (ICT) and in the field of Business Engineering have come to the conclusion that successful (re)engineering of the involved systems starts with a thorough understanding of the business processes of an organization. Conceptual modeling of business processes is deployed on a large scale to facilitate various purposes (Figure 1-1).

![Figure 1-1: Business Process Modeling Purposes (Hommes et al, 2001a)]

Business Process Redesign (BPR), as advocated by Hammer and Champy (1993) aims at radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed. In Activity Based Costing (ABC), a cost management methodology proposed by Caplan and Atkinson (1989), the business process model is used as a basis for calculating the direct costs of products and services. Both ISO certification and Total Quality Management (TQM) focus on the Quality System of an organization, requiring a model of a business's organizational structure, responsibilities, procedures, processes and resources. Furthermore, business process models are used to carry out discrete event simulations (de Vreede et al, 1998).
Another important purpose of business process modeling is its use as a tool for requirements engineering. Many businesses face the problem to support their processes by new technology. The first step in these projects is to understand how their processes work, to understand that what the to be developed technology will be going to support. The first step is then to understand the business processes that are going to be supported by the new technology in such a way that it is helpful for the alignment of the technology later on. Disciplines such as Information Systems Development, Workflow Automation, the selection of so-called Commercially Of The Shell (COTS) software and Enterprise Resource Planning (ERP) utilize business process modeling to understand the business processes that are supported by the technology that is introduced in the organization.

The increasing popularity of business process modeling results in a rapidly growing number of modeling techniques and tools. An overview of Kettinger et al (1997) identifies numerous techniques and tools for the purpose of BPR, TQM, Simulation, ABC and ISO certification. Dommelen et al (1999) published an overview of the 168 modeling tools available in the Netherlands. Their purpose varies from the development of workflow management systems to support processes (e.g. Protos, Petrinet+) to the analysis (Protos, Testbed) and simulation (B-Wise, Arena, Expect) of processes. As part of the initial phase of this research, several reviews of tools available on the Internet have been merged, resulting in an overview of approximately 350 modeling tools, all claiming to support 'effective', 'comprehensible', 'compact', 'suitable' etc. conceptual business modeling.

In fact, techniques are seldom tested on the above-mentioned claims, since little research has been carried out regarding the quality of techniques for conceptual business modeling and the tools that support these techniques. Existing frameworks for evaluating quality, focus on the quality of software and information systems modeling techniques, rather than on business process modeling techniques. Apart from the fact that these frameworks do not have a business focus, there is criticism about the 'vagueness' of quality properties and the lack of operationalization (Gillies, 1992, Lindland et al, 1994).

The lack of appropriate means to assess (evaluate) the quality of this rapidly growing number of business modeling techniques and the dominant role these techniques and tools can have in for instance Business Process Reengineering, ERP system implementation, Total Quality Management and Workflow Automation, justifies the development of a conceptual framework for understanding and evaluating the quality of these techniques.

Such a framework should aid not only the evaluation of techniques, but also the comparison of techniques, the selection of the appropriate technique for a certain situation and the (re)design of techniques (Method Engineering).
1.2 Research Objective and Questions

As concluded in the previous section, there is a need for a framework for understanding and evaluating the quality of business process modeling techniques. The aim of the research presented in this thesis is to develop such a framework (c.f. Hommes, 1999). This is summarized in the following research objective:

<table>
<thead>
<tr>
<th>Research Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>To develop and validate a framework for understanding and measuring the quality of techniques for conceptual business process modeling</td>
</tr>
</tbody>
</table>

In order to accomplish this objective, knowledge of three main research areas is to be acquired. These disciplines are:

- **Conceptual modeling techniques** - Techniques for constructing abstract, simplified views of a certain phenomenon in (a possible) reality that we wish to represent for some purpose.

- **Business Process Modeling** - The modeling of activities of people working on a collaborative task that has been broken down into a structure of specialized, coordinated activity, supported by technology.

- **Understanding and Measuring the Quality** - Understanding and measuring the total of properties and characteristics of a phenomenon that are relevant for performing its assigned function.

To acquire knowledge of the individual areas, the following research questions are derived out of the research objective (the letters in brackets indicate the individual disciplines involved, being Conceptual modeling techniques, Business Process Modeling and understanding and measuring Quality):

1. **What must be understood by 'conceptual modeling technique'?** (C)
2. **What must be understood by 'business processes'?** (B)
3. **What must be understood by 'quality'?** (Q)

To accomplish a synthesis between the individual areas, the following questions must be answered. The answer to the final question is the framework that is to be constructed, establishing a synthesis between all three fields:

4. **What is a 'business process modeling technique'?** (C/B)
As said, conceptual modeling techniques aid the construction of abstract, simplified views of a certain phenomenon in (a possible) reality that we wish to represent for some purpose. The phenomena that are modeled are business processes.

5. How can modeling techniques be described to make quality assessable? (C/Q)

In order to measure the properties and characteristics of a certain phenomenon, a (formal) description of the phenomenon is required. The phenomena that are evaluated in this thesis are conceptual modeling techniques, therefore, a (formal) description of conceptual modeling techniques is required.

6. How is the quality of a business process modeling technique evaluated? (C/B/Q)

This last research question integrates the three research areas. The answer to this question is a complete, integrated framework for measuring the quality of business process modeling techniques.

1.3 Philosophical Foundation and Research Methodology

A research methodology describes the underlying assumptions towards scientific research, i.e. the philosophical position, and the methods and techniques that are implied by this position (Swanborn, 1986). In this section, the assumptions with regard to this research are made explicit and the research approach adopted in this thesis will be presented and justified in this context.

In line with De Leeuw (1990) and Nissen (1991) it is assumed that there is not one research methodology for all scientific research but that the characteristics of an individual research project determine the methodology that is most suitable for that project. Furthermore, the research approach is influenced by a research tradition, the axioms (assumptions or beliefs) regarding the nature of scientific research. In the design of the methodology for this research project, the underlying assumptions will be made explicit and following from that, the research approach taken will be unfolded.

Bernard (1994) distinguishes what he believes to be the two dominant research traditions in modern science: the positivist and the interpretivist tradition. In order to categorize research into positivist or interpretivist, it is useful to distinguish between the underlying ontological and epistemological assumptions, i.e. the beliefs one has about the nature of reality and how knowledge about this reality can be obtained (Klein, 1996).

The basic ontological assumption underpinning the positivist tradition is that reality is objectively given and that it can be described independent of the researcher. The role of scientific research is thus to systematically acquire objective
knowledge about phenomena known to exist in this reality. This assumption is strikingly described by Popper (1963) when he states that positivism assumes that the truth is manifest, sometimes clearly present, sometimes 'covered', but always waiting to be literally 'dis-covered' by the researcher.

Although this viewpoint is still very dominant in modern science, there are several (classical) epistemological problems concerning it:

- In the case of simple individual observations of reality, there is no guarantee that the image that perception gives us of reality actually corresponds with reality, since, in order to verify this, we need to compare this image of reality with reality itself, but, we can only access reality itself through perception (c.f. Magee, 1999).

- Even if we could be sure that our perceptions give an accurate account of a possible reality, the work of philosophers such as Kant (1998) shows that there is a limit to what we can perceive. The fact that science is constantly developing more sophisticated measurement and observation instruments to extend this limit, in fact proves its existence. This made Kant distinguish between the phenomenal world, the world of that what can be perceived from objects and the nomenal world, the world that consists of things an sich, that what remains when we 'strip' an object of all its perceivable attributes. We have no access to this world.

- With regard to generic, inductive theories of reality there is the well-known induction problem. Inductive theories generalize simple individual observations done in the past to general theories that predict individual observations in the future. The underlying idea that the future resembles the past is a presumption rather than an objective truth; it cannot be guaranteed that there will be no observation that falsifies the theory in the future. A theory that is constructed as such that it excludes the possibility of falsification (i.e. explains everything might happen in the future) has, oddly enough, no predictive value at all.

The interpretivist tradition has tried to overcome the problems concerning the reliability of perceptions by being more moderate in their claims. The basic ontological assumption underpinning the interpretivist tradition is that the perception and interpretation of a possible reality cannot be separated from the researcher. Knowledge is seen as subject dependent interpretations of phenomena in reality. As opposed to positivism, it does not take reality as a starting point, but subject dependent perceptions and interpretations of it.

Next to the positivistic and interpretivist tradition I would like to briefly discuss social constructivism (c.f. Searle, 1995) as a third tradition (see Falckenberg et al. (1998) for an application in the Information Systems discipline). As an ontological starting point, social constructivism recognizes that a possible reality is not 'out
there' ready to be discovered by the researcher. The basic ontological assumption is that we shape reality through mental constructions and, more important, further socially construct reality, using language as a vehicle to communicate these constructions.

In my opinion, sound scientific research should contribute to an understanding of the world in which we live by means of the construction of scientific theories and artifacts that aid us to conceive and perceive this world. Just as science should build measurement instruments that shape the way we perceive, it should build instruments that shape the way we conceive: generic scientific theories. This opinion is consonant with the interpretivist and social constructivist tradition.

Theories are mental constructions that shape the way we conceive, the truths that they yield are not objective, but must be seen in the light of the theory. This can be illustrated by numerous examples. That Rutherford and Bohr showed us that it is possible and useful to conceive the world around us in terms of particles called electrons and atoms does not mean that the world actually consists of electrons and atoms: Modern quantum mechanics gives us another conception of the world. Newton showed us that it is possible and useful to place the things we perceive in absolute space and time although Einstein's space-time gives us another conception of the world. It is possible and sometimes useful to conceive light as waves, possible and sometimes useful to conceive it as particles. Numerous other examples can be found in Kuhn (1969).

Another example that shows that it is misleading to think that theories provide objective truths about the world around is comes from a major flaw in the work of even a great thinker such as Kant. In his time, Newtonian space and time were generally accepted and taken for absolute truths by Kant. Based on this assumption, Kant believed that absolute space and time are the instruments for ordering our perceptions. After the introduction of Einstein's space-time we must conclude that it is not absolute space and time that orders our perceptions, but simply Newton's mental construction. To me, these examples show that theory is an instrument to conceive the world around us.

Predictive value is often mentioned as a quality of a good theory. I think that this does not do justice to the achievements of the philosophy of Karl Popper (c.f. Popper, 1963). It is not so much that a theory should predict a future state of affairs, it should restrict, impose constraints on the future state of affairs. A good theory should therefore have restrictive value. It should provide us with a conception of the world that predict things not to happen in the future so that we can test theories by means of the Popper's falsification principle. Theories with restrictive value provide us conceptions that reduce the overwhelming complexity of the world around us and make it controllable.
Theories should be justified by application and evaluation. Descriptive research should therefore give an accurate account that it is possible and useful to conceive the world as prescribed by the developed theory and design research should build useful artifacts based on the theory. Scientific knowledge is the result of the conception of the world according to the theory and design knowledge is the result of building artifacts based on the theory.

The aim of the research described in this thesis is to develop a framework that allows one to evaluate business process modeling techniques. This framework is a theory that allows one to make conceptions of the phenomenon business process modeling technique. By means of the application of the proposed framework I will show that it is possible and useful to conceive the phenomenon business process modeling technique as proposed in this thesis. Furthermore I will evaluate the usefulness of the proposed framework as well as its limitations. A more precise description of the research set-up will be given in the next section.

1.4 Outline of this thesis

Part I: Introduction
The thesis is divided into three parts, each part consisting of three chapters. The first part concerns a discussion of underlying assumptions towards research and following from that a description of the way that this research was conducted (this chapter), the definition of basic terms that are used throughout this thesis (chapter 2) and a more specific literature study on existing work regarding the evaluation of models and modeling techniques (chapter 3).

Part II: Proposition
The second part of this thesis concerns the proposal of a theoretical framework for understanding and evaluating business process modeling techniques. It must be seen as a theory that helps us to better understand this relatively new phenomenon. The proposition takes into account the shortcomings and flaws found during the literature research. The proposition is divided over three chapters. Chapter 4 deals with the concept 'quality' in relationship with models and modeling techniques. In Chapter 5 a description language is proposed that allows the capturing of business process modeling techniques. Chapter 6 describes the proposition of a measurement scheme based on the proposed description language that allows the measurement of the qualities identified in chapter 4.
Part III: Application and Validation
The third part of the thesis concerns the application and evaluation of the proposed theory. Aim is to show that it is possible and useful to evaluate the business process modeling techniques as proposed in the second part of the thesis. Chapter 7 concerns the application of the framework on four business process modeling techniques, chapter 8 concerns the application of the measurement scheme and in chapter 9 the application is evaluated and conclusions are drawn as well as directions for further research are proposed. Figure 1-2 shows the outline of the thesis in more detail.
**Correctness**: Application of Measures presumed indicative for correctness of the resulting models. Applied to 4 modeling techniques (15 aspect modeling techniques).

**Usefulness**: Determine the extent to which the techniques are capable of modeling business processes.

---

**Proposition**

- Ch 4: Understanding Quality of Modeling Techniques: Assuring the correctness & usefulness of models
- Ch 6: Measurement Scheme to measure the relevant properties of modeling techniques
- Ch 5: Formal description language for capturing modeling techniques
- Ch 8: Application and Validation of the Measurement Scheme
- Ch 7: Application & Validation of the Description Language

**Validation**

**Assuring Correctness & Usefulness**

**Figure 1-2: Outline of the thesis**
2
THE CONCEPTUAL MODELING
OF BUSINESS PROCESSES

2.1 Introduction

A sound research project should start with the creation of a clear understanding with regard to the terms and definitions of the phenomena under study. In this chapter, concepts such as 'model', 'conceptual model', 'business process model' will be investigated systematically, as well as the techniques and methods for constructing these models. The terms and definitions of these concepts form the basis for the theories that will be developed in the chapters to come. Section 2.2 concerns the definition of conceptual modeling. Section 2.3 pays attention to the role of conceptual modeling techniques and methods. In particular the application of these techniques to model business processes is further elaborated in section 2.4. Section 2.5 provides an overview of commonly used techniques for modeling business processes. Conclusions are drawn in section 2.6.

2.2 Conceptual Modeling

Models

Somebody that uses a system B that exists independent of a system A to acquire knowledge about A, uses B as a model of A. (Apostel, 1960). In general, a model is something that functions as a representation of something that we wish to understand. Models are the result of the application of a theory that prescribes us how we can conceive the thing that we wish to understand. This basic notion is in line with the work of e.g. Nauta (1974).

Scale models, for example, are physical objects of which some properties and relative proportions resemble the properties and proportions of the phenomenon that they model. Other properties, such as the absolute size or the material of which the scale model is constructed, may differ. More abstract is a collection of billiard balls that can serve as an analogue model of an ideal gas. Again, some properties of
the model, such as interaction relationships that change location and impulse, provide useful analogies and others do not.

In science, especially mathematical models are used to shape the way we conceive all kinds of phenomena. Such models are abstract in the sense that phenomena are conceived in terms of abstract mathematical entities. So-called meta-mathematical models, or meta models in short, are models that are used to understand mathematical models themselves. A meta model is a model that is constructed to understand another model.

Models can be used in a descriptive or a prescriptive (normative) way. When a phenomenon in reality is perceived and conceived in the light of a particular theory, this results in a descriptive model of that phenomenon. In design sciences we also see a more prescriptive or normative use of models. This way round, the modeling process starts with a given model and intervenes in reality so that at the end it corresponds to the model. An example of this use of models is to build a new road based on a design made in advance. In this example, the road design is the model. It is used to intervene in reality in such a way that afterwards reality corresponds to the conceptual model.

The same way that descriptive models describe and predict with regard to an existing reality, prescriptive models describe and predict phenomena with regard to a future reality, the reality that comes into being by intervening in the existing reality so that it corresponds to the prescriptive model.

Note that the terms 'descriptive' and 'prescriptive' are somewhat misleading. In fact it is not the model that is descriptive or prescriptive, it is the use of the model in relation to reality that is descriptive or prescriptive. In fact, we construct conceptual systems that can be used as prescriptive and descriptive models of phenomena in reality.

**Conceptualization**

The adjective 'conceptual' in 'conceptual model' indicates that it is a model that consists of concepts. 'Concept' stems from the Latin word 'conceptum', which is the past participle of 'concipere', which means 'to conceive'. A 'conceptual model' is a model that consists of things that are the result of the act of conceiving.

There are various research disciplines that are engaged in the investigation of conceptual modeling (Craig *et al*, 1998). On the one hand, cognitive psychology treat concepts as constituents of thought, internally represented in a persons mind by as a mental image. On the other hand, logic and mathematics study general relationships between concepts and propositions concerning concepts, regardless of how it is represented in an individual mind. In this thesis, the stress will be on the logic and mathematical treatment of concepts.
With regard to the idea of reality and the conception of it, the view that has dominated our western way of thinking for centuries can be traced back to the ancient Greek philosophers. Plato for example distinguishes two worlds: the world of senses and the world of ideas. The world of senses is a world that consists of things that we can observe by touching or feeling it. The things in the world of senses are just shadows of the things that exist in the world of ideas. Clearly this is a distinction between concept and that what corresponds to the concept in a possible reality. Plato's view was adopted later on by Aristotle. Aristotle postulated a world of substance and form. Substance is that what a thing is made of and form is that what a thing actually is. Out of the form of things we create ideas by reasoning about them. So, on the one hand, Plato believed in a world of innate, perfect ideas of which things in the world of senses are imperfect shadows. Aristotle turned this view upside down by postulating a world of things that have substance and form and a world of ideas that comes into existence when we reason about the form of things. Aristotle categorized all the things in the world of senses into concepts and by doing so became the founder of modern logic.

When concepts are discussed in modern science, classes of things are meant. In line with e.g. Sowa (1984) the concepts that are used as such will be called generic concepts. Examples are the generic concepts 'Car', 'Person' and 'Philosopher'. Individual concepts are thoughts of examples or instantiations of these generic concepts, e.g. 'the car with license plate 01-AB-CD', 'the person John', 'Socrates' and 'Plato'. Generic concepts cannot have a direct correspondence with a thing that exists in reality. On the other hand, individual concepts might correspond to a thing perceivable in reality but do not necessarily have to: we are able to perceive 'the person John', but not 'the number 5'.

Another important question is how concepts, being the elements of thought, are grouped together from complex concepts or thoughts (for example Wittgenstein's language functions). The thought that 'philosophers are persons' draws a relationship between the generic concepts 'Person' and 'Philosopher'. On the other hand there seem to be thoughts that draw relationships between individual concepts such as 'John owns the car with license plate 01-AB-CD'. It can be the case that thoughts themselves are instances of a generic class. E.g. the thought that 'John is married to Jill' is a structure that on the one hand relates 'John' and 'Jill' and on the other hand is an example of the generic concept 'Marriage'. Relationships that are often used in conceptual modeling, such as generalization, aggregation, association and instantiation are all examples of thoughts that somehow group individual and/or generic concepts.
Table 2-1: Classification of thoughts to often used relationship types

<table>
<thead>
<tr>
<th>Examples of thoughts</th>
<th>Concepts Involved</th>
<th>Often classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>'there are persons'</td>
<td>generic</td>
<td>Classification</td>
</tr>
<tr>
<td>'philosophers are persons'</td>
<td>generic</td>
<td>Generalization</td>
</tr>
<tr>
<td>'wheels are part of cars'</td>
<td>generic</td>
<td>Aggregation</td>
</tr>
<tr>
<td>'persons can own cars'</td>
<td>generic</td>
<td>Association</td>
</tr>
<tr>
<td>'John is a person'</td>
<td>individual, generic</td>
<td>Instantiation</td>
</tr>
<tr>
<td>'John owns the car 01-AB-CD'</td>
<td>individual, individual</td>
<td>Association Inst.</td>
</tr>
<tr>
<td>'wheel 1234 is part of car 01-AB-CD'</td>
<td>individual, individual</td>
<td>Aggregation Inst.</td>
</tr>
</tbody>
</table>

Language

Language is the vehicle for reasoning with thoughts and communicating thoughts with others. With language we express thoughts. We perceive and conceptualize the world according to theories that we have and express these conceptualizations in a language.

On the one hand, language allows us to express what we think linguistically, on the other hand, it could be argued that what we are able to think depends on what we are able to express linguistically. It is generally believed that language is shared and socially constructed although the ability to use language seems to be innate. The study of linguistics treats languages as complex systems and seeks explanations of the language's syntax, semantics and pragmatics (c.f. Nauta, 1974):

- **Syntax** - the rule system that determines the properly constructed expressions in a language. In contrast to natural languages, formal languages are subject to precise rules. Typical for this class of languages is that their syntax is established by means of a set of rules that prescribe precisely what kind of utterances are allowed.

First order predicate logic, introduced by Aristotle, has been the most important and influential formal language for expressing thoughts and reasoning with thoughts. This use of this language will be illustrated by a well-known example. Take the following three thoughts, expressed in natural language: '1) All men are mortal, 2) Socrates is a man, therefore 3) Socrates is mortal'. In first order predicate logic, these expressions are translated as follows:

\[
\forall x [M(x) \rightarrow S(x)] \\
M(a) \\
S(a)
\]
Capital letters, such as the 'M' and 'S' in the example, are called predicates. In general they are used to correspond to features or properties of things. Lower case letters 'a', 'b', 'c', ..., 't' are called individuals. Predicates can be used to specify a feature or property of an individual thing, such as M(a) and S(a) in the example. Lower case letters x, y, z are called variables, they are used in formulas in combination with quantors such '∀' and '∃'. In line with the example, it is more precise to speak of the generic concept 'M(x)' instead of 'Men', the individual concept 'M(a)' instead of 'John' and e.g. 'Owns(a,b)' instead of 'John owns the car with license plate 01-AB-CD'.

• Semantics - the ways that expressions in a language form meaning. Although a good syntax enables precise reasoning and proving of theorems out of other theorems it does not describe the ways that the expressions in a language form meaning. The specification of the meaning of expressions in a formal language is called an 'interpretation' of the formal language. Examples of classes of interpretations are empirical interpretations in which the expressions of the formal language refer to a possible state of affairs in reality that can be validated by empirical observations and mathematical interpretations in which the expressions refer to ideal mathematical entities such as numbers, circles, lines, etc. The key to the meaning of a sentence is the notion of a truth-condition. A statement's meaning determines a condition that must be met if it is true. For words that express single concepts this is a condition of existence, for phrases that express complex thoughts this is the actual truth-condition. The sentence 'There is a person John' has a meaning because it can be true in case that John exists. The sentence 'John is married to Jill' has meaning because it can be true in case a certain state of affairs in reality holds.

• Pragmatics - the practical use of language and intended effect of the language. It is the actual use of a language that determines how to interpret it. With the utterance "I'm hungry" I could intend to share the knowledge that I have not eaten enough, I could hint someone to give me food or that it could be an invitation for the daily lunch.

Knowledge
Knowledge comes in many forms, such as to know people, to know how to ride a bicycle, to know that you're thirsty, to know that the particles in an ideal gas have a similar behavior as a collection of billiard balls, to know that the sun rose today at 7.30 or to know that 1+1=2. Consequently, knowledge can be classified in various ways.

A distinction can be made between techné and épistémé. The distinction between knowledge about how to do things or craftsmanship (techné) and knowledge about the state of affairs of some reality (épistémé) originates from Aristotle. How to ride a bicycle is an example of techné, the fact that a bicycle is a special kind of vehicle is
an example of epistêmê. The term *factual knowledge* will be used throughout this thesis to denote epistêmê.

As discussed in previous sections, individual thoughts can be expressed as sentences in a (formal) language and have the property that they can be true if what is stated is in accordance with reality or false if this is not the case. It was concluded that thoughts do no necessarily have to be in accordance with reality but have the property of being in accordance with reality or not. Factual knowledge is composed out of those thoughts that are in accordance with reality.

Another important distinction is the distinction between *generic* factual knowledge and *individual* factual knowledge. Individual factual knowledge is composed out of those true thoughts that concern individual concepts, for example, the fact that the sun rose at 7.30 this morning. Generic factual knowledge does concern generic concepts. An example is the fact that the particles in an ideal gas have a similar behavior than a collection of billiard balls.

The distinction between *a-priori* and *a-posteriori knowledge* is described by Kant (1998). A-priori knowledge is factual knowledge that one can have without appealing to perceptions of reality, such as the fact that 1+1=2. Especially in mathematics and logic, a-priori knowledge is deduced out of the basic axioms governing these disciplines. The fact that the sun rose today at 7.30 is a-posteriori knowledge, since one has to actually perceive the sun rising to obtain this knowledge.

The distinction between *scientific* and *non-scientific* knowledge is known as the demarcation problem (Popper, 1963). The distinction applies in particular for generic knowledge. In modern science, the most prominent criterion for distinguishing between the two is the falsification principle. Generic knowledge is scientific if we can derive individual knowledge from it that can be tested true or false. In this light, to know that the particles in an ideal gas have a similar behavior than a collection of billiard balls, is an example of scientific knowledge. We can think of experiments to test the truth of individual examples.

It is important to realize the dual character of knowledge: there is something that is known and there is someone that knows. Knowledge cannot be seen apart from the person that has the knowledge or group of persons that share knowledge. We use *information* to share knowledge between individuals. As said, knowledge is composed out of true thoughts that can be expressed using a (formal) language. As soon as we express these thoughts in a language and use a certain medium to transfer these expressions to other individuals we speak of *communication* or the exchange of information. Information is composed out of thoughts expressed in a particular language using a particular medium.
Theory, Model & Reality

Figure 2-1 shows the relationship between theory, model and the reality that is modeled. A theory is conceived as a coherent set (system) of generic knowledge prescribing how a certain phenomenon in reality can be modeled. Models are instantiations or applications of the theory. In their turn, models can be conceived as a coherent set (system) of individual, a-posteriori knowledge, the result of perceiving and conceiving a particular phenomenon in reality in the way that is proposed by the theory. This process is called conceptualization. On the other hand, realization is the process in which we build artifacts in reality according to a prescriptive or normative model of that artifact.

![Diagram](image)

**Figure 2-1: Theory, Model & Reality**

Figure 2-2 depicts the classical view on the origin of theory, earlier touched upon when Aristotle’s view on conceptualization was discussed. Aristotle thought that generic concepts (ideas) and generic knowledge comes into being by first perceiving things in reality (individual concepts and knowledge about these concepts) and after that classifying these perceptions into generic concepts. Also the classical empirists such as Hume and Locke (McGee, 1999) placed empirics (perception of reality) before conception and the construction of theory.

![Diagram](image)

**Figure 2-2: Classical View on the Origin of Theory**

As said earlier in chapter 1, the view concerning the origin of theory adopted in this thesis is that theory is a mental construction, an instrument that shapes the way we conceive the world around us. We cannot derive theories out of individual
perceptions and conceptions because we cannot conceive things without having a theory in advance. Theory prescribes how to conceive reality. In contrast to the view of empirists such as Hume and Locke, first theory is constructed and after that, reality is perceived and conceived the way that the theory prescribes (Magee, 1999).

Take for example the modeling of small particles that was used as an example before. The modeling technique used by Rutherford and Bohr was to model the world around us in terms of the generic concepts 'electron', 'proton' and 'neutron' and generic knowledge such as the forces that act upon these things. In ancient Greece, the philosopher Empedocles thought of the generic concepts 'water', 'fire', 'earth' and 'air' to model the same phenomena. This illustrates the various ways in which theories can shape our world.

The use of prescribed generic concepts and knowledge to model a phenomenon such as the human brain and especially the transience of the use of those concepts is illustrated by the following quotation of John Searle (1984): "Because we do not understand the brain very well we are constantly tempted to use the latest technology as a model for trying to understand it. In my childhood we were always assured that the brain was a telephone switchboard. ('What else could it be?') I was amused to see that Sherrington, the great British neuroscientist, thought that the brain worked like a telegraph system. Freud often compared the brain to hydraulic and electro-magnetic systems. Leibniz compared it to a mill, and I am told some of the ancient Greeks thought the brain functions like a catapult. At present, obviously, the metaphor is the digital computer". Either it turns out to be possible and useful to construct models of reality in the light of a particular theory or it turns out that it is not possible or not useful to construct models of reality in the light of a theory, see Figure 2-3.

![Figure 2-3: Falsification and Justification of Theory](image)

To summarize, a conceptual model is, as any model, something that is constructed to better understand the phenomenon that is modeled. Specifically, a conceptual model is a coherent set of individual concepts and knowledge concerning the modeled phenomenon, shaped by the way that the theory prescribes to conceive reality.
2.3 Conceptual Modeling Methods and Techniques

Conceptual Modeling Techniques
According to Webster's online dictionary, a technique is the manner in which technical details are treated. In general, a conceptual modeling technique is considered to be a body of technical knowledge that guides modelers through the construction of a conceptual model of (a certain phenomenon in) reality. It reduces the complexity of the modeling task by making knowledge with regard to the following constituents explicit:

- a modeling language - an important constituent of a modeling technique is a formal or semiformal modeling language. The underlying theory on how to conceive the phenomenon under study is worked out by means of this language. The modeling language consists of a syntax that allows the expression of concepts and their relationships. The modeling language reduces the complexity of the modeling task by reducing the expressiveness of natural language to a language that allows the description of concepts and interrelationships that are relevant for the purpose at hand. Furthermore, the (semi)formality of conceptual languages reduces the ambiguity with regard to the interpretation of the meaning of expressions compared to informal languages such as natural language. Most of the conceptual modeling languages studied in this thesis have a diagrammatic denotation;

- modeling procedures - another constituent of a modeling technique is the explicitation of procedures or recipes for the perception and conception of the elements that constitute the conceptual model. These procedures specify the sources out of which the model is derived, e.g. text documents, direct observation or interviews with domain experts and ways of retrieving concepts out of these sources, e.g. noun phrase recognition, recognition of key words, patterns, etc.;

- reference models - other constituents are so-called reference models or patterns. These reference models are predefined conceptual models that contain knowledge that helps the modeler to recognize (structures of) concepts. Examples of reference models are models of best practices in a certain line of business, such as the Enterprise Resource Planning (ERP) reference models (Post, et al, 1996), models prescribing recurrent patterns or structures such as the ActionWorkflow loop that prescribes that each sequence of actions in the concept of a 'workflow loop' consists of a proposal, agreement, performance and satisfaction phase (Medina-Mora, 1992) or agreed upon standards such as ISO standards for product data;

- the definition of a coherent set of aspect modeling techniques - In order to model different perspectives on the same phenomenon in a comprehensive way, most
modeling techniques distinguish several coherent aspect modeling techniques that each focus on a single system's perspective. An example of a multi aspect modeling approach is the UML that incorporates for instance Use Case modeling, Class modeling, State-transition modeling, and many more (OMG, 2001).

In this thesis, the term 'modeling technique' is used to denote both single perspective modeling techniques and multi aspect modeling techniques as a whole. Modeling techniques within a multi modeling technique that focus on one perspective will be referred to 'aspect modeling techniques'.

The following distinction is made between 'aspect model' and 'whole model': An aspect model is part of a whole model and focuses on a particular aspect of the modeled phenomenon. Together, all aspect models of a certain phenomenon constitute a whole model of the phenomenon. Examples of aspect models within UML are a Class model, Use Case Model and Sequence Model of a certain software system (OMG, 2001). Together these aspect models constitute a whole model of the software system. An aspect model can be represented in a diagram, a whole model is represented by representing the diagrams of the separate aspect diagrams.

**Conceptual Modeling Methods**

Whereas the previous sections dealt with the understanding of conceptual models and the role of techniques, this section elaborates on methods for conceptual modeling, the so-called conceptual modeling methods.

The term 'method' is not to be mistaken with the term 'technique' although this is often the case in both theory and practice. The English term 'method' stems from the Latin 'methodus' and from the Greek 'methodos'. Roughly, the term 'methodos' can be translated as 'Dealing with ("meta-") the way ("hodos")'. Unfortunately, there is not much consensus with regard to the exact meaning of the term, leave alone its relationship to related terms such as 'methodology', 'technique' and 'procedure'. In general, a method concerns the ordered way of working seen in time and can be conceived as a system consisting of interrelated tasks that have to be carried out to solve a particular class of problems. The order in which tasks can be carried out is an example of an interrelationship between tasks. For each task, techniques are the body of technical knowledge that guides modelers through the accomplishment of the task.

In the particular case of a conceptual modeling method, the tasks that have to be carried out concern the construction of conceptual models. Therefore, a 'conceptual modeling method' can be viewed as a system of interrelated tasks that have to be carried out to construct a conceptual model of a certain phenomenon. It is convenient to use the distinction between aspect models as a criterion for
distinguishing different modeling tasks. As such there is a one-to-one relationship between task and aspect model: Each task in the modeling method is concerned with the construction of a certain aspect model. Between tasks there can be binary precedence relationships defining the order in which tasks have to be carried out.

2.4 Conceptual Business Processes Modeling

If conceptual modeling is deployed to apply a theory that prescribes to conceive certain phenomena in reality as 'business processes', then we speak of 'conceptual business process modeling'. Conceptual models of (an aspect of) the phenomenon 'business process' are called 'conceptual business process models' or 'business process models' in short. The term 'enterprise modeling' that is sometimes used is treated as a synonym for conceptual business process modeling. By consequence, 'enterprise models' are a synonym for business process models. 'Business Process Modeling Techniques' are techniques that support the modeling of business processes.

In this chapter, a few words are spent on the current consensus on how to conceive business processes. This idea is further elaborated in chapter 4 when the suitability of techniques for modeling business processes is addressed. In literature, a business process is often defined as a structure of organizational or inter-organizational activities that are necessary to accomplish a product or service. Examples of this definition are "an ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs" (Davenport, 1993) or "A set of activities that, taken together, produce a result of value to a customer" (Hammer et al, 1993). Taylor (1993) describes a business process as a set of specialized activities where coordination is achieved through communication.

The above definitions are summarized by fitting them in a more precise systematic description. It is concluded that a business process is a system that performs activity, i.e. it executes a sequence of goal-directed actions (c.f. Bunge, 1979). More specific, it is an organizational system, or 'organization': A system that performs organized or coordinated activity in a sequence of goal-directed actions. This coordination is achieved through communication.

For a primary process, the goal is directly related to the production of a result of value for a customer (Hammer et al, 1993). For a secondary process, i.e. a process that creates the necessary conditions for other processes to be carried out, its goal is indirectly related.

The above description is generally agreed upon and therefore a vast majority of modeling techniques allows business processes to be modeled as sequences of goal-directed actions. The way that details are filled in differ and depend very much on
the purpose for which modeling techniques are used. To conclude with, the relationship between theory, business process modeling techniques, process models and reality is depicted below.

![Diagram showing the relationship between (Possible) Reality, Hypotheses regarding the nature of business processes, Theories on BP Modeling, Business Process Models, realization, and conceptualization.](image)

Figure 2-4: Business Process Modeling

2.5 Commonly Used Business Process Modeling Techniques

Various approaches for modeling business processes exist. Depending on the approach that is taken to make an inventory of these techniques, the outcome may differ. In order to come to an overview of commonly used techniques I have combined three approaches. Firstly, I have taken a look at the information systems books that are currently used in the information systems curriculum. Secondly I have examined the report of Dommelen et al (1999) that made an overview of business process modeling tools available on the Dutch market, by order of the Dutch department of finance. And finally, I have taken a look at recent proceedings of conferences in this area.

The conclusion that can be drawn from the examination of the information system books of Alter (1996), Zwass (1998) and Gupta (2000) is that both three books prescribe techniques such as Data Flow Diagramming and Flow Charting to model business processes, eventuating via pseudo-code in computer programs that support these processes. Furthermore they prescribe Entity Relationship Diagramming to model the information used and produced by business processes, eventuating in databases that the programs work on.

Another approach to make an inventory of modeling techniques is to study the automated modeling tools that support these techniques. It is important to notify that there is distinction between techniques and tools (see Figure 2-5). Modeling techniques offer a set of modeling concepts that allow the modeler to conceive reality in a certain way. Modeling techniques can be supported by various tools, ranging from pen and paper to sophisticated automated modeling tools. Tools offer functionality to support the application of techniques. Although in theory there is
an n:m relationship between tool and technique, in practice we often see that there is a 1:1 relationship between technique and a tailor made modeling technique, often a dialect of a commonly known technique. Free drawing tools such as pen and paper, automated drawing tools such as Microsoft® Powerpoint, et cetera, have a 1:0 relationship with techniques: They lack any methodological foundation.

![Figure 2-5: Tools and Techniques](image)

Dommelen et al (1999) presented an overview of tools together with the modeling techniques that they support. This overview can be summarized as depicted in Table 2-2. Most techniques supported by the tools are specific dialects of the commonly known technique.

**Table 2-2: Tools & Techniques (Dommelen et al, 1999)**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Technique(s) supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIS</td>
<td>Event Driven Process Chains (EPC's)</td>
</tr>
<tr>
<td>BPWin</td>
<td>Integrated DEFinition (IDEF)</td>
</tr>
<tr>
<td>BWise</td>
<td>Unified Modeling Language (UML), Petri-nets</td>
</tr>
<tr>
<td>DEM/SE</td>
<td>Petri-nets</td>
</tr>
<tr>
<td>Designer/2000</td>
<td>Petri-nets, Entity Relationship Diagramming (ERD/IE)</td>
</tr>
<tr>
<td>MAVIM</td>
<td>-</td>
</tr>
<tr>
<td>Movix</td>
<td>-</td>
</tr>
<tr>
<td>Protos</td>
<td>Petri-nets</td>
</tr>
<tr>
<td>SA/BPR</td>
<td>Integrated DEFinition (IDEF)</td>
</tr>
<tr>
<td>SDW</td>
<td>Data Flow Diagramming (DFD) and Entity Relationship Diagramming (ERD)</td>
</tr>
</tbody>
</table>

Finally, when we take a look at recent proceedings of conferences concerning Advanced Information Systems Engineering and Business Process Management

- **Structured techniques** - Amongst the structured techniques, the classical Data Flow Diagrams and Entity Relationship Diagrams are reckoned as applied by Yourdon (1989). A tool that supports structured modeling is for example SDW.

- **Flow Chart Techniques** - Flow Charting is a technique for modeling business processes as a simple structure of activities together with input and output flows without much restriction and precision. In general, this technique is not a based on a formal language definition. Examples of tools supporting this technique are AllClear, BONAPART and multipurpose tools such as Microsoft Visio.

- **CPN based techniques** - Colored Petri-nets are a variant of the classical Petri-nets as introduced by Petri (1962). Petri-nets offer a theoretically underpinned, precise way of modeling the states and state transitions in business processes, often used in the context of workflow modeling. An Dutch example of a tool supporting a variant of Petri-net modeling is Protos.

- **IDEF based techniques** - As opposed to the above techniques that must be considered as techniques that focus on a single perspective when modeling business processes, IDEF is a so-called multi modeling technique that offers several aspect modeling techniques that cover different perspectives of an organization. For example, IDEF0 conceives a business as consisting of business functions, IDEF1 models the information in organizations and IDEF3 conceives a business process as consisting out of states and state transitions. Tools that support IDEF are, amongst others, Workflow Modeler and Business Object Modeling.

- **Object Oriented techniques** - Object Oriented Techniques are currently the standard for modeling software systems. The Unified Modeling Language (UML) is the standard language for modeling these systems. More and more it can be observed that Object Oriented thinking is applied to model business processes and organizations. An example of tools supporting object oriented approaches is Rational Rose.

- **EPC oriented techniques** - Event Driven Process Chains are currently widely used to model business processes by means of events and functions, especially in the context of Enterprise Resource Planning (ERP) systems like SAP. It offers a number of views (aspect modeling techniques) to model the different
perspectives of an organization. The most important tool for modeling EPC's is the ARIS toolset.

- *Speech Act oriented techniques* - Speech Act oriented techniques stress on the communication that takes place in organizations to coordinate actions. Typically organizations are modeled as actors. These actors coordinate their work by exchanging speech acts, such as to 'request' some actor to do something or to 'declare' to another actor that something has been carried out. An important modeling tool and workflow system that supports these techniques is the toolset of ActionTech.

### 2.6 Conclusions

In this chapter, the basic terms and definitions were discussed that will be used throughout this thesis. Systematically the terms conceptual modeling (encompassing the definition of theory, model, concept and knowledge), conceptual modeling technique and method were introduced. In particular, business process modeling techniques were addressed.

With regard to conceptual modeling, a conceptual model was defined as a coherent set of individual concepts and knowledge concerning the modeled phenomenon, shaped by the way that the theory prescribes to conceive reality. A distinction was made between generic knowledge (theory) and individual knowledge (model). Furthermore attention was paid to the constituents of modeling techniques and modeling methods.

An overview of commonly used business process modeling techniques was derived out of a study of existing books used in the information systems curriculum, an inventarization of tools used in the Netherlands and by taking a look at recent conferences.
3

EXISTING APPROACHES FOR EVALUATION

3.1 Introduction

The aim of this chapter is to provide an overview of existing approaches that are somehow related to the evaluation of modeling techniques, to discuss these approaches and to draw conclusions on their advantages and disadvantages. The chapter is organized as follows: Section 3.2 elaborates on existing quality criteria for conceptual modeling as well as integrated frameworks for understanding quality.

In the previous chapter, a distinction between multi modeling technique and aspect modeling technique was made. Aspect modeling techniques are used to model different system perspectives. Section 3.3 gives an overview of the literature on these perspectives.

The use of meta models, the comparison of meta models and the use of ontological comparisons as a means to measure the suitability of a technique to model business processes is elaborated in section 3.4. Section 3.5 elaborates on the various ways in which the known quality criteria are measured. The chapter ends with conclusions on the existing approaches.

3.2 Quality Criteria and Frameworks

Scientific research into the quality of models used in information systems and business process modeling starts approximately in the late seventies and early eighties. An example publication is the report of Griethuisen (1982) on the principles governing conceptual modeling (Griethuisen, 1982). At that time, the focus was mainly on the quality of data models, in particular on Entity Relationship Models. Publications such as those of Batini et al (1992), Wang et al (1994) and Moody et al (1994) mention tens if not hundreds characteristics that a good data model should possess. Although the focus is primarily on data models, many of the proposed characteristics are not restricted to this perspective and can serve as
criteria for other perspectives as well. As a starting point for literature research into the quality of models, the criteria that are most often mentioned and are not specific for data modeling are discussed below:

- **Completeness** - the extent to which the model contains all statements that can be expressed about the modeled domain (Batini et al, 1992, Moody et al, 1994, Wand, 2000, Reingruber et al, 1994).

- **Correctness**
  - *syntactic correctness* - the extent to which statements expressed in the model are in accordance with the syntax of the modeling language used;

- **Consistency** - the extent to which statements expressed in the model do not contradict each other (Wand, 2000), (Wang et al, 1994).

- **Minimality** (also mentioned as Simplicity) - the size and complexity of the model (Batini et al, 1992, Moody et al, 1994, Wand, 2000).

- **Comprehensibility** (also mentioned as Understandability, Readability, Self-explanatory) - the ease with which concepts and thoughts in the model can be understood by the users of the model (Batini et al, 1992, Moody et al, 1994).

- **Predictive value** - the extent to which statements can be derived out of the model that are in accordance with the domain modeled (Wand, 2000).

At least in the early stages, neither attention was paid to the justification or completeness of the proposed criteria nor to the interrelations that may exist between the proposed criteria. Furthermore, the means to measure the extent to which models comply with these criteria are not provided. A possible way to do this is to use metrics that calculate values indicating the extent to which a criterion is fulfilled, as was done before with the measurement of software quality (Boehm, 1978).

**Model Quality according to the Krogsie Framework**

A serious attempt to propose an integrated framework for understanding quality in conceptual modeling originates from the work of Lindland et al (1994). This framework, originally intended for understanding the quality of requirements specifications, was extended by Krogsie et al (1995) to resolve earlier shortcomings and to include ideas presented in another framework that was introduced by Pohl (1993). In Krogsie et al (1999), the framework is further developed and presented as a framework for understanding the quality of conceptual modeling. The
discussion in this section is based on this version of the framework. For the sake of simplicity, it will be referred to as the Krogstie framework.

![Diagram of the Krogstie Framework](image)

**Figure 3-1: Krogstie Framework (Krogstie et al, 1999)**

The Krogstie framework is closely linked to linguistic and semiotic concepts. The six semiotic levels (Stamper, 1973) being physical world, empirics, syntactics, semantics, pragmatics and social world are easily recognized as six equally named quality goals in Figure 3-1. In addition the semiotic quality goals, knowledge quality and language quality are added as goals. The most important notions in the framework will be discussed below. To be able to understand these quality goals, the framework distinguishes, amongst others, the following sets:

- **The audience** $A$. The elements of this set are individual actors, organizational actors and technical actors. Elements of this set are called *participants* in the modeling process. A social actor is typical a person in a certain role, a technical actor is typically a computer program, e.g. a case tool. The *social audience* is the subset of $A$ consisting of social actors, the *technical audience* is the subset of $A$ consisting of technical actors.

- **The language extension** $L$, i.e. the set of all statements that are possible to make according to the vocabulary and syntax of the modeling language used. The framework allows the use of multiple languages which will not be discussed here.
- *The externalized model* $M$, i.e. the set of all statements in someone's model of part of the perceived reality written in a language.

- *The modeling domain* $D$, i.e. the set of all statements that can be stated about the situation at hand.

- *The relevant explicit knowledge of the audience* $K$, i.e. all possible statements of the audience that would be correct and relevant for addressing the problem at hand.

- *The social audience interpretation* $I$, i.e. the set of all statements which the social audience perceives that an externalized model consists of.

- *The technical audience interpretation* $T$, i.e. the set of all statements which the technical audience perceives that an externalized model consists of.

Based on these sets, the quality goals and ways to measure the extent to which these goals are achieved are defined as follows:

- *Physical Quality* - deals with the physical representation of models. According to the authors, its basic features are externalization, the degree to which the knowledge of the audience has been recorded using a conceptual modeling language and internalizability, the degree to which the externalized knowledge is persistent and available to the audience. An example of a measure for externalization is the number of statements known about the domain that are not yet stated in the model, $\#(K \setminus M)$, divided by the number of statements known of the domain $\#K$. With regard to internalizability no concrete measures are mentioned, but e.g. the adoption of the meta model of the language by temporarily adding or removing concepts based on their relevance is seen as an important activity in this context.

- *Empirical Quality* - deals with the variety of elements distinguished, error frequencies, coding (shapes or boxes) and ergonomics for Human-Computer Interaction. Since these issues influence the appearance of statements but not the statements itself, there are no set-theoretic measures proposed. A number of guidelines for graph aesthetics and graph layout is given.

- *Syntactic Quality* - deals with the correspondence between the model $M$ and the language extension $L$. A measure for syntactical correctness is that all statements in the model are according to the syntax and vocabulary of the language, i.e. $M \setminus L = \emptyset$. Furthermore, the $M$ should not lack statements that are implied by the language's syntax.
• **Semantic Quality** - deals with the correspondence between model and domain. Two goals in this context are validity and completeness. Validity means that all statements made in the model are correct and relevant for the problem, i.e. $M \setminus D = \emptyset$. Completeness means that the model contains all the statements that would be correct and relevant for the domain, i.e. $D \setminus M = \emptyset$. By consequence, $D = M$ holds for models that are semantically valid and complete. It is argued that under certain circumstances, e.g. a low consensus amongst the audience, equivalence between $D$ and $M$ is not reachable and one should focus on minimizing $\#(M \setminus D)$ and $\#(D \setminus M)$, i.e. feasible validity and completeness.

• **Perceived Semantic Quality** - deals with the correspondence between the audience's interpretation of a model and his or hers knowledge of the domain. Perceived validity means that all statements of the audience's interpretation of the model are part of the audience's knowledge of the domain, i.e. that $I_i \setminus K_i = \emptyset$ for a certain member $i$ of the audience $A$. Perceived completeness means that the statements that are part of the audience's knowledge of the domain are in the audience's interpretation of the model, i.e. $K_i \setminus I_i = \emptyset$ for a certain member $i$ of the audience $A$.

• **Pragmatic Quality** - tackles the correspondence between the model and the audience's interpretation of it. The pragmatic goal is comprehension, i.e. that $I_i = M$ for a certain member $i$ of the audience $A$. Under certain circumstances, e.g. a low consensus between the audience, $I_i = M$ is not reachable and one should strive for minimizing the differences between the two, i.e. feasible comprehension.

• **Social Quality** - tackles the problem of agreement between the members of the audience. In order to measure this, differences between the sets $I_i, M_i$ and $K_i$ of each member $i$ of the audience $A$ are evaluated.

The Krogstie framework explains quality of models by means of a number of sets and interrelations between these sets. In addition to metrics that are based on set relationships, additional guidelines for quality are given. It is important to note that the framework deals with quality on the individual model level rather than the quality of modeling languages and techniques. Illustrative examples are syntactic quality, the degree to which an individual model is in accordance with the language and semantic quality, the degree to which an individual model is in accordance with the domain.

The choice of sets of statements as the basis of the framework introduces some difficulties. First of all, the framework does not include the appropriate sets to be able to measure physical and empirical quality. Since these qualities deal with the representation of the statements rather than the statements themselves, the framework does not offer any adequate metrics for measuring these qualities. E.g. It
is not clear why the degree of 'externalization', i.e. the degree to which the knowledge of the audience has been recorded using a conceptual modeling language, is a measure for physical quality.

More difficulties emerge when one starts wondering what the elements are of these 'sets of statements', i.e. the extension of the sets. To determine whether something belongs to the extension of the set $L$ that contains all statements that are possible according to the language's syntax seems feasible by checking the statements against the formal syntax of the language. However, the extension of the set $D$ that consists of all statements that 'can be stated about the situation at hand' is hard or impossible to determine, and, in what language are these statements expressed? Determination of the extension of the sets $K$ (relevant explicit knowledge of the audience) and $I$ (social audience interpretation) comes down to introspection by the audience. Furthermore, since these statements need not be stated in the language $L$, one wonders in what language these statements are formulated and if these sets can contain the same statements in different languages, meaning the same things.

The framework offers no means to describe the structure of modeling languages but describes languages by the set of all statements possible according to the language. If we consider formal constraints on sets of statements as part of the syntax of the language as is usually done, then problems occur. Consider e.g. the statements 'Peter is a boy' and 'Peter is a girl'. Most languages would allow these statements to be uttered, so both statements belong to the set $L$. However, if both these statements are in the same model $M$, many would call this model 'inconsistent', regardless of the domain $D$. In Krogstie's framework, 'consistency' is not mentioned and the example model would be syntactically correct.

In the framework, 'semantic quality' is defined as concerning the relationship between model and domain, the extent to which the model corresponds to the domain. As argued in chapter 2, semantics is commonly used in linguistics to discuss the meaning of statements, regardless of whether a statement that has meaning is in correspondence with the domain modeled. Statements that have a truth condition, i.e. statements that can be true or false, have meaning and semantic quality. This leaves in the middle whether a statement is actually true or false, i.e. in accordance with the domain or not. A statement being in accordance with the domain is a valid statement.

Another point of criticism is the definition of 'pragmatic quality' that is not in accordance with how pragmatism of language is normally studied. The pragmatic aspect of a language deals with the intended effect of language. E.g. a speaker uttering the sentence 'It is dark here' not as an assertion but with the intended effect that someone switches on the light. It is clear that this kind of quality is not measured by measuring the consensus of the audience on the meaning of a model as is proposed by the Krogstie framework.
It can be concluded that the Krogstie framework gives a reasonable account of the quality of individual models (not modeling techniques). One could argue whether some linguistic terms have been interpreted in the correct way. It lacks to give appropriate objective measures due to the difficulty of discovering the extension of the proposed sets on which these measures rely.

**Guidelines Of Business Process Modeling**

Becker *et al* (2000) have presented guidelines for sound business process modeling, using a previously developed Guidelines of Modeling (GoM) framework. The framework distinguishes correctness, relevance, economic efficiency, clarity, comparability and systematic design as model quality properties and provides several guidelines that modelers should apply in order to improve the quality of resulting models.

![Diagram of the Guidelines of Modeling (GoM) framework](image)

**Figure 3-2: The Framework for the Guidelines of Modeling (GoM) Becker *et al* (2000)**

The framework distinguishes three levels of guidelines. Level 1 concerns general guidelines regarding the model quality properties mentioned above. Level 2 concerns guidelines specific for the purpose for which the model is used (e.g. for Simulation purposes) and level 3 concerns guidelines specific to the modeling technique used (e.g. specific for Event Driven Process Chains).

With regard to correctness, a general guideline is that a model should be syntactically correct (that is, in accordance with the syntax of a modeling language) and semantically correct (that is, being consistent with the real world). The
relevance of a model is assured when the modeler selects the proper object system to be modeled, the relevant modeling technique and develops the relevant models. Economic efficiency deals with the use of reference models, model re-use and the use of tools.

The clarity property is said to deal with 'the pragmatic aspect of the semiotic theory': readability, understandability and usefulness. It could be questioned whether this is indeed in line with the semiotic theory. In this thesis, pragmatics is interpreted as presented earlier in section 2.2, as the intended use of a statement in a particular language. Comparability concerns the extent to which models are comparable to other models. Consistent use of all guidelines over time in a modeling project contributes to models that are comparable to each other. The last property concerns systematic design. Guidelines prescribe that there should be well-defined relationships between different models, e.g. between process models and information models.

A number of conclusions can be drawn from this work. Firstly, that the quality properties and guidelines concern the level of individual models. They are not guidelines for the construction of modeling techniques. The work does however reference to modeling techniques. One of the guidelines to assure the relevance of a model is the selection of the proper modeling technique. This thesis addresses the evaluation of techniques prior to that choice.

Secondly, the way that quality properties are distinguished is sometimes not in line with other work. For example, the way that the clarity property is said to deal with 'the pragmatic aspect of the semiotic theory' as discussed above is not in line with the definition of pragmatics used in chapter 2 of this thesis, the intended use of a statement in a particular language. With regard to systematic design, interrelationships between different models, e.g. between process models and information models are discussed. In my opinion, these relationships can be viewed as consistency relationships. Overlapping information in one model, e.g. a process model, should be consistent with information in another model, e.g. an information model. Therefore, these interrelationships can be seen as syntactical rules between individual models and can be grouped under the (syntactical) correctness property. An individual model should be syntactical correct as well as the total of all coherent models.

Language Characteristics according to FRISCO
The FRISCO report (Falkenberg et al, 1996) represents the result of the work of the IFIP WG 8.1 Task Group FRISCO to provide a general framework that encompasses the fundamental concepts in the Information Systems field. Although the applicability of this framework is limited to the information systems discipline, it provides three relevant characteristics of modeling language quality in general and thus also for languages for modeling business processes. As opposed to the
Krogstie framework, these characteristics are not on the level of an individual model but on the level of modeling techniques.

The first property that is discussed in the FRISCO framework is the property of *expressiveness*, defined as the degree to which a given modeling language is capable of denoting the models of any number and kind of application domains.

Secondly, *suitability* is identified as an important characteristic. It is defined as the degree to which a modeling language is generally applicable or specifically tailored for a particular task.

The last characteristic discussed in the FRISCO framework is the *arbitrariness* of a language. It is described as the degree of freedom one has when modeling a given domain. The more liberal a language is, the more semantic equivalent models can be made. A modeling language that has zero determinism is said to be *deterministic*.

It is important to note that the three proposed characteristics deal in particular with the modeling language that is used and not so much with the individual models themselves. They are characteristics at the technique level. Expressiveness and determinism are regardless of the domain that is modeled whereas suitability introduces domain and task specificness. The FRISCO report does not go into detail about the evaluation of languages based on the proposed characteristics.

<table>
<thead>
<tr>
<th>Property</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressiveness</td>
<td>the degree to which a given modeling language is capable of denoting the models of any number and kinds of application domains</td>
</tr>
<tr>
<td>Suitability</td>
<td>The degree to which a modeling language is generally applicable or specifically tailored for a particular task</td>
</tr>
<tr>
<td>Arbitrariness</td>
<td>The degree of freedom one has modeling a given domain. The more liberal a language is, the more semantic equivalent models can be made. Deterministic languages have zero arbitrariness.</td>
</tr>
</tbody>
</table>

**Language Quality Criteria According to Sindre**

Sindre (1990) proposes a framework for language quality in conceptual modeling in general in which he distinguishes between two categories of criteria, viz. criteria related to the conceptual basis of the language and criteria related to the external representation of the language. For each of these categories, the following main groups of criteria are defined:
- **Perceptibility** - or, how easy is it for users to grasp the language? With regard to the conceptual basis of the language, the following four criteria are proposed: 1) the concepts should be *natural*, i.e. supporting the way human beings prefer to reason about the domain, 2) the concepts should be easily distinguished, 3) the *number of concepts* should be reasonable, 4) the use of concepts should be consistent throughout the whole set of statements that can be expressed in the language. With regard to the external representation of the language, the following criteria are proposed: 1) the symbols chosen for the concepts should be intuitive understandable, 2) symbols should be easily distinguished, 3) symbols should not represent more concepts in different contexts, 4) strive for symbolic simplicity 5) emphasis in symbols that represent important concepts;

- **Expressive power** - or, what can be expressed in the language? With regard to the conceptual basis, criteria are: 1) concepts should be *general* rather than specialized. 2) all thinkable combinations of concepts into statements should be embodied in the language. 3) language must be *formal, unambiguous* but capable of capturing vague constructs. 4) statements in the language must be *extendible* with other statements providing more details about the first. Expressive power does not apply to the external representation of the language.

- **Expressive economy** - or, how effective can things be expressed in the language? With regard to the conceptual basis, criteria are: 1) kinds of statements that are *frequently used* should be as brief as possible. 2) kinds of statements that are *important* should be as brief as possible. With regard to the external representation, criteria are: 1) symbols that are understood in the context must be omitted. 2) special symbols for concepts that are frequently used or important should be used. 3) limit the number of times a single concept is represented 4) no meaningless symbols 5) diagrams have more potential for expressive economy than texts.

- **Method/tool potential** - or, how easily does the language lend itself to proper method and tool support? With regard to the conceptual basis, criteria are: 1) the conceptual constructs should facilitate *work division*. 2) The conceptual constructs should allow *automatic reasoning*. With regard to the representation, criteria are: 1) the language should lend itself for man-machine communication 2) Avoid features that are exhausting to look at, e.g. blinking. 3) Provide information filtering: ways to hide concepts or to browse through concepts. 4) Fit on standard paper and screen formats.
The criteria mentioned by Sindre (1990) can be summarized as follows:

**Table 3-2: Sindre's Language Criteria Summarized**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Conceptual Basis</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptibility</td>
<td>natural concepts, distinguishable concepts, reasonable number of concepts, consistent use of concepts</td>
<td>understandable symbols, distinguishable symbols, no homonyms, symbolic simplicity, emphasize important symbols</td>
</tr>
<tr>
<td>Expressive Power</td>
<td>general concepts, combinations of concepts, formal and unambiguous, extendibility of statements</td>
<td>NA</td>
</tr>
<tr>
<td>Expressive Economy</td>
<td>brief frequently used or important concepts</td>
<td>omit understood or meaningless symbols, special symbols, avoid synonymy, use diagrams instead of text</td>
</tr>
<tr>
<td>Method/Tool Potential</td>
<td>facilitate work division, allow automatic reasoning</td>
<td>allow man machine interaction, avoid exhausting symbols, information filtering</td>
</tr>
</tbody>
</table>

The Sindre framework is an addition to the three characteristics proposed earlier in the FRISCO framework. Furthermore it offers guidelines for a qualitative modeling language. It lacks precise definitions and ways to measure the proposed characteristics. For example, ‘statements that are important should be as brief as possible’. One could ask oneself, what is ‘important’, what is ‘brief’ and what is ‘as brief as possible’?

### 3.3 Perspectives in Modeling

As said, in order to deal with the complexity of an overall conceptual model of a certain phenomenon, it is common to break down the model in a number of aspect models that each focus on a certain aspect of the modeled phenomenon, i.e. the modeling concepts used are divided over several aspect modeling techniques. This principle is also applied in business process modeling techniques.

In general it is common to make a distinction between those models that focus on the *static* aspects of the system and those that focus on the *dynamic* aspects. Within the field of business modeling and information systems development, it is generally recognized to distinguish three perspectives. These 'classical perspectives' are (Olle et al, 1991): *data, process* and *behavior*. The data perspective (also called
'structural' or 'informational' perspective focuses on the static aspect, process (also 'functional' or 'data flow') and behavior ('event', 'control') on the dynamic aspect. E.g. Wieringa (1998) speaks of a functional perspective, useful pieces of behavior offered to the environment of the system, a behavioral perspective, the way that functions are ordered in time and a communication perspective, the exchange of messages between system and environment. Krogstie (1999) also recognizes the structural, functional, behavioral, communication perspectives and adds the rule perspective, object perspective and role/actor perspective. Below each of these perspectives are discussed together with the possible generic concepts suitable for modeling these perspectives. This discussion is followed by some critical comments.

- **Structural perspective** – This perspective describes the static structure of a business process. The core generic modeling concepts will be called the object class (also called entity, object type, class). The following types of interrelationships between object classes are often distinguished: association, generalization, aggregation. The generalization and aggregation relationships are binary, the arity of the association relationships differs between techniques. All techniques allow the use of binary relationships, most of the allow n>2-ary relationships and some allow unary relationships. Often object classes can be assigned attributes. Example techniques for modeling this perspective are Entity-Relationship modeling (Chen, 1976), Object Role modeling (Halpin, 2001) and UML Class modeling (OMG, 2001).

- **Functional perspective** – This perspective describes the business function. Core generic concepts that can be used to model this perspective are functions (often called ‘processes’), input and output information flows and information stores. Furthermore it is common to order functions in functional decompositions. The most important representative for modeling this perspective is the Data Flow Diagram (Yourdon, 1989).

- **Behavioral perspective** – This perspective conceives a business process as a finite state machine. Starting from an initial state, the state of the business process change by means of state transitions. Core generic concepts are therefore states and state transitions. Sometimes states and transitions are grouped together to form aggregate states and transitions. State transitions diagrams are found in UML (OMG, 2001) but also Petri-net models and other behavioral oriented approaches can be mapped to these state transition models.

- **Rule perspective** – This perspective covers the rules that govern the proper execution of business processes. Each rule has an antecedent and a consequent and is in the form ‘if [antecedent] then [consequent]’. Several types of rules can be distinguished. As argued by Krogstie (1999), the modeling of knowledge by means of rules is not novel, however, it’s application to model business processes is quite recent. Typical core generic modeling concept is the concept of a rule.
Antecedents and consequents are expressions for which e.g. the concepts for modeling the structural perspective can be used. Krogstie mentions the technique ‘Tempora’ as an example modeling technique that covers this perspective. When the consequents of rules are treated as business goals, one could also incorporate goal oriented techniques such as the goal model in Business Modeling with UML (Eriksson et al, 2000) in this perspective.

- **Object perspective** – According to this perspective, business processes are conceived as a collection of objects that influence each other some way or another. It is based on the object oriented modeling paradigm that is well known in the software engineering discipline. Objects have a static as well as dynamic aspect. For modeling the static part, concepts for covering the structural perspective can be used: objects, classes, attributes and relationships. For modeling the dynamic part, objects are assigned operations that can be invoked by other objects. Objects are grouped into classes. The most prominent technique for modeling according to this paradigm is UML (OMG, 2001).

- **Communication perspective** – This perspective conceives a business process as a collection of actors that cooperate within processes and coordinate their work by means of communication. At a low level this communication can be modeled by concepts such as actors and messages that exchange between these actors, e.g. the UML Sequence model. At a higher level, messages between actors can be conceived as steps in communication patterns or conversations as is done in e.g. DEMO (Dietz, 1996).

- **Actor and Role perspective** – According to Krogstie (1999), the actor and role perspective originates from the work on intelligent agents in artificial intelligence. Organizations are conceived as a system of actors that have their own responsibilities with respect to the actions happening in the system. ALBERT is mentioned as a representative modeling technique.

A number of critical comments can be made with regard to the distinction between perspectives as made by Krogstie, in particular concerning overlap and interrelationships between distinct perspectives:

- The rule perspective overlaps with the behavioral perspective, since in the behavioral perspective one is also able to model pre and post conditions of a state transition, comparable to the antecedent and consequent of a rule. The firing of a rule can be conceived as the occurrence of a state transition.

- The communication perspective overlaps with the behavioral perspective. When conversations are viewed as patterns of communication acts or the exchange of messages, than these conversations can be modeled as finite state machines
(behavioral perspective). It simply becomes modeling the behavioral aspect of the communication in an organization.

- The communication perspective partly overlaps with the actor and role perspective since both perspective model actors with responsibilities that act upon each other.

- The object perspective (object orientation) combines the structural perspective and behavioral perspective, it is a combination of existing perspectives.

- The actor and role perspective overlaps with the object perspective. Actors that interact with each other can be modeled as objects interacting with each other according to the object oriented paradigm.

From a workflow-oriented point of view, business process models can be seen as the input process definition of a workflow engine that automatically distributes work through an organization (WFMC, 2003). For such a process definition, the following perspectives should be covered (Aalst, 2002, Aalst et al 2002):

- The Control-flow perspective (or process perspective) describes the control-flow, i.e. the ordering of tasks. It describes activities and their execution ordering through different constructors, which permit flow of execution control, e.g. sequence, choice, parallelism and join synchronization. Activities in elementary form are atomic units of work and in compound form modularize an execution order of a set of activities.

- The Data perspective (or information perspective) describes the data that are used, distinguishing between business and processing data. It describes business documents and other objects that flow between activities as well as local variables of the workflow appearing in pre- and post-conditions of activity execution.

- The Resource perspective describes the structure of the organization and identifies resources, roles and groups. It provides an organizational structure anchor to the workflow in the form of human and device roles responsible for executing activities.

- The Operational perspective (or task perspective) shows the content of the individual steps in the processes. It describes the elementary actions executed by activities, where the actions map into underlying applications through activity-to-application interfaces allowing manipulation of the data within applications.

Clearly, some perspectives have a more specific focus on the use of business process models as a process definition for workflow systems, others however are comparable
to the perspectives discussed earlier. The control flow perspective can be related to the behavior perspective. That is, both perspectives describe how the current state of a process changes and how these changes are restricted. The data perspective is to some extent related to the structural perspective (as mentioned by Krogstie, 1999). Although both perspectives deal with the modeling of information, the structural perspective models information on a higher abstraction level that does not include representational issues such as 'documents' and 'files'. The resource perspective can to a certain extent be related to the actor role perspective since both perspectives capture actors and their responsibilities. The operational perspective is clearly workflow oriented and cannot be compared to the before mentioned perspectives.

A conclusion that can be drawn here is that the attempt to compare and relate different perspectives mentioned in existing literature with each other appears to be difficult and artificial. An important reason for this difficulty is that existing publications on 'perspectives' have quite different underlying ideas concerning what a 'perspective' actually is. There is the desire that perspectives are orthogonal views that one can have on a certain system (e.g. the structural view next to the behavioral view on a system). In practice this is rarely achieved. Perspectives are seen as a description of a system category itself (e.g. the communication perspective to describe the category of communication systems, or an information perspective to describe the category of information systems). Furthermore, perspectives are used to denote a complete modeling paradigm underlying modeling techniques (e.g. the rule perspective denoting a business rules paradigm and the object perspective denoting the object orientation paradigm).

In order to bring clarity to this discussion, a clear distinction has to be made between different system categories, the system perspectives as views that one can have on systems of a certain system category and the basic paradigm underlying these system categories. This distinction will be discussed in chapter 4.

### 3.4 Analysis of Concepts and Conceptual Structures

In order to further study the modeling concepts that modeling techniques offer to model the different system perspectives, it is important to give a precise description of these modeling concepts and their interrelationships. The way to do this is to construct a model of the modeling technique, a so-called meta model.

One way of working to achieve this is to analyze documentation available describing the modeling technique under study. Another way of working is to reengineer existing tools that support a certain modeling technique present in the tool. We will encounter examples of both ways of working. With regard to the way of modeling, two important families of commonly used meta modeling approaches can be
distinguished: the family of Entity Relationship approaches and the family of Fact Modeling approaches.

**Two families of Meta Modeling Techniques**

The *Entity Relationship* family of approaches starts with a publication of Peter Chen (1976), as a graphical notation for specifying information based on the relational model earlier developed by Codd (1970). The basic idea underlying ER modeling is that information can be conceived as a collection of entities and relationships between these entities. Entities can be grouped in entity types and relationships can be grouped in relationship types. Furthermore, attributes can be assigned to entities. The original modeling approach has been extended with the capability to model n-ary relationship types, generalization relationships and to aggregate relationship types into new entity types. Well-known variants are e.g. the Extended ER models and the Information Engineering (IE) approach of James Martin.

![Simple ERD Example](image)

**Figure 3-3: Simple ERD Example**

Rosemann *et al* (1998) and Zur Mühlen (1999) have applied ER models to reengineer workflow systems. They present ER models of the different workflow perspectives on workflow systems. Another ER related meta modeling approach is that of Rossi *et al* (1996) which will be discussed later on in this chapter.

*Fact Modeling* approaches appear in the 1970s. Nijssen (1976) proposed NIAM as a linguistically oriented framework for modeling types of objects and fact types in which these object types play roles. The basic idea underlying Fact Modeling is that information can be conceived as facts that in their turn refer to objects and that these facts and objects can be grouped into fact types and object types. Gradually the modeling approach was extended by e.g. the distinction between non-lexical and lexical object types and by generalization relationships between object types. Amongst others, the following variants have been developed: Object Role Modeling (ORM), developed by Halpin (2001), Fully Communication Oriented Information Modeling FCO-IM (1996) and Predicator Set Modeling (PSM) by Hofstede *et al* (1993).
Various researchers have used Fact Modeling as a meta modeling approach to capture the concepts of modeling techniques. Especially PSM has been successfully applied to create meta models of e.g. Yourdon’s structured analysis method (Verhoef, 1993). In this approach, complex object constructors are used to group the modeling concepts of different aspect modeling techniques. By doing so, also relationships between different aspect modeling techniques can be modeled by means of fact types referring to the complex object constructors.

Both the application of Entity Relationship Approaches as well as Fact Modeling Approaches are to a certain extent suitable for constructing meta models of modeling techniques. The work of e.g. Verhoef (1993) shows that also different aspect modeling techniques can be modeled separately.

Comparison of Meta Models

Once the modeling concepts and their relationships are captured in a meta model, these meta models can be compared amongst each other or compared to a reference ontology. A comparison takes place by mapping the concepts and relationships of one theory to the other theory, as depicted in Figure 3-5. Note that the definition of theory as depicted in Figure 3-5 is reused here.

From a multi perspective point of view, an important precondition for a comparison is of course that both theories cover the same perspective of the system that is modeled. For example, it does not make sense to compare a meta model of a process modeling approach to a meta model of a data modeling approach.

The work of Zur Mühlen addresses the use of Entity Relationship Models to capture the data model underlying several workflow management systems and compares these systems on the basis of these data models. The main result of these
comparisons is that it shows deficits in the workflow systems that are compared to each other. According to Zur Mühlen (1999) and Rosemann et al (2002), the most essential information comes from the comparison of entity types. Furthermore, relationship types, their cardinalities and attributes of entity types can be compared.

Ontological Analysis

The word 'ontology' is used to refer to the philosophical investigation of existence, or being. This investigation may be conducted in three directions: 1) investigation of the concept of being, asking what 'being' means, 2) investigation into what it is for something to exist and 3) concerned with the question 'what exists?' or 'what general sorts of things are there?'. It is common to speak of a philosopher's ontology, meaning the kinds of things they take to exist, or the ontology of a theory, meaning the things that would have to exist for that theory to be true (Craig et al, 1998).

It is the last direction of investigation, asking oneself what general sorts of things there are, that has been applied often the recent years to investigate the kind of concepts that a modeling techniques should consist of to be able to model a certain phenomenon in reality. When a number of theories A are compared to one theory B that serves as a so-called reference ontology, we speak of ontological analysis (Figure 3-6).

![Figure 3-6: Ontological Analysis](image)

Within the field of Information Systems Modeling, one asks oneself what the sorts of things are that an Information System consist of (e.g. the FRISCO approach discussed in Falkenberg et al (1996) or the BWW ontology discussed in this section and in various publications amongst which Green et al (1999)). To a lesser extent, Business Process Modeling techniques have been investigated in the same way, asking oneself what the sorts of things are that a business process modeling technique should contain in order to construct appropriate business process models. An example that will be discussed in this section is the TOVE (TOronto Virtual Enterprise) Ontology (Fox, 1992).
An ontological investigation may result in an ontological framework or ontological view that represents the basic structure of modeling concepts that a modeling technique should contain to be able to model a certain phenomenon. Such a reference model is then compared to the meta model of a certain technique in order to find out the conceptual omissions and redundancy in a certain technique. More often, these ontological analyses or comparisons are carried out less formal as described above, one just roughly compares an existing ontology with a modeling technique’s ontology, without constructing the reference ontology and meta model. Examples of ontological comparisons can be found in Green et al (1999), Opdahl et al (2000) and Milton (2001).

The procedure for comparing the reference ontology to the ontology of the technique that is being evaluated consists of two kinds of mappings:

- **Interpretation Mapping** - A mapping from the set of modeling concepts in a technique to the reference ontology with the purpose of identifying:
  
  - *Excessive modeling constructs* – Constructs in the modeling technique that have no counterpart in the reference ontology. They are present in the modeling technique but are not necessary according to the reference ontology.
  - *Overloaded modeling constructs* – Constructs that have more than one counterpart in the reference ontology, indicating that these constructs have no elementary meaning.
  - *Precise definitions in terms of the reference ontology* - Constructs that are mapped to the ontology have a precise definition in terms of the reference ontology.

- **Representation Mapping** - A mapping from the reference ontology to the set of modeling concepts in a technique with the purpose of identifying:
  
  - *Redundant modeling constructs* - A single construct in the reference ontology that has several counterparts in the modeling language.
  - *Deficits in the modeling language* - Constructs in the reference ontology that have no counterpart in the modeling language.

The different interpretation and representation mappings can be depicted as follows (Figure 3-7).
Wand et al. (1993) have proposed a systematic approach for ontological analysis and evaluation that is representative for the approaches followed by Green et al. (1999), Opdahl et al. (2000) and Milton (2001). They have presented a reference ontology and a procedure for comparing the reference ontology to the technique to be evaluated.

The reference ontology consists of a set of constructs that specify what they believe are the necessary and sufficient constructs to describe the structure and behavior of the real world. These models are based on an ontology defined by Bunge (1977) and are often referred to as the Bunge-Wand-Weber (BWW) model. The BWW model has been used to evaluate the terminology used in data flow models, entity relationship models, object-oriented models, the relational model, NIAM and CASE tools (c.f. Green et al., 1999). The BWW ontology is summarized in the table below.

### Table 3-3: Ontological Constructs in the BWW model (Green et al., 1999)

<table>
<thead>
<tr>
<th>Ontological Construct</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THING</strong></td>
<td>A thing is the elementary unit in the BWW ontological model. The real world is made up of things. Two or more things (composite or simple) can be associated into a composite thing.</td>
</tr>
<tr>
<td><strong>PROPERTY:</strong></td>
<td>Things possess properties. A property is modeled via a function that maps the thing into some value. For example, the attribute “weight” represents a property that all humans possess. In this regard, weight is an attribute standing for a property in general. If we focus on the weight of a specific individual, however, we would be concerned with a property in particular. A property of a composite thing that belongs to a component thing is called a hereditary property. Otherwise it is called an emergent property. Some are properties of pairs or many things. Such properties are called mutual. Non-binding mutual properties are those properties shared by two or more things that do not “make a difference” to the things involved; for example, order relations or equivalence relations. By contrast, binding mutual properties are those properties shared by two or more things that do “make a difference” to the things involved. Attributes are the names that are used to represent properties of things.</td>
</tr>
<tr>
<td>Class</td>
<td>A class is a set of things that can be defined via their possessing a single property.</td>
</tr>
<tr>
<td>Kind</td>
<td>A kind is a set of things that can be defined only via their possessing two or more common properties.</td>
</tr>
<tr>
<td><strong>STATE</strong></td>
<td>The vector of values for all property functions of a thing is the state of the thing.</td>
</tr>
<tr>
<td>Ontological Construct</td>
<td>Explanation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Conceivable State Space</td>
<td>The set of all states that the thing might ever assume is the conceivable state space of the thing.</td>
</tr>
<tr>
<td>State Law:</td>
<td>A state law restricts the values of the properties of a thing to a subset that is deemed lawful because of natural laws or human laws. The <strong>stability condition</strong> specifies the states allowed by the state law. The <strong>corrective action</strong> specifies how the value of the property function(s) must change to provide a state acceptable under the state law.</td>
</tr>
<tr>
<td>- Stability Condition</td>
<td></td>
</tr>
<tr>
<td>- Corrective Action</td>
<td></td>
</tr>
<tr>
<td>Lawful State Space</td>
<td>The lawful state space is the set of states of a thing that comply with the state laws of the thing. The lawful state space is usually a proper subset of the conceivable state space.</td>
</tr>
<tr>
<td>Conceivable Event Space</td>
<td>The event space of a thing is the set of all possible events that can occur in the thing.</td>
</tr>
<tr>
<td><strong>TRANSFORMATION</strong></td>
<td>A transformation is a mapping from one state to another</td>
</tr>
<tr>
<td>Lawful transformation:</td>
<td>A <strong>lawful transformation</strong> defines which events in a thing are lawful. The <strong>stability condition</strong> specifies the states that are allowable under the transformation law. The <strong>corrective action</strong> specifies how the values of the property function(s) must change to provide a state acceptable under the transformation law.</td>
</tr>
<tr>
<td>- Stability Condition</td>
<td></td>
</tr>
<tr>
<td>- Corrective Action</td>
<td></td>
</tr>
<tr>
<td>Lawful Event Space</td>
<td>The lawful event space is the set of all events in a thing that are lawful.</td>
</tr>
<tr>
<td>History</td>
<td>The chronologically-ordered states that a thing traverses in time are the history of the thing.</td>
</tr>
<tr>
<td>Acts On</td>
<td>A thing acts on another thing if its existence affects the history of the thing.</td>
</tr>
<tr>
<td>Coupling:</td>
<td>Two things are said to be coupled (or interact) if one thing acts on the other. Furthermore, those two things are said to share a binding mutual property (or relation); that is, they participate in a relation that &quot;makes a difference&quot; to the things.</td>
</tr>
<tr>
<td>- Binding</td>
<td></td>
</tr>
<tr>
<td>- Mutual Property</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets.</td>
</tr>
<tr>
<td>System Composition</td>
<td>The things in the system are the composition.</td>
</tr>
<tr>
<td>System Environment</td>
<td>Things that are not in the system but interact with things in the system are called the environment of the system.</td>
</tr>
<tr>
<td>System Structure</td>
<td>The set of couplings that exist among things within the system, and among things in the environment of the system and things in the system is called the structure.</td>
</tr>
<tr>
<td>Subsystem</td>
<td>A subsystem is a system whose composition and structure are subsets of the composition and structure of another system.</td>
</tr>
<tr>
<td>System decomposition</td>
<td>A decomposition of a system is a set of subsystems such that every component in the system is either one of the subsystems in the decomposition or is included in the composition of one of the subsystems in the decomposition.</td>
</tr>
<tr>
<td>Level Structure</td>
<td>A level structure defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.</td>
</tr>
<tr>
<td>External Event</td>
<td>An external event is an event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment of the thing, subsystem, or system.</td>
</tr>
<tr>
<td><strong>STABLE STATE</strong></td>
<td>A stable state is a state in which a thing, subsystem, or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event).</td>
</tr>
<tr>
<td>Unstable State</td>
<td>An unstable state is a state that will be changed into another state by virtue of the action of transformations in the system.</td>
</tr>
<tr>
<td>Internal Event</td>
<td>An internal event is an event that arises in a thing, subsystem or system by virtue of lawful transformations in the thing, subsystem or system.</td>
</tr>
<tr>
<td>Well-defined Event</td>
<td>A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known.</td>
</tr>
<tr>
<td>Poorly-defined Event</td>
<td>A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known.</td>
</tr>
</tbody>
</table>

*Core ontological constructs are written down in bold capitals*
Note that the procedure compares 'constructs' instead of individual modeling concepts. Furthermore, one must take into account the reference ontology itself is nothing more than a subjective choice of concepts shaping the way reality is viewed. Milton (2001) states that the reference ontology is not necessarily 'better' than the ontology of the evaluated technique. As a result, reference ontology and ontology of the evaluated technique are interchangeable. A consequence of this interchangeability is for instance that representation mapping not only shows deficits in the modeling language, but could also show excessive modeling constructs in the reference ontology.

Green et al (1999) have evaluated Event Driven Process Chains (Scheer, 1994) using the BWW as reference ontology. They reported the lack of understandability of the BWW constructs, the difficulty of applying the constructs to loosely defined modeling techniques and the limited empirical testing of the implications of BWW evaluations as shortcomings. It is noteworthy that the comparison in Green et al (1999) required a considerable amount of skill of the evaluators and is therefore liable to human judgment and error. This skill is required because the meaning of concepts used must be compared rather than the terms itself. An example of this mapping is the EPC Event Type, which is according to the authors, the same as a BWW state. Furthermore, the comparison showed some deficits in the EPC's such as the lack of a means to define system environments.

Another evaluation based on the BWW model is an evaluation of OPEN Modeling Language (OML) (Firesmith et al, 1998), carried out by Opdahl (1999). Again, the comparison required a considerable amount of skills of the evaluator. The analysis revealed redundant concepts in OML as well as overlap in constructs such as OML-interactions and messages that overlap with OML-states and transitions. It resulted in better definitions of the OML in terms of BWW concepts. Noteworthy is that the OML concept of 'responsibility' has not counterpart in the BWW ontology and is therefore marked as excessive. Indeed there is no place for it in the mechanistic worldview presented in the BWW ontology, but one could ask oneself how excessive it is.

Besides BWW, Opdahl also used the FRISCO framework (Falkenberg et al, 1996) as an alternative reference ontology to evaluate the OML. This evaluation showed that an evaluation of OML using BWW as a reference model leads to another interpretation of OML concepts than the evaluation of OML using the FRISCO framework as a reference model. The reason for this was ascribed to the differing underlying philosophical assumptions of the BWW ontology and the FRISCO ontology. The 'metaphysical realism' of the BWW ontology assumes that things are 'concrete' things in reality, as opposed the 'constructivism' of the FRISCO ontology that sees things as 'conceptions' in the minds of human actors. For example, the OML concept of 'Object' was interpreted as a real thing when mapped to the BWW ontology and interpreted as a conception when mapped to the FRISCO ontology.
In early publications, comparisons between theories and the BWW ontology can be seen as representation mappings, carried out without the use of meta models. The BWW Ontological Constructs as presented in Table 3-3 were used as a checklist to check whether a certain construct is present in the theory that was evaluated, appealing to available literature and documentation on the theory (Firesmith et al., 1998, Opdahl, 1999, Green et al., 1999).

In more recent publications, a meta model of the BWW ontology is constructed and compared to meta models made of the theory that is evaluated. An example of these mappings (Rosemann et al., 2002) is depicted in Figure 3-8, where individual Event Driven Process Chain concepts such as ‘function’ and ‘event’ are mapped to the BWW reference ontology concepts ‘transformation’ and ‘state’.

![Diagram](image)

Figure 3-8: Ontological Analysis Using Meta Models

General criticism that I have concerning the comparison of theories with a reference theory is that the choice of reference theory is subjective. The problem that stands in the way the objective drawing of conclusions is that someone that believes in theory A will take theory A as the reference theory mentioned in Figure 3-7. An interpretation mapping from some other theory B to this reference theory A will show excessive and overloaded concepts in B and a representation mapping will show redundant and deficit concepts in B. However, the other way round, someone that believes in theory B will take B as reference theory. An interpretation mapping from some theory A to B will show excessive and overloaded concepts in theory A. Furthermore, a representation mapping will show redundant and deficit concepts in theory A.

This is also illustrated by the comparison made in Figure 3-8, which can be interpreted in two ways. On the one hand, it is supposed to be an ontological analysis of ARIS - EPC and the UML activity diagram, making use of the BWW
meta model as a reference ontology. On the other hand, it can be seen as a(n incomplete) comparison between three theories. The later interpretation, which I prefer, gives rise to the more fundamental questions that should be asked: Based on what underlying way of thinking or paradigm can we compare these three theories? Why is it possible to map e.g. EPC functions to BWW transformations and not for instance to the BWW state concept? And what if two different theories are based on different paradigms, other than a mathematical, discrete dynamic systems paradigm, can we still compare them or are they incommensurable (c.f. Kuhn, 1962)?

The reference meta model used in the above-mentioned comparison is a simplification of Bunge’s way of thinking to shape reality. A simplification as depicted in Figure 3-8 leaves more interesting questions unanswered, such as how exactly lawful state space and lawful transitions in this state space are modeled.

Analysis of Patterns
The same interpretation and representation mappings that can be carried out for single concepts can also be carried out for ways that concepts can be combined in so-called patterns. By doing so, excessive, overloaded, redundant and deficit patterns can be recognized.

![Diagram of Theory and Reference Theory](image)

Figure 3-9: Comparison of Patterns

The work of Aalst et al (2002) is an example where especially representation mappings are carried out to identify deficit patterns in a particular theory. The reference theory consists of a number of patterns, being abstractions from a concrete form that keeps recurring. During the representation mapping from the reference theory to the theory, it is analyzed whether a theory is able to produce models that can capture these patterns. This way, deficits in a theory can be brought to light. The work of Aalst et al (2002) focuses in particular on the evaluation of the control flow in workflow languages implemented in (commercial) workflow tools.
The following patterns constitute the reference theory:

- **Basic Control Patterns** – These patterns concern the elementary aspects of a control flow, including the sequencing of activities, the execution and synchronization of parallel subprocesses, the exclusive choice between alternative subprocesses and the simple merge of these subprocesses.

- **Advanced Branching and Synchronization Patterns** – These patterns concern more advanced patterns for branching and synchronization, e.g. patterns where one or more alternative subprocesses can be selected and synchronized or merged in various ways.

- **Structural Patterns** – These patterns concern iteration and termination of (sub)processes.

- **Multiple Instances** – These patterns concern the ability to make multiple instantiations of subprocesses during the execution of a process. Several variants can be distinguished based on e.g. whether or not the number of multiple instances is known during design or run time or whether synchronization is required.

- **State-based Patterns** – These are patterns that can only be modeled when the state of a process is explicitly modeled. An example is the difference between an explicit and an implicit (exclusive) choice. During an *explicit* choice, a decision is taken which path to follow in the remainder of the execution of a process, based on information available at that time. Most modeling languages allow the modeling of this kind of choice. During an *implicit* choice, the choice is postponed. One of the alternative paths that may follow the choice will ‘claim’ the execution when certain environmental conditions are met. This last pattern can only be modeled when states are made explicit.

- **Cancellation Patterns** – These patterns concern the cancellation of activities. An activity could cancel the execution of another activity or a complete process.
The workflow patterns proposed by Aalst *et al.* (2002) have proven to be a useful instrument for evaluating the expressiveness of workflow languages implemented in workflow tools. Since workflow tools are supposed to support the execution of all thinkable low-level control flow details that may occur in practice, it is important that they are expressive enough to capture all thinkable patterns. What can be concluded is that a business process modeling technique that is used to produce complete, detailed, executable process specifications should be able to model the patterns mentioned above. On the other hand, when process models are e.g. used to offer insight in less structured processes or abstract from execution details, it could be questioned whether such a technique should be able to model all these patterns. Clearly the level of expressiveness depends on the purpose for which modeling techniques are used.

### 3.5 Quality Measurement

**Complexity Metrics According to Rossi *et al.***

The application of complexity metrics, as proposed by Rossi *et al.* (1996), is a systematic approach for measuring the properties of modeling techniques. The values for these complexity metrics are computed on the basis of the technique's meta model, which holds the promise of being straightforward and free of human judgment. Contrary to the frameworks of Krogstie and Sindre, discussed above, the focus in this approach is more on the measurement procedures and less on the interpretation of these measures with respect to the method's quality. The two components of the approach are the OPRR method modeling language and a suite of independent and aggregate metrics based on this language. Both components will be described below, followed by a short discussion.

The OPRR (Object, Property, Relationship, Role) method modeling language (Smolander, 1990) consists of a formal definition and graphical representation. The discussion will be limited to the formal definition which is a tuple \( M = \langle O, P, R, X, r, p \rangle \) consisting of four finite sets and two mappings:

- **A set of object types** \( O \). Instances of these object types are rather loosely defined as 'things which exist on their own', the reason behind this definition is presumably the ability to distinguish objects from properties, relationships and roles that do not exist on their own. Object types themselves are not defined;

- **A set of property types** \( P \). Properties are defined as describing or qualifying characteristics associated with object types, relationship types or role types;

- **A set of relationship types** \( R \). A relationship is an association between two or more objects;
- A set of role types $X$. A role is a name given to the link between an object and its connection with a relationship;

- A mapping $r: R \rightarrow \{x | x \in \wp (X \times (\wp (O) \smallsetminus \{\emptyset\})) \land |x| \geq 2\}$. In the symbol ‘$\wp$’ denotes a powerset, a set placed between ‘$|$’ symbols denotes the cardinality of the set. Translated into colloquial language, the mapping $r$ maps a relationship type to a member of the set of powersets of role types and powersets of objects. For instance, given that $O = \{\text{Class}\}$, $R = \{\text{Generalization}\}$, $X = \{\text{Super_class}, \text{Sub_class}\}$, an example element of the mapping $r$ could be as follows: $<\text{Generalization}, \{<\text{super_class}, \{\text{Class}\}\}, <\text{sub_class}, \{\text{Class}\}\}>$;

- A partial mapping $p: \{O \cup R \cup X\} \rightarrow \wp (P)$. This mapping assigns the properties to object types, relationship types and role types.

The table below shows the complexity measures that are based on the OPRR language that is defined above. Independent and aggregate measures are distinguished as well as technique level and method level measures. Independent measures measure a single characteristic of a technique or method, whereas an aggregate measure combines one or more independent measures in an overall measure of complexity.

### Table 3.4: Metrics Suite

<table>
<thead>
<tr>
<th>Technique Level Independent Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
</tr>
<tr>
<td>$</td>
</tr>
<tr>
<td>$</td>
</tr>
<tr>
<td>$P_O (M_T, o)=</td>
</tr>
</tbody>
</table>

53
\[ \mathbb{P}_O(M_T) = \frac{1}{|O_T|} \sum_{o \in O_T} P_o(M_T, o) \]

The average number of properties per object type shows the average number of properties per object type for a given technique.

\[ P_R(M_T, e) = |p_e(e)| + \sum_{x \in \text{r}_e(e)} |p(\text{role}(x))|, \ e \in R_T \]

The number of properties of a relationship type and its accompanying role types shows for a given relationship how many properties it and its accompanying role types have. \( \text{role}(x) \) is a function that returns the role associated with an \( x \in \text{r}(e) \). In line with the example above, given that \(<\text{super}_\text{class}, \{\text{Class}\}> \in \text{r}(\text{Generalization}) \), \( \text{role}(<\text{super}_\text{class}, \{\text{Class}\}>) \) would yield: \{\text{Class}\};

\[ \mathbb{P}_R(M_T) = \frac{1}{|R_T|} \sum_{e \in R_T} P_R(M_T, e) \]

The average number of properties per relationship type shows the average of the above metric for all relationship types in a technique.

\[ R_O(M_T, o) = \{ e \in R_T | o \in \bigcup_{x \in \text{r}_T(e)} \text{objects}(x) \}, \ o \in O_T \]

The number of relationship types per object type shows how many relationships are connected to a given object type. \( \text{objects}(x) \) is a function that has \( x \in \text{r}(e) \) as signature and returns the objects associated with an element \( x \) of \( \text{r}(e) \). In line with the example above, given that \(<\text{super}_\text{class}, \{\text{Class}\}> \in \text{r}(\text{Generalization}) \), \( \text{objects}(<\text{super}_\text{class}, \{\text{Class}\}>) \) would yield: \{\text{Class}\};

\[ \mathbb{R}_O(M_T) = \frac{1}{|O_T|} \sum_{o \in O_T} R_O(M_T, o) \]

The average number of relationship types per object type shows the average of the above metric for all object types in a technique.

Technique Level Aggregate Measures

The individual measures above describe the individual characteristics of a technique. Aggregated measures are used to measure the overall complexity of a technique, based on these individual measures above.

\[ C(M_T, o) = \frac{P_o(M_T, o)}{\sum_{x \in \text{r}_T(e)} P_R(M_T, e)} \]

The division of work is a quotient between the number of internal properties and the number of external properties of a given object type.

\[ \overline{C}(M_T) = \frac{1}{|O_T|} \sum_{o \in O_T} C(M_T, o) \]

The average division of work is the average of the division of work for all object types in a technique.

54
The total complexity of a technique is the length of a three dimensional vector consisting of the number of object types, relationship types and property types of a technique. The use of a three dimensional vector makes it also possible to analyze the direction of the vector;

Method Level Independent Measures

In the work of Rossi et al (1996), methods are treated as collections of individual techniques. Based on this definition of a method, method level independent measures are defined. In general, these measures are the sum of technique level independent measures for each technique in a method.

\[ |O_M| = \sum_{T \in M} |O_T| \]

The count of object types per method is the sum of the count of object types per technique, for all techniques in a method;

\[ |R_M| = \sum_{T \in M} |R_T| \]

The count of relationship types per method is the sum of the count of relationship types per technique, for all techniques in a method;

\[ |P_M| = \sum_{T \in M} |P_T| \]

The count of property types per method is the sum of the count of property types per technique, for all techniques in a method;

Method Level Aggregate Measures

Method level aggregate measures show the overall complexity of a method. In general, they sum the technique level aggregate measures for each technique in a method.

\[ C(M) = \frac{1}{|O_M|} \sum_{T \in M} \sum_{O_T} C(M,T,O) \]

The average division of work per method is the sum of the average division of work per technique, for all techniques in a method;

\[ C'(M) = \sqrt{|O_M|^2 + |R_M|^2 + |P_M|^2} \]

The total complexity per method is the length of a three dimensional vector consisting of the number of object types, relationship types and property types of a method.

The work of Rossi et al (1996) provides a well-founded approach to the measurement of the properties of modeling techniques, based on the meta model of the concepts used in a modeling technique. The focus is limited to the measurement of complexity. Complexity metrics are presumed to be measures for the comprehensibility and ease of use properties of a modeling technique. Properties such as the relevance of certain concepts to model certain domains are therefore not provided. As opposed to the quality framework of Krogstie, the Rossi approach focuses on the technique level rather than the individual model level.
The approach is not so 'free of human judgment' as one might think. Although the metric suite provides objective measures that compute values out of the meta model without any human judgment, the construction of the meta models on which this measurement is based is still an activity that leaves room for human interpretation of the technique under study. Also the choice of a meta modeling language that distinguishes object types, properties and relationships contributes to the freedom of choice that a modeler has. The distinction between property and object type as well as the distinction between object type and relationship is not always clear. It can be concluded that sources for human interpretation and judgment are pushed aside to the construction of the meta models rather than eliminated.

In the approach, a method is considered to be nothing more than a collection of techniques. This definition has two flaws. Firstly, a method is an interrelated structure of techniques rather than a collection of separate techniques. E.g. UML Class diagrams and Use Case diagrams are interrelated rather than separate techniques because they share certain modeling concepts (OMG, 2001). The metrics suite that is presented cannot take into account these interrelationships, since the method modeling language is not able to capture them. This first flaw is admitted by Rossi et al (1996). Secondly, a method is even more than an interrelated structure of techniques. In general (see chapter 2) a method is supposed to be a way of working, a structure of activities (each supported by a technique) that have to be carried out in order to solve a particular class of problems, possibly in a certain order. E.g. it is often said that the UML is not a method as opposed to e.g. the Unified Process, but a collection of interrelated techniques because it lacks a way of working, an ordering of modeling activities. Because of the two flaws mentioned above, the focus of the method level measures is rather limited.

The use of power sets in this work is incorrect, see the example used for the definition of the mapping \( r \), it contains the same element twice. Contrary to classical set theory, the power set that is used in Rossi et al (1996) may contain an element more than once, see also Rossi et al (1996) pg.149, the set \{Class, Class\}. Usually multi-sets or bags are used in this context. These constructs are mappings from an element to the number times the element occurs in the 'set'.

**Inquiring and Experimental Approaches**

Some qualities are very difficult to quantify by means of the type of metrics discussed above. So-called inquiring approaches make use of experts that are able to recognize quality rather than to measure it using objective metrics. Such an approach can be found e.g. in Moody et al (1994). In this approach, example data models are reviewed by experts that give rates ranging from 0 (extremely bad) to 10 (extremely good) on a selected number of quality aspects. According to Moody et al (1994), accuracy and reliability of measurement is improved by a clear metric definition, an appropriate reviewer population and appropriate reviewer expertise.
The above-mentioned approach concerns the evaluation of individual models rather than modeling techniques. Extending this approach to the evaluation of techniques instead of models makes the approach even more complex. An additional expert (or group of experts to maintain objectivity) has to construct the model using the technique that is to be evaluated. The constructed models are in their turn reviewed by other experts. Also a possible bias of experts towards a certain modeling technique forms a potential hazard.

Also model experiments can be thought of as source for measurement of the quality of modeling techniques. The basic principle behind this type of measurement is to set up an experiment where a group of modelers constructs models using a modeling techniques and a particular modeling case. Afterwards, the models are judged on their quality by means of formal or inquiring metrics. Bajaj (2002) reports from an experiment where subjects where asked to answer questions about certain conceptual models to understand the readability of those conceptual models. These experiments are measurements on the individual model level rather than measurement on the quality of the techniques used.

Compared to measurement of the quality of models, measurement of the quality of the modeling process is a rather underdeveloped area. Maier (1999) proposes a number of experimental metrics in the context of data modeling that can easily be applied to modeling in other areas:

- $P_{NEW}$ – This metric divides the number of newly developed entity types, relationships and attributes modeled by the number of man-hours spent.

- $P_{CHANGE}$ – This metric divides the number of changed entity types, relationships and attributes by the number of man-hours spent.

- Variability – Finally, variability is a measure for the speed with which a model changes over time. It is calculated by dividing the number of changed elements by the total number of elements multiplied by the time spent.

### 3.6 Conclusions

Based on the literature survey, a number of conclusions regarding the evaluation of the quality of modeling techniques can be drawn.

First of all one must clearly distinguish between the quality of modeling techniques and the quality of models that are the result of applying these techniques. Most of the frameworks presented in this chapter focus on the quality of individual models (Krogstie et al, 1999, Becker et al, 2000), whereas the aim of this thesis is to understand the quality of modeling techniques. For some quality properties the
distinction between technique level and model level is clear, e.g. syntactic correctness can only apply to individual models. Other quality properties such as complexity and comprehensibility, apply to both technique and model level and consequently a clear distinction has to be made.

A comparison between the evaluation frameworks of e.g. Batini et al (1992), Krogstie et al (1999) and Becker et al (2000) shows that there is a large diversity in the identification, interpretation and classification of quality properties. Some flaws in the interpretation and grouping of quality properties have been identified. The framework that will be proposed in the next chapter has to overcome these shortcomings.

The same that was concluded regarding the diversity in the identification and interpretation of quality properties holds even more for the identification and interpretation of modeling perspectives. As said earlier, the main reasons for this diversity is the lack of a clear consensus on what a perspective is. Although there is the desire that perspectives are orthogonal views that one can have on a certain system, they are seen a description of a system category itself or they are used to denote a complete modeling paradigm underlying modeling techniques. The framework that will be proposed in the next chapter has to give a clear account on what a system perspective is and should go beyond an arbitrary selection and grouping of these perspectives.

Both Fact Modeling as well as Entity Relationship Modeling are a suitable starting point for the construction of meta models of modeling techniques, although not specifically tailored for this purpose. For the meta modeling language to be proposed in this thesis, it should be reconsidered whether certain detailed modeling constructs (e.g. the specific modeling of attributes, label types, et cetera) are necessary and whether certain modeling constructs should be added (e.g. regarding the modeling of relationships between different perspectives).

Meta models can be compared to each other or to a reference meta model. The choice of reference meta model is subjective and therefore comparison to a reference meta model is the same as comparing arbitrarily chosen meta models to each other. More fundamental questions are unanswered: Based on what underlying way of thinking or paradigm can we compare these meta models? What if two different theories are based on different paradigms, other than a mathematical, discrete dynamic systems paradigm, can we still compare them or are they incommensurable? The BWW meta model (Rosemann et al, 2002) is a simplification of Bunge’s way of thinking to shape reality (Bunge, 1977, Bunge, 1979) that leaves more interesting questions such as how exactly lawful state space and lawful transitions in this state space are modeled, unanswered.
The workflow patterns of Aalst *et al* (2002) provide a useful checklist to verify the expressiveness that is required for the construction of complete, detailed, executable process specifications. This type of evaluation is very purpose (workflow) dependent. When process models are e.g. used to offer insight in less structured processes, abstracting from execution details, it could be questioned whether this expressiveness is required.

The formal measurements provided by Rossi *et al* (1996) are promising instruments for the qualification of the quality properties of modeling techniques. The measures of Rossi are based on a meta model that does not take into account the different aspect models and their interrelationships. The meta modeling language together with the metric suite that will be developed in this thesis should take these relationships into account.

In the following three chapters to come, propositions will be made to overcome the shortcomings that were reported on the basis of this literature research.
Part II

Proposition
Chapter 4: A Proposal for Understanding the Quality of Business Process Modeling Techniques

4

A PROPOSAL FOR UNDERSTANDING THE QUALITY OF BUSINESS PROCESS MODELING TECHNIQUES

4.1 Introduction

The next three chapters of this thesis comprise the part that describes the proposition of a framework for the evaluation of business process modeling techniques, which will be named the Q-Me (Quality based Modeling Evaluation) framework. This proposition encompasses three elements, which will be described in three subsequent chapters:

- As concluded in chapter 3, existing frameworks focus on the individual model level rather than the technique level or fail to make a clear distinction between both. Furthermore, the way that properties are defined lacks clarity. The system of quality properties that will be proposed in this chapter integrates and extends relevant existing work, taking into account the mentioned shortcomings.

- Another conclusion drawn out of the literature survey was that most existing frameworks lack the means to give a precise description of the modeling language that is to be evaluated. Especially the description of state-of-the-art multi modeling techniques is an underdeveloped area. In order to allow a precise description of the modeling techniques under evaluation, the C-Me description language will be proposed in chapter 5. It allows the modeling of concepts, their relationships to reality and the grouping of generic concepts into aspect modeling techniques;

- Based on the previously introduced description language and quality properties, a measurement scheme is proposed in chapter 6. This scheme describes procedures for measuring the quality properties, making use of the description language.
4.2 Modeling Quality

It is not easy to provide the term 'quality' with an exact definition. Often, the term 'quality' is used as a synonym for 'property' or 'attribute', for instance when it is mentioned that someone possesses the quality of being nice. One uses the term to denote that someone plays a certain role, for instance when we say that a professor acts in the quality of promoter. In this thesis, the meaning of the term is restricted to what Webster's dictionary calls 'degree of excellence', a measure that places the things that are measured on a scale that ranges from 'good' to 'bad'.

After delimiting the meaning of quality as a measure of something being good or bad, the next problem is to define what 'good' and 'bad' actually is in general and in the context of the phenomenon of which the quality is determined. For one, 'good' means 'conformance to requirements' and for another 'fitness for use' (c.f. Flood, 1993).

It is interesting to note that quality is a property of things that perform functions for us. Searle (1995) argues that human beings are capable of assigning functions to things in reality. We can either assign functions to things that already exist, such as the use of stones to create fire with or create things to perform functions for us such as hammers to hit nails with and cars to drive in.

Along with the ability of assigning functions to things comes the ability to access how well or badly things perform these functions for us, i.e. the quality of the thing. It does not make sense to assess the quality of physical things that have not been assigned a function: it is senseless to ask oneself whether a mountain is a good or bad mountain or whether something is a good or bad stone: they simply are mountains and stones. However, when asking the same questions with respect to things that perform functions, it suddenly makes sense: it makes sense to ask whether a hammer is a good or bad hammer. As soon as we assign a stone the function to make fire it suddenly makes sense to speak of a good or bad firestone.

Quality is a property of things that perform functions for us. Without the existence of human beings there would be no functions assigned to things and thus no quality. Quality is the extent to which a certain thing performs its assigned function. To measure quality is to assess how well a certain thing performs the function that was assigned to it, thereby placing it on a scale that ranges from good to bad. This general definition unites most of the existing definitions that were mentioned before such as 'fitness for use' and 'conformance to requirements'.

A common approach for understanding and evaluating the quality of a phenomenon is, given the function that the phenomenon is to perform for us, to subdivide quality in a number of quality properties. This approach can also be found in the ISO definition of quality: "the total of properties and characteristics of a product or
service that are relevant for satisfying specific requirements and obvious necessities”. E.g., a car is a good car when it is safe, reliable, clean, fast, economic, etc. This approach is also taken in existing frameworks for understanding modeling quality as discussed in chapter 3. A good model is e.g., consistent, comprehensible, relevant, etc. This approach is taken as a starting point for the proposal for understanding quality that is developed in this thesis.

Existing frameworks for understanding quality do not always make a clear distinction between the quality of modeling techniques and the quality of the models that are the result of the application of those techniques. However, it is obvious that there is distinction between the function that has been assigned to a business process model and the function that is assigned to a business process modeling technique. In order to obtain a clear understanding of the quality of modeling techniques it is important to draw a clear distinction between the quality of the two and to understand the relationship between the properties of a modeling technique and the properties of a model.

The goal of this chapter is to create an understanding of the quality of business process modeling techniques. However, in order to obtain such an understanding, attention has not only to be paid to the quality of these techniques but also to the quality of models (the result of the application of modeling techniques) and the way that these phenomena influence each other's qualities.

Figure 4-1: Model, Technique and Tool Interrelationship

In fact, a search after the quality of modeling techniques starts with an understanding of the quality of models since after all, the aim of the application of modeling techniques together with the support offered by tools is to construct good models. The foundation of the quality of modeling techniques, i.e. the question what a good model is, will be elaborated in section 4.3. A modeling technique offers the generic concepts and diagrammatic notations to aid the construction of good models. The way that these constituents influence the quality of models will be discussed in section 4.4, the quality of techniques.
4.3 Quality of Models

As said, the cornerstone of a framework for evaluating the quality of modeling techniques is the definition of the quality of the models that result from the application of a technique and the support by a tool. The function of a model is to create an abstraction from a phenomenon in reality, suitable for the purpose at hand.

Before I proceed with a detailed elaboration on model quality I would like to make a short comment here on the ambiguous use of the term model quality especially regarding business models, originating from a difference in viewpoint on the definition of what a good business model is. For example, one could ask oneself, 'Is the business model of Amazon.com a good business model?' or 'Is the business model of Apple better than that of Microsoft?'. These are not the type of questions answered in this thesis since these questions do not refer to the quality of the model itself but to the quality of the business situation described in the model. In this thesis a model is 'good' when it corresponds to the phenomenon modeled, regardless of how 'well' or 'badly' that phenomenon works or is supposed to work.

In line with both an empirical study by Wang et al (1994) and theoretical viewpoints such as e.g. that of Wand (2000) which will be elaborated briefly below, I propose two basic characteristics that a good model should posses: a good model is a model that is on the one hand a correct and on the other hand a useful description of a phenomenon in reality for the purpose at hand.

Wang et al (1994) confronted a population of modeling experts with more than 300 keywords related to modeling quality, such as relevance, validity, comprehensibility, compactness, formality, etc. and asked the experts to give marks to these keywords that indicate their importance for assuring modeling quality. It turned out that the keywords that were found most important can be grouped into two groups: those that were related to the correctness of the model and those that were related to the usefulness of a model. Correctness ranges from a model's correspondence to the phenomenon that is modeled to its correspondence to syntactical rules of the modeling language and usefulness can be seen as the model being helpful for the specific purpose at hand, e.g. the implementation of a workflow system. Also the four main characteristics of a good model that Wand (2000) mentions, viz. completeness, consistency, elegance and predictiveness can be grouped into the correctness of a model (complete, consistent) and the usefulness (predictiveness). Elegance is a measure for stating as much as possible about a phenomenon with as few as possible statements.

The correctness property of a model is independent of the purpose for which the model is used whereas the usefulness property depends on the purpose for which the model is constructed. A model of the electrical system of a car can be perfectly
correct, that is, in correspondence with reality and using the concepts described by the modeling technique but is totally useless for the purpose of fixing an oil leakage. For determining the usefulness of business process models, the purposes for which business process modeling are used (Figure 1-1) should be taken into account. The picture below shows the purpose dependency of the usefulness property for e.g. ISO certification, as opposed to the purpose independence of the correctness property.

![Diagram showing purpose dependency of usefulness]

**Figure 4-2: Quality and Purpose**

A common starting point for defining the quality of a conceptual model is anchoring it in the linguistic properties of the modeling language, of which syntax and semantics are most often applied (e.g. in Reingruber et al, 1994). A broader approach, based on semiotics rather than linguistics, is to derive the quality of a model from the semiotic ladder of Stamper (1973) as was done by Krogsstie et al (1999). Next to syntax and semantics, the semiotic ladder also includes empirics and pragmatics.

However, as argued before, in my opinion, the term semantic quality is often used in the wrong manner, mistakenly denoting the extent to which a model corresponds to the phenomenon that is modeled. In my opinion, a more correct interpretation of the term semantic quality is to say that a model is semantic correct if each statement in it has a precise, unambiguous meaning, regardless of whether it is in accordance with that what is modeled. E.g. the statement 'all ravens are white' is semantic correct since it has meaning, however, it is not valid since it can easily be falsified. Pragmatic quality deals with the practical use of language and intended effect of it. The intended effects of the statements are not interesting and there is no way that a modeling technique can influence these effects. In most cases, the statements are what Habermas (1984) would call 'constantiva', i.e. claims to truth, and what Searle (1969) would call assertives. Since it makes no sense to study the intended effect of statements in a business process model, pragmatic quality is not studied in further detail.
Semantic correctness is a condition for determining the validity, the extent to which statements in a model correspond to that what is modeled. Therefore, in the quality properties that will be proposed here, I would like to make a further distinction between internal correctness (empirical, syntactical, and semantic quality) and external correctness (validity). This further distinction is depicted below and will be clarified in the succeeding paragraphs.

Figure 4-3: Model Quality further elaborated

**Empirical Correctness**
Having defined correctness and usefulness as the two main characteristics of a good model, let us first dwell upon the correctness property of a model. Before defining the subsequent model qualities, let us recall the distinction that was drawn between aspect model and whole model in section 2.3: An aspect model is part of a whole model and focuses on a particular aspect of the modeled phenomenon. Together, all aspect models of a certain phenomenon constitute a coherent whole model of the phenomenon. Examples of aspect models within UML (OMG, 2001) are e.g. a Class model, Use Case Model and Sequence Model of a certain software system. Together these aspect models are said to constitute a whole model of the software system.

Figure 4-4: Example Process Model
Figure 4-4 shows an aspect model that focuses on a certain perspective of a business process. It will be used as an example throughout this section. There are two types of propositions that are expressed in this diagram: 1) propositions with regard to the existence of actions and 2) propositions with regard to the precedence of these actions. The technique that was used to construct the model in Figure 4-4 prescribes that propositions of the first kind are denoted by boxes containing the name of the action that is known to exist and that propositions of the second kind are denoted by arrows pointing from one box to another expressing that the action that is represented by the box that the arrow points to is preceded by the action that is represented by the box that the arrow departs from. Furthermore, such a technique could prescribe that the precedence relationship is transitive, i.e. if A precedes B and B precedes C it follows that A precedes C, and it could prescribe that the precedence relationship is asymmetric, i.e. that if A precedes B, it follows that B cannot precede A. In total the knowledge contained in this diagram consists of 9 propositions. Figure 4-4 shows these propositions expressed in natural language.

The following property concerns the empirical correctness of a model:

- **Empirical correctness** - An aspect model is empirical correct if and only if each proposition contained in it is expressed using the graphical notation (shapes, boxes, text, etc.) that is prescribed by the modeling technique. A whole model is empirical correct if and only if all its constituting aspect models are empirical correct;

![Diagram](image)

**Figure 4-5: Empirical Incorrect Model**

Figure 4-5 shows an empirical incorrect model. It contains graphical elements that are not prescribed by the language. Therefore it cannot be derived what kind of propositions these graphical elements express. Models that are empirical correct may vary with respect to their syntactic correctness which is defined as follows:
Syntactical Correctness

A model is syntactic correct when it is in accordance with the syntactical rules prescribed by the technique. It encompasses a number of properties: syntactical correctness (an individual proposition being in accordance with the syntax of the language), consistency (a set of propositions that does not contradict each other) and well-formedness (a set of propositions being in accordance with the well-formedness rules of the technique). The properties will be discussed in more detail.

- **Syntactic correctness** - An aspect model is **syntactic correct** if and only if each proposition contained in it is expressed according to the syntax of the modeling language that is prescribed by the modeling technique. A whole model is syntactic correct if and only if all its constituting aspect models are syntactic correct;

![Diagram](image)

**Figure 4-6: Syntactic Incorrect Model**

Figure 4-6 shows a syntactic incorrect model. Although the model uses the graphical elements that are prescribed by the technique, i.e. is empirical correct, the way that these elements are combined to form complex propositions is incorrect. Each arrow should be drawn between two boxes.

With regard to a model being in accordance with the syntactical rules that are prescribed, I would like to add two properties concerning syntactical quality, being consistency and well-formedness.

- **Intra model Consistency** - An aspect model is **intra model consistent** if and only if there are no two propositions contained in it that contradict each other. A whole model is intra model consistent if and only if all its constituting aspect models are intra model consistent;
Figure 4-7: Intra Model Inconsistent Model

Figure 4-7 shows a model that is intra model inconsistent. Although all propositions in it are expressed in accordance with the rules that are prescribed by the technique, it is incorrect because it contains two propositions that contradict each other: 'write down order' precedes 'copy order' and 'copy order' precedes 'write down order'. This is not in accordance with the prescribed rule that if A precedes B, it cannot be the case that B precedes A. With respect to multi modeling techniques, inconsistencies may also occur between propositions in different aspect models. Such a model is said to be inter model inconsistent. Inter model consistency is defined as follows:

- **Inter model Consistency** - A whole model is *inter model consistent* if and only if there are no two propositions (one proposition in one aspect model, one proposition in the other aspect model) that are contradicting each other. Inter model consistency is not defined for aspect models;

In order to clarify this property, Figure 4-8 shows a whole model that consists of a UML collaboration model and a UML sequence model. Both models allow the modeling of the exchange of messages between objects. In the collaboration model we see that the message 'deliver' is sent from operator to delivery boy. In the sequence model we see that the same message is sent the other way round, from delivery boy to operator. This whole model is inter model inconsistent because according to the UML consistency rules, the sender and receiver of a single message should be the same throughout all aspect models.
Another type of inconsistency between aspect models is depicted in Figure 4-9. This whole model is not so much inconsistent because of a violation of consistency rules that are explicitly and precisely formulated by the modeling language but more because of the violation of implicit rules that exist because of a lack of formalization in the technique. The figure shows a whole model that consists of the previously discussed process model as well as a UML like class model of which the modeling technique is assumed known to the reader.

The whole model is inter model inconsistent because out of the process model it can be derived that the cook receives orders whereas the delivery boy receives order copies. It is very likely that the class model contradicts this. When orders are
passed to cooks and order copies to delivery boys, it seems more likely that delivery boys own the order copies and cooks own the orders.

It must be said that this second kind of inconsistency lacks any formal measure because it requires an interpretation of the model. Therefore it can only be said that it is very likely that this whole model is inter model inconsistent instead of being absolutely certain. Because practice shows that these inconsistencies often appear in process models, it is worthwhile to mention it here and investigate it later, see e.g. Business Modeling with UML in chapter 7.

An aspect modeling technique might impose restrictions on the well-formedness of a model. An aspect model that does not obey these restrictions is called intra model mal-formed. An example of a well-formedness rule is the restriction that every process model should contain exactly one begin state and one end state. It is thinkable that every single proposition in the model is syntactically correct, i.e. formed according to the prescribed syntax but that it misses propositions that are prescribed by the technique.

- **Intra model well-formedness** – An aspect model is intra model well-formed if it complies to the well-formedness rules imposed on it. A whole model is intra model well-formed if all its constituting aspect models are intra model well-formed.

The same that can be said about well-formedness within a single aspect model can be said about the well-formedness of a whole model. A whole model that does not obey restrictions on the well-formedness of a whole model is called inter model mal-formed. An example is a restriction that every object class in a UML class model should be further specified by a UML state transition model.

- **Inter model well-formedness** - A whole model is *inter model well-formed* if and only if it complies to the well-formedness rules imposed on the inter model relationships;

**Semantic Correctness**

Models that are empirical and syntactic correct may vary with respect to their semantic correctness, the extent to which the truth of propositions, i.e. being in accordance with the phenomenon that is modeled, can be tested. Propositions may concern physical reality or social reality. Propositions that concern physical reality must be testable by means of observations, propositions that concern social reality must be testable by checking the consensus on them. E.g. the proposition "the action 'pick up phone' precedes the action 'write down order'" in Figure 4-4 is semantically correct when its truth can be tested by checking consensus about it amongst the people involved.
Evaluation of Business Process Modeling Techniques

- **Semantic correctness** - An aspect model is *semantic correct* if and only if each proposition contained in it has a precise meaning, i.e., it being in accordance with the phenomenon that is modeled can be tested true or false. A whole model is semantic correct if and only if all its constituting aspect models are semantic correct;

**Validity**

Finally, validity is the last property concerning (in particular the external) correctness of a model:

- **Validity** - An aspect model is *valid* if and only if all propositions in it are in accordance with the phenomenon that is modeled. A whole model is valid if and only if all its constituting aspect models are valid.

![Figure 4-10: Invalid Model](image)

The figure above shows an example of a model that is presumably invalid. Although it is syntactically correct, semantically correct and might be valid, it is more likely that 'write down order' precedes 'copy order'.

The notions empirical, syntactic, semantic correctness and finally validity are ordered by a conditional relationship: empirical correctness is a precondition for assessing a models syntactical correctness, syntactical correctness in its turn is a condition for determining the semantic correctness of a model, and so on. In other words, given all correct or incorrect models of a same phenomenon in reality, constructed with the same technique, it can be said that the set of syntactic correct models is a subset of the set of empiric correct models. The set of semantic correct models is a subset of the set of syntactic correct models, and so on. In the ideal situation, the set of correct models on the highest semiotic level contains just one element. In that case, the applied technique is said to be deterministic.
Usefulness

Next to correctness, the other main characteristic of a good model is its usefulness. Although it can be concluded from research of Wang et al (1996) and Wand (2000) that this characteristic is judged at least as important as that of correctness, existing frameworks such as that of Krogstie (1999), do not take this characteristic into consideration, at least not in the sense as defined below. In general, it can be said that the usefulness of an aspect model depends on the extent to which its construction is a contribution to the achievement of the purpose at hand and that the usefulness of a whole model depends on the usefulness of its constituting aspect models. Note that usefulness is measured on a continuous scale rather than the measurement of the correctness of a model, which results in a model being either correct or not correct.

Let me first give an example of two models that are constructed using the same technique, that are correct in any of the senses above and different with respect to usefulness. Consider a very general purpose modeling language such as a state transition diagram, providing no further guidelines on what kind of things the world consists of and what to model or not to model. If we apply this technique to model business processes, we can do this in many ways, depending on what the way of thinking of the individual modeler is with regard to what comprises a business process. The outcomes could vary from the modeling of physical transitions on a product, the moving of documents through the organization, to the modeling of the mental states and mental transitions of the people involved in a process. All three models are correct in the above senses, but still vary with regard to usefulness.

![Diagram A](image)

**Figure 4-11: Models covering a different Abstraction Level**

Or, consider the example in Figure 4-11. Let us presume that the technique used in this example models activities and their precedence and that activities are defined in a rather general way, such as 'something causing a state change in the business process'. Both models shown above are empirical, syntactical, semantic correct and valid. Both models are the result of applying the same modeling technique on the
same business process. However, the outcome of the modeling process differs. The cause of this difference is that a different abstraction level has been chosen to model this business process.

In Figure 4-11(A), a rather low level of detail has been chosen that includes communication technological aspects, e.g. the phone, and considers an order as a written document that has to be copied before it can be passed to the cook and delivery boy. In Figure 4-11(B), a higher level of detail has been chosen, abstracting from communication technology and not considering an order as a written document that has to be copied. The difference between the two models was probably caused due to the fact that the definition of 'activity' in the technique left too much leeway. The usefulness of a model is to a large extent determined by the generic concepts that a technique offers to model a business process for a certain purpose. Later on in section 4.5, I will discuss those concepts suitable for constructing useful models.

It leads to the following quality property on the individual model level:

- **Usefulness** – An aspect model is useful if and only if the propositions contribute to the achievement of the purpose at hand. A whole model is useful if and only if all of its constituting aspect models are useful;

As said, usefulness is purpose dependent: when a model is used to derive requirements from for the (re)design of a certain artifact, the model itself should not contain details of this artifact in order not to be biased with regard to a specific design solution. For example, when business process models are constructed for the purpose of requirements engineering for Information Systems, this principle important. In the sections to come, section 4.4 and 4.5, it will be discussed how a modeling technique can contribute to the construction of respectively correct and useful models.

### 4.4 Quality of Modeling Techniques: Assuring Correctness

In the previous section a foundation for an understanding of the quality of business process modeling techniques was built by specifying what the results of the application of such a technique (the models) must answer to. In this section I will proceed to build on this foundation by identifying and defining those characteristics of modeling techniques presumed to contribute to the correctness and usefulness of models.

Starting point is that the purpose of modeling techniques is to facilitate the efficient and effective construction of models. Therefore, the next step is to revisit the properties of a good model that were proposed before and to consider properties of modeling techniques that are presumed to influence these model properties. After
that, the properties that were identified will be systematically listed and provided of a clear definition.

In general, it is presumed that a modeling technique can contribute to correct and useful models in the following two ways:

- **Correctness** – The modeling technique should be arranged in such a way that it aids the construction of correct models, prevents the modeler from making errors of this kind. Guideline for the investigation of the properties of a good modeling technique is therefore that a good modeling technique reduces the chance of making incorrect models. The correctness properties of a model will be revisited and properties that are presumed to be of influence on making errors of this kind will be proposed;

- **Usefulness** – As concluded in the previous section, at the end, the usefulness of a model, in the sense that every proposition made in the model is relevant for achieving the purpose at hand is to a large extent determined by the generic concepts that a technique prescribes, i.e. the propositions that are allowed by the modeling language. Therefore, generic concepts suitable for modeling business processes will be proposed.

In this section I will address how a modeling technique can contribute to the construction of correct models. Before identifying and defining properties that contribute to this, let us recall the distinction that was drawn between aspect modeling technique and multi modeling technique made in section 2.3: The focus of an aspect modeling technique is the construction of a single aspect model whereas a multi modeling technique has a much broader focus that comprehends the construction of the whole model. Just as the whole model comprises all aspect models, a multi modeling technique comprises all aspect modeling techniques.

**Prerequisites for Understanding Quality**

Prerequisite for being able to understand and measure the quality of a technique is of course that its constituting components are well described. As concluded, the prescribed graphical notation and the syntax of the modeling language are prerequisite for determining a model's empirical, syntactic, semantic correctness and validity. Formal measurements on a technique are possible only if these prerequisites have been satisfied; otherwise, presumptions with regard to the respective constituents have to be made. The necessity of those descriptions is captured in three technique properties expressing that the technique should be **empirical-, syntactical-, semantic accurate**:

- **Empirical accuracy** - an aspect modeling technique is empirical accurate if all modeling constructs are provided with a unique diagrammatic notation; A multi
modeling technique is empirical accurate if and only if all its constituting aspect modeling techniques are empirical accurate;

- **Syntactical accuracy** - an aspect modeling technique is syntactical accurate if it contains a set of production rules specifying how elementary constructs can be combined to form complex modeling constructs and a set of rules that allows to decide whether a resulting aspect model is consistent and well-formed; A multi modeling technique is syntactical accurate if and only if all its constituting aspect modeling techniques are syntactical accurate;

- **Semantic accuracy** – an aspect modeling technique is semantic accurate if all the modeling constructs are clearly and unambiguously defined; A multi modeling technique is semantic accurate if and only if all its constituting aspect modeling techniques are semantic accurate.

The construction of a meta model capturing the concepts and their interrelationships aids the syntactic accuracy. One can think of semantic accuracy in two dimensions. One dimension describes the extent to which the concepts used have a meaning that is understood and shared by the stakeholders that have to interpret the model. Another dimension describes the extent to which the concepts used have a formal meaning, a meaning that can be mapped on a formal system such as a computer system. The ideal positioning of accuracy on these dimensions depends on the purpose for which models are used.

Below, the properties that make up the correctness and usefulness of a model mentioned in the previous section are revisited and properties of techniques that are presumed to be of influence on the correctness of the resulting models are identified and defined.

**Assuring Empirical Quality**

Starting point for empirical correctness is that all concepts and thoughts that can be expressed using the language are provided with a unique notation. This prerequisite is already enforced by the empirical accuracy property of a technique. Furthermore, this notation should be chosen in such a way that it complies with agreed upon notational intuitive rules such as those mentioned by e.g. Sindre (1990) and Rumbaugh (1996): distinguishable, consistent throughout all aspect modeling techniques, etc. Below notational intuitiveness is defined as a property that is presumed to influence the empirical correctness of a model. The precise rules that the technique has to comply with are given in chapter 6.

- **Notational intuitiveness** - an aspect modeling technique is notational intuitive if its notation is in accordance with agreed upon notational intuitiveness rules; a multi modeling technique is notational intuitive if all of its constituting aspect modeling techniques are notational intuitive;
The figure below shows an example of a contra-intuitive notation, when given that the diamonds represent activities, the boxes represent exclusive choices and the arrows point to the activity or choice that precedes (!) the other activity or choice. In this example, A is followed by C or by D, based on a decision made in B.

![Diagram showing contra-intuitive notation](image)

**Figure 4-12: Example of a contra-intuitive notation**

Rules for notational intuitiveness that can be thought of are:

- Generic concept as well as individual concept should be recognizable, e.g. a rectangle to represent the generic concept action, a number and text in it to identify an individual action;
- Consistent use of notation throughout aspect models, e.g. arrows are used to represent precedence throughout all aspect models;
- Consistent with past practice, e.g. a rectangle to represent the generic concept action, a diamond to represent the concept choice, an arrow to represent precedence.

**Assuring Syntactical Quality**

The syntactical quality of a model is determined by its syntactical correctness, intra- and inter model consistency and intra- and inter model well-formedness. With regard to the syntactical quality of models, it is presumed that the more restrictive the syntactic rules are defined, the higher the probability is that rules will be violated during the construction of models and thus the higher the probability of a model with low syntactic quality. Therefore, *syntactic freedom* is identified as a property that influences model quality.

- **Syntactic freedom** - an aspect modeling technique has high *syntactic freedom* if it does not put restrictions on the construction of complex constructs out of concepts. The opposite of syntactic freedom is syntactic strictness. It is *strict* if it restricts how concepts can be combined to form complex constructs. The *syntactic freedom* of a multi modeling technique is determined by the average syntactic freedom of its constituting aspect modeling techniques;
Precise measures for the degree of syntactic freedom will be given in chapter 6. Figure 4-13 can function as an example to explain the syntactic freedom of modeling techniques. Let us assume that the arrow is not used as homonym for both precedence and document input but is restricted to express precedence (what ever this might mean with regard the relationship between e.g. document and action). Now it can be concluded that the flow chart has a very high syntactic freedom. It contains the symbols representing the concepts action, document, environmental process and initial action, and furthermore, an arrow drawn between any of these for concepts is syntactically correct. It is obvious that a technique that is not very restrictive with respect to its syntax results in models that have a high probability of being syntactically correct.

Figure 4-13: Example of Syntactic Freedom

With regard to the inter model consistency and well-formedness between distinct aspect modeling techniques it is presumed that the level of coherence between these techniques influences the inter model consistency of the resulting aspect models of the respective modeling techniques. More specific, it is presumed that the more coherency there is between aspect modeling techniques, the higher the probability that a consistency rule is violated and thus the higher the probability of an inconsistent whole model. Therefore, coherence is identified as a property that is presumed to influence model quality.

- **Coherence** - two aspect modeling techniques are *coherent* if there are propositions thinkable that can be expressed in both aspect modeling techniques. A multi modeling technique is coherent when its constituting aspect modeling techniques form a coherent system. A multi modeling technique is per definition coherent;
The considerations above can give birth to the impression that an ideal modeling technique would be a modeling technique without any restricting rules, allowing the modeler to model any construction he or she wants. This observation is true, however, as we will see later on, such a technique would on the other hand not at all be useful for modeling business processes because it allows anything to be modeled and has no restrictive value at all. Therefore a balance must be found.

**Assuring Semantic Quality**

A technique should assure that all statements possible according to the syntax are statements that can be tested on their being true or not by comparing them to the phenomenon that is modeled. This can be assured by *semantic accuracy* as mentioned before.

**Assuring Validity**

A model is valid when the propositions contained in it are true, that is, in correspondence with physical or social reality. Evidence for propositions concerning physical reality can be obtained by perception and conception. As said in chapter 1, there are however philosophical problems with regard to the extent that validity can be checked: We are comparing a model with the phenomenon that is modeled, but how can we have access to a phenomenon without modeling it? Evidence for propositions concerning social reality can be obtained by communication and conceptualization, checking the consensus on agreements made. Again, absolute certainty is not achieved since a consensus is inter-subjective rather than objective.

In Krogstie's framework, discussed in chapter 3, the measurement of validity (semantic quality in Krogstie's framework) is reduced to the set-theoretic comparison of the statements expressed in the model with the statements that 'can be said about the domain'. This must be considered as shifting the problem of evaluating semantic quality to the problem of how to determine the set of statements that 'can be said about the domain' rather than simplifying or solving the actual problem.

In an attempt to bypass this problem, I will propose and explain the *determinism* property of a modeling technique as a property indicative for the validity of resulting model. Imagine two modeling techniques A and B and a representative population of subjects that constructs models using both A and B on a single case description. It turns out that when the subjects use technique A to model the case, all resulting models are equal, i.e. contain the same statements about reality. The application of technique B results in models that differ a lot from each other. In this case, technique A is said to be more *deterministic* than technique B. A technique is completely deterministic if there is, given the technique and the domain that is modeled, only one possible resulting model of that domain. This model is the only valid model of the domain.
• **Determinism** - An aspect modeling technique is deterministic when the repeated application of the technique on a domain results in the equivalent aspect models of that domain. A multi modeling technique is deterministic when all of its constituting aspect modeling techniques are deterministic;

Interesting with regard to the modeling of business processes is, that there are concepts, e.g. an invoice of which there exists both empirical evidence, an observable document that represents the invoice, as well as a consensus of what an invoice is, being in accordance with an agreed upon business model. Other concepts such as 'business objectives' and 'business goals' have no observable representatives and therefore we are restricted to checking the consensus on them. In particular concepts that are used in a metaphorical way such as 'information flows' suffer from this lack of means for validation, since no observer has ever witnessed information flowing through a room.

A clear definition of the concepts and knowledge used in the technique is assumed to positively influence the determinism of a technique and the semantic quality of the resulting models. Each concept used in the technique for business process modeling should be, as much as possible, provided with a definition that refers to established, agreed upon core concepts used in discrete dynamic systems modeling. Such a structure of concepts, or ontology will be proposed later on in this section.

**Complexity**

Finally, the complexity of a modeling technique is presumed to influence the correctness of the resulting models. In chapter 6, a number of metrics to measure the complexity of modeling techniques will be proposed.

• **Complexity** – The more concepts and knowledge the modeler needs to know in order to apply a multi modeling technique, the more complex a multi modeling technique is.

### 4.5 Quality of Modeling Techniques: Assuring Usefulness

I find it important to state that the framework that will be unfolded in the remainder of this section is a generic framework, suitable for evaluating the usefulness of business process modeling techniques. An important flaw that has to be avoided is that this framework in specified in such a way that it is biased towards the way of thinking underlying the modeling techniques that we are going to evaluate with it later on. Although it is tempting to define such a framework, I adhere to the idea that the main purpose of the evaluation framework is that its application allows to show the differences between modeling techniques with regard to their usefulness for modeling business processes.
Discrete Dynamic Systems

The basic definition of a system is it being a collection of interrelated elements. Therefore, it has composition (the elements) and structure (the interrelations between the elements). The nature of the elements and interrelations determines the system class.

A first distinction that can be made when modeling systems perspectives is a distinction between function and construction. The functional perspective models a system as a set of functions or processors that are linked to each other by means of inputs and outputs. The constructional perspective models a system as discrete dynamic system, systems that contain elements that change each others internal state as will be explained in further detail throughout this section.

Although classical modeling techniques such as Data Flow Diagrams model business processes as functions with inputs and outputs, i.e. take a functional perspective as starting point, the basic paradigm underlying the business process modeling techniques studied in this thesis, a vast majority of business process modeling techniques in the Netherlands (BWise, DEM, DEMO, Petri-net+, Protos, Testbed, et cetera, see Dommelen (1999)) and the vast majority of business process modeling techniques commonly used in the world (Event Driven Process Chains, Petri-nets, Activity Charts, UML Business Extensions, et cetera) is the presumption that it is possible and can be useful to conceive things in reality as business processes by using the discrete dynamic systems paradigm as the underlying theory. Therefore, this way of thinking will be adopted as a starting point for evaluating modeling techniques. Let me explain the notion of a discrete dynamic system.

In line with e.g. Bunge (1979), I define a discrete dynamic system as collection of interrelated elements that are able to change each other's internal state. These elements can themselves be (discrete dynamic) systems, in that case we speak of an aggregated system. The internal state of an element changes by means of discrete state transitions. The collection of all states that the elements of such a system can be in is called the state space of the system. A transition is the change of a system from one state to another state. State laws describe between which states transitions are allowed. The lawful state space is a subset of the state space, containing those states that are allowed, i.e. can be fulfilled by following the state laws of the system. The path of transitions between states that a system has followed is called the history of the system. In this thesis, the term 'process' is often used as a synonym for the term 'discrete dynamic system'.
There are numerous examples of discrete dynamic systems that can be thought of. A single traffic light for example. The state space of this system is a set containing three elements \{Red, Yellow, Green\}. The state law prescribes that Green→Yellow, Yellow→Red and Red→Green are law full transitions between these states. Next to artificial systems, also natural processes can be modeled as discrete dynamic systems, ranging from the changing of the four seasons to chemical reactions. Systems that follow a socially constructed state law such as a game of tennis can again be conceived as a discrete dynamic system. Some systems have clear begin and end states: a tennis game starts with 0-0 and ends with one of the players winning or one of the players giving up; others, such as natural processes don't.

Two or more systems can be combined together to form an aggregate system. Then, a vector containing all states of the subsystems forms the new current state of the aggregate system. E.g. the combination of four traffic lights on a crossing results in a system of traffic lights of which the state is described by a vector \(v \in \{\text{Red, Yellow, Green}\}\). An example of a state that the system could be in is the vector \(v = <\text{Red, Green, Red, Green}>\), or the vector \(v = <\text{Green, Green, Green, Green}>\). It is obvious that it would not be wise to include this last state in the lawful state space of the system.

As said, most techniques view the working of a business or organization as a discrete dynamic system as well. Not only the whole business or organization can be viewed as an aggregated discrete dynamic system, all its subsystems that can be thought of comprising this system, such as the production system, logistic system, financial system, personnel system can be treated as discrete dynamic systems.

**Actors and Systems**

Another important concept, next to the notion of a discrete dynamic system is the notion of an actor. This concept is briefly touched upon by Bunge's 'acts on' construct but needs, in my opinion, more elaboration. The question addressed dealt with is, what actually causes the system to change from one state to another. According to Bunge (1977) and later on adopted by the BWW ontology, an element
is said to 'act on' another element if it changes the history of the later element. Together these elements constitute a discrete dynamic system. Here I would like to introduce the concept of an actor as a thing or system that causes transitions in the system that it acts on.

It is worthwhile to distinguish deterministic and non-deterministic state laws. Non-determinism means that there are states in a system of which the state law prescribes more than one lawful subsequent state. On the other hand, determinism means that given the current state of the system, the state law prescribes at most one lawful subsequent state. Non-determinism implies that the actor that acts upon an element in the system has to make decisions with regard to how to affect its history. The traffic light mentioned above is an example of a deterministic state law. Given that the current state is green, we know that the next state will be yellow. There is no actor that can change the history of this system because there is no choice between Red and Yellow given that the current state is Green.

An actor that makes decisions that affect the history of a system needs to have information to motivate these decisions. An actor needs to know the current state or past states (history) of other systems in order to decide how to change the system that the actor acts upon. Traffic lights offer actors that act upon cars information on the current state of the traffic light system. They need to have this information to influence the history of the car that they drive.

Consider for example a discrete dynamic system that consists of three elements, viz. the assembly of products out of components as a subsystem, inventory control as a subsystem, and an actor. Typical transitions in an assembly process are that a certain component is added to the product that is assembled. The current state of the product reflects the extent to which it is assembled. Typical transitions in the inventory control system are that components enter or leave the stock. The current state of the inventory control system reflects how many components there are on stock. The actor that assembles (acts upon the assembly system) needs to know the current state of the assembly process as well as the current state of the inventory control system, in order to make the right decisions.
What we have discussed so far is depicted in Figure 4-15. Actors cause the transitions in discrete dynamic system to occur. In order to do so, they might need to know the state of other subsystems. If the nature of these systems is physical, e.g. the process of building a house, it is often possible to acquire this knowledge just by perception of the system. In other cases, when the nature of the system is non-physical, e.g. socially constructed like the game of tennis we discussed before, to get to know the state of the system is a more complicated process of acquiring information by means of communication.

The ability to cover the discrete dynamic systems paradigm that was introduced above results in the first property to assure usefulness:

- **System Expressiveness** - An multi modeling technique is *system expressive* if the concepts and knowledge primitives contained in it form a language that is expressive enough to allow modeling according to the discrete event systems paradigm as discussed above;

### Perspectives and System Categories

In chapter 3 it was concluded that there is not much consensus on what a system perspective is and what perspectives should be covered by a modeling technique. In this chapter an attempt is made to address this problem, using the discrete dynamic systems paradigm as a starting point. To make order out of all perspectives mentioned in chapter 3, it is important to introduce two notions. The first notion is the notion of *basic perspectives* inherent to the discrete systems paradigm: the *structural* and *behavioral* perspective.
On the one hand, the *structural* perspective describes the state space of the system. For this end, for example Entity Relationship or Object Role models can be used. The *behavioral* perspective describes the lawful transitions in this state space. For this end, process models such as Petri-nets or Event Driven Process Chains can be used. The last mentioned technique does not purely model this perspective, it also models the actors causing transitions.

The second notion that must be introduced is the notion of the *system category* to which the transitions, state space, et cetera belong. E.g. the definition of the concept 'transition' in the discrete dynamic systems paradigm is broad enough to encompass actions such as 'producing a chair', 'make a calculation', 'sending a letter', 'entering product data into a database' and so on. Business process models that included these actions could be empirical, syntactical and semantically correct and valid, but not always useful as a model to for instance derive requirements for an information system from. Therefore, the general concepts such as transitions, states, actors and so on, need to be specialized to be suitable for modeling business processes for the purpose at hand.

![Diagram of Actors and System's State Space]

Figure 4-16: System Categories

In chapter 2, a basic definition of a business process was already given. A business process is often defined as a structure of organizational or inter-organizational activities that are necessary to accomplish a product or service of value to a customer (c.f. Davenport, 1993, Hammer *et al*, 1993), coordinated by means of communication (Taylor, 1993).
As a first possible classification, one could distinguish core (primary), supportive (secondary) and managerial processes. In other words, a business system is an aggregate system that consists of primary, secondary and managerial subsystems.

The desired end state of primary processes is to deliver a product or service to a customer. Transitions that take place in the primary subsystem are therefore directly related to the product or service that is created. Although there are several ideas on how to identify these transitions, a generally agreed upon idea is that each transition should add value to the product or service (Porter, 1985). Examples of primary processes are logistics, production, sales and service. With regard to the logistics system of an organization we can speak about the moving of a product from one place to another as a logistic transition and the current position of all products as the logistic state. We can see the manufacturing of a single product as a transition in the production system and the current stock as the state of the production system, and so on.

Supportive or secondary processes are processes that are not directly related to the products or services delivered to a customer, but indirectly take care that the primary process can take place. Examples of supportive systems are personnel and organizational processes, financial processes and purchasing processes. With regard to the financial system of an organization we can speak about payments as financial transitions and the financial state of an organization, the financial position. With regard to the personnel system we can speak about the hiring of a new employees as personal transitions and about the current set of employees as the personnel state of the organization. We can imagine that an actor that wants to carry out a transition in the personnel system first inspects the financial state of the organization, and so on.

The third type of processes is the managerial process. They create the conditions for the primary and secondary processes to take place by long term decisions, taking into account the company's goals, objectives and external influences. With regard to the management system we can speak about a single decision taken as a managerial transition and the set of all decisions taken as the current state of the managerial system.

It is important to remember that these discrete dynamic systems do not deploy activity by themselves. Actors cause the transitions to occur and actors need information of the state of other systems to make the correct decisions with regard to the changing of the history of the system that they act upon. Between the systems there can be interrelationships that the actor has to take into account when acting upon these systems. When this idea is applied to the world of business systems we see that the actors form the glue between all subsystems of an organization. This idea is depicted in Figure 4-17.
The following system categories are distinguished:

- **Business System** – lawful transitions and state space of the core, supportive and management processes. Subsystems of this system are e.g. the production system, financial system, personnel system and purchasing system. An example of a transition in such a system is that a certain item is purchased and an example state is the total of purchased items.

- **Information System** – lawful transitions and state space of the information system supporting business processes and coordination between processes. An example of a transition is creation of information or change of existing information. The state represents all information known at a certain point in time.

- **Documental System** – lawful transitions and state space of the system for representing information. An example of a transition is the entering of order information into a computer system or the writing down of this information on a piece of paper. The state is represented by the state of e.g. all database systems in an automated organization.

- **Communication System** – lawful transitions and state space of the system by which work (transitions in each of the above mentioned systems) is coordinated. An example of a transition is the external request of a customer to deliver a product or an internal request or confirmation. These transitions are part of conversations. The state of this system is the aggregation of the states of all conversations in an organization.
An evaluation of the business suitability of a modeling technique should investigate which *perspectives* of which *system categories* are modeled and how these perspectives interrelate. E.g. what system category is modeled when a process model contains activities such as 'enter order into system' and what transitions does such an activity cause in the information system, a transition in what kind of system is a trigger caused by an incoming letter or invoice? Business suitability is defined as follows:

- **Business Suitability** – A multi modeling technique is Business Suitable when the concepts and knowledge primitives are grouped into aspect modeling techniques able to model the relevant perspectives of relevant system categories.

**Purpose suitability**

The various purposes for which business process modeling is used influences the importance of above mentioned quality properties. To illustrate this, a few examples will be mentioned. When business process modeling is used in order to derive requirements for systems that support these processes it is obvious to make a clear cut between the modeling of business systems and the modeling of the systems that support these systems, since for each business transition it must be determined how it will be supported.

When business processes are modeled as process specification for workflow management systems, transitions in all possible system categories that are relevant for the execution of processes should be modeled. Furthermore, the meaning of the concepts offered by the technique should be such that it can be mapped to the semantics of a formal computer system. When business process modeling is used for e.g. ISO certification we might want to include concepts such as decision and responsibility that are understood by the stakeholders. The process of information systems design can be viewed as a model transformation process from informal models that are understandable by stakeholders to more formal models that are understandable by information systems engineers. The final goal might be to end up with a description of the structural perspective of the information system together with constraints on transitions that can take place in the information system. When we model for the purpose of e.g. Activity Based Costing, especially financial transitions must be modeled together with their relationship to other business transitions.

In general it can be concluded that the purpose for which business process modeling is used makes demands on the previously mentioned quality properties. These are e.g. demands on *business suitability* (stress on particular perspectives and system categories), *semantic accuracy* (should the concepts offered have a semantics that can be interpreted by a computer or easily understood by stakeholders) and *coherence* (when one model is derived out of another model during a modeling process).
4.6 Conclusions

This chapter is the first of three chapters in which the integrated approach for evaluating the quality of modeling techniques is proposed. The underlying presumption of this framework is that quality can be subdivided into a number of quality properties that in their turn can be operationalized or quantified by means of metrics. In this chapter I have proposed the quality properties of a modeling technique.

As a first step towards the identification of important technique properties, I have defined the quality of models. The quality properties of models were identified based on existing literature and flaws in existing approaches. By doing so, two main properties were identified: a model should be correct and it should be useful. The correctness property was further subdivided into internal correctness and external correctness. Internal correctness concerns correctness regardless of the models correspondence to the phenomenon that is modeled, such as empiric, syntactic and semantic quality. External correctness concerns correctness with regard to the phenomenon that is modeled, the validity of the model. As said, next to being correct, a model should also be useful for the purpose at hand. This usefulness is to a large extent determined by the generic concepts that a modeling technique offers to the modeler.

With regard to the correctness of a model it can be concluded that the definitions of syntactic correctness are to a large extent in line with the work of Krogstie (1999) and other existing work discussed in chapter 3. A contribution is a more precise definition of empirical quality and the addition of the consistency and well-formedness properties. The division into intra- and inter model consistency is an original contribution aimed at tackling both the lack of these properties in existing frameworks and the inability of existing measurement approaches to take into account relationships between aspect models (c.f. work of Rossi et al, 1996, section 3.5).

As opposed to the existing frameworks, semantic quality is treated fundamentally different, more in line with the definition in chapter 2. In Krogstie's framework for example, semantic quality is something that can be determined by comparing a set containing statements that can be said about the domain and a set containing the statements made in the model just as syntactic quality is determined by comparing the set of statements in the model to the set of all statements possible according to the syntax of the language. Problem of this simplification is that it is impossible to determine the set of statements that can be said about the domain.

The next step concerned a definition of the quality of techniques. Properties have been identified presuming to influence the quality of the resulting models that was defined earlier on. Figure 4-18 lists those quality properties.
The first three properties are presumed to influence the internal correctness of models, based on the principle that the least restrictive the technique is, the fewer errors will be made in the resulting models. These considerations can give birth to the impression that an ideal modeling technique would be a modeling technique without any restricting rules, allowing the modeler to model any construction he or she wants. In one way, this observation is correct. However, as we will see later on, certain constraints are unavoidable and a multi modeling technique is per definition coherent, causing the consistency problems. A balance between not being restrictive and being too restrictive must be found by e.g. identifying coherency relationships and other restrictions within techniques that are not necessary and have no function.

Determinism is a measure for determining the validity of resulting models. The less determinism, the larger the variance in outcomes and the less sure we can be that the outcome is a valid representation of the phenomenon that is modeled. The complexity of a technique is presumed to influence the overall number of errors made in models. The last two properties, expressiveness and suitability are presumed to influence the usability of models. With expressiveness, the capability of the modeling technique to model business processes as discrete dynamic systems is meant. With suitability, the capability of the modeling technique to model those dynamic systems required for the purpose at hand.

The operationalization of the properties by means of measurements is the subject of chapter 6. As a step towards this objective, I will now proceed by proposing the language for capturing modeling techniques. Based on this language, the measurement scheme will be proposed.
5

A PROPOSAL FOR CAPTURING BUSINESS PROCESS MODELING TECHNIQUES

5.1 Introduction

In the previous chapter, a proposition for a framework for understanding the quality of business process modeling techniques has been made. In this chapter, a first step towards the operationalization of this framework is made by proposing and formalizing the means to describe the constituting elements of a modeling technique. The final step towards a framework for understanding and evaluating modeling techniques will be discussed in chapter 6, where a framework of measurements is proposed, based on the formal means for description that are unfolded in this chapter.

In this chapter, the C-Me description language (Capturing Models for Evaluation) will be introduced, thereby answering research question five. The C-Me description language is derived from the language that was first published in Hommes et al (2001). It is part of the fact modeling family of approaches discussed in chapter 3, specifically tailored to model the modeling concepts of multi modeling techniques.

5.2 Describing Modeling Techniques

As said in section 2.2, a conceptual model is a coherent set of individual concepts and knowledge that is constructed to better understand the phenomenon that is modeled, shaped by the way the generic concepts and knowledge a theory offers to conceive reality. The construction of a system of conceptual things (concepts) and knowledge that corresponds to reality was called the process of 'conceptualization'.

Figure 5-1 depicts a possible reality, the conceptual world and its relationship. It depicts the conceptual world as a system of which the elements are conceptual things (concepts) that may correspond to things that may exist in reality (objects).
The idea of real things, conceptual things and their relationships can be made clear by a number of simple examples. The criterion for deciding whether a thing is a real thing is whether its existence can be perceived. To the class of real things belong both permanent things corresponding to concepts such as 'Car', 'Person', 'Tree', 'Banknote' and 'River' as well as transient things corresponding to concepts such as 'Rising of the sun', 'Sound of a passing car' or 'Somebody picking up a banknote'.

The most intuitive class of concepts is the class of things that are the result of a one-to-one perception and conception of a real thing based on its physical features. This class will be called the class of physical concepts. Characteristic of this class is that there is a one-to-one relationship between real thing and conceptual thing, based on the physical features of the real thing. Examples of this class of concepts are 'Tree' and 'Mountain', which are physical concepts that correspond one-to-one to permanent real things. 'An apple falling from a tree' is an example of physical concept corresponding to a transient real thing.

Human beings use real things for various purposes. In other words, they have the capability to assign functions to real things. We can use water as a drink and trees as fuel. But more often we create real things to perform a function for us, such as paper and ink to write on, hammers to hit nails with, knives to cut with and matches to make fire. Functional concepts are the result of a conceptualization of a real thing based on the function that humans assign to it (c.f. Searle, 1995). Just as a physical concept, a functional concept is also the result of a one-to-one conceptualization of a real thing, however in the case of a functional concept, based on its function rather than its physical features. As being said, examples of functional concepts are 'Drink', 'Fuel' and 'Screwdriver'.

Note that the real things that correspond to functional concepts can always be treated as physical concepts as well. We can treat a hammer as a physical concept when we want to predict how fast it falls from a shelf, yet as a functional concept if we need something to hit nails with. We can treat a car as a physical thing if we want to repair it, as a functional concept 'Vehicle' if we want to drive it.
A third class of concepts is the *representation concept*. A representational concept is the conceptualization of a real thing that is utilized to *represent* other concepts or knowledge with which it does not correspond by itself. This representation is by human convention. We use the term 'document' to refer to the real thing that corresponds to a representational concept.

Representational concepts are both physical and functional. They are physical, since one must be able to observe them in order to interpret the concepts or knowledge that the thing that it represents. When conceiving a document as a physical concept, one speaks of the *medium* of the document. Media can be either permanent, such as a piece of paper containing written text, or transient, such as the sound of someone pronouncing a text. A document can also be conceived as a functional concept since its function is to communicate concepts and knowledge. Some form of language is used to communicate the concepts and knowledge represented.

Examples of representational concepts are a ‘Stop sign’, which represents the knowledge that one has to stop entering a crossing, a 'Road map', which represents knowledge about the infrastructure of a country or city. But also, and that is of special importance for understanding the means used to model all kinds of phenomena in reality, such as business processes, are the schemas and diagrams that represent knowledge of the phenomenon that is modeled.

Finally, a fourth class of concepts that is distinguished is the class of *abstract concepts*. Things that belong to this class are things that are used in social traffic as agreed upon concepts that have no counterpart in the world of senses. They exist because of human convention (c.f. Searle 1995). An example of an abstract concepts is e.g 'Marriage'. In social traffic, we can use the representational concept ‘marriage certificate’ to communicate the marriage. It is itself not a marriage but something that we use to substitute for the marriage. Conceptualized as a physical concept, the certificate is just a piece of paper with ink on it. Furthermore, its function is to communicate the institutional concept 'Marriage', hence it can be conceptualized as a functional concept. It represents knowledge about the marriage that has taken place, and therefore it can be conceptualized as representational concept.

**Concepts and Knowledge**

In chapter 2, I have discussed how concepts form complex thoughts, that these thoughts have a truth condition and that validated true thoughts are factual knowledge. Conceptual models contain factual knowledge about the phenomenon that they model and are used in twofold in this thesis: Business Process Models that contain factual knowledge about business processes and models that are used to understand modeling techniques that contain factual knowledge about the modeling technique that is to be understood. Below, I will elaborate on the representation of
generic and individual concepts and how knowledge regarding these concepts is treated in the C-Me language.

Let us recall the distinction between individual and generic concepts made in chapter 2 by means of a well-known example of deductive reasoning which goes as follows: '1) All men are mortal, 2) Socrates is a man, therefore 3) Socrates is mortal'.

The three statements are expressions of knowledge. The first statement, 'All men are mortal', is the expression of generic knowledge, since it the concept 'Man' to which the knowledge refers is a generic concept. The second and third statement, respectively 'Socrates is a man' and 'Socrates is mortal' are expressions of individual knowledge since they refer to the individual concept 'Socrates'.

![Diagram](image)

**Figure 5-2: The Individuality and Generality of Concepts**

An individual concept may correspond directly to a particular object in reality. Which is represented in the figure above by a solid arrow between object in reality and individual concept in the conceptual world. Generic concepts are an abstraction over these individual concepts. They group individual concepts into classes of concepts, based on an equivalence relationship. As depicted in Figure 5-2, their relationship to individual concepts is 1:n. Knowledge about an individual concept being an instance of a generic concept such as 'Socrates is a man' is represented by a solid arrow pointing from an individual concept to the generic concept to which it belongs.

It is important to note that only the individual concepts may have a direct correspondence to reality. Since abstraction is a rational process, generic concepts do not directly correspond to reality and their existence can therefore not be validated. At most, their existence can be communicated.

Often generic concepts are grouped and placed in an order by what is often called a classification scheme. This is denoted by the sub concept relationship in Figure 5-3, a solid arrow pointing from one generic concept to another generic concept. It
expresses the generic knowledge that one generic concept is a specialization of another generic concept, e.g. the knowledge 'Philosophers are men'. Apart from that one can say that the set of individual concepts that are instances of 'Philosopher' is a subset of the individual concepts that are instances of 'Men'.

Figure 5-3: Sub concept and type-instance relationships

Apart from concepts, conceptual models contain factual knowledge concerning concepts. In line with the definition of knowledge in chapter 2, factual knowledge is viewed as a set of true thoughts, i.e. thoughts in accordance with reality, each thought referring to one or more generic or individual concepts. A single true thought will be called a knowledge primitive. Knowledge primitives that refer to individual concepts belong to individual knowledge, whereas knowledge primitives that refer to generic concepts belong to generic knowledge.

Again it is important to note that only the individual knowledge primitives may have a direct correspondence to reality. Since generic knowledge primitives refer to generic concepts, which are abstractions from the individual ones, they do not directly correspond to the world of senses. The only way to validate generic knowledge primitives is through the validation of particular individual knowledge primitives.
Figure 5-4: Individual and Generic Knowledge Primitives

Figure 5-4 shows the representation of knowledge primitives in the proposed framework. The generic knowledge primitive 'Owns' expresses the generic knowledge that cars can be owned by persons. The individual knowledge primitive 'Jill Owns 123456' expresses the individual knowledge that a person called Jill owns the car '123456'. As shown, knowledge itself is part of the conceptual world and can never be part of reality itself.

In order to represent conceptual constructs that are both generic concepts as well as knowledge as is the case with the concept 'marriage', the objectification relationship is introduced. The notation used for this objectification relationship is a dotted line between knowledge primitive and concept. This relationship is illustrated in Figure 5-5.

Figure 5-5: Example of Objectification

Language

As said in chapter 2, a problem inherent to utterances in colloquial language is that their syntax and semantics are often ambiguous due to context dependence. In contrast to colloquial languages, formal languages are subject to precise rules. The table below shows the correspondence between the C-Me language and first order predicate logic.
### Table 5-1: Correspondence Predicate Logic and Graphical Language

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Example in First Order Predicate Logic</th>
<th>Example in proposed Graphical Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Concept</td>
<td>Car(x)</td>
<td>![Generic Concept Diagram]</td>
</tr>
<tr>
<td>Individual Concept</td>
<td>Car(123456)</td>
<td>![Individual Concept Diagram]</td>
</tr>
<tr>
<td>Generic Knowledge Primitive</td>
<td>Owns(x,y)</td>
<td>![Generic Knowledge Primitive Diagram]</td>
</tr>
<tr>
<td>Individual Knowledge Primitive</td>
<td>Owns(123456,Jill)</td>
<td>![Individual Knowledge Primitive Diagram]</td>
</tr>
<tr>
<td>Knowledge Reference</td>
<td>∀x,y [Owns(x,y) → Car(x)]</td>
<td>![Knowledge Reference Diagram]</td>
</tr>
<tr>
<td>Sub type relationship</td>
<td>∀x [Philosopher(x) → Person(x)]</td>
<td>![Sub type relationship Diagram]</td>
</tr>
<tr>
<td>Objectification</td>
<td>no representation in first order predicate logic</td>
<td>![Objectification Diagram]</td>
</tr>
<tr>
<td>Cardinality</td>
<td>∀x,y,z [Owns(x,y) ∧ Owns(x,z) → y = z]</td>
<td>no representation</td>
</tr>
<tr>
<td>Transitivity</td>
<td>∀x,y,z [LargerThan(x,y) ∧ LargerThan(y,z) → LargerThan(x,z)]</td>
<td>no representation</td>
</tr>
<tr>
<td>Symmetry</td>
<td>∀x,y [MarriedTo(x,y) → MarriedTo(y,x)]</td>
<td>no representation</td>
</tr>
</tbody>
</table>

Apart from the concept of objectification, the syntax of the graphical language is a subset of predicate logic. It is less expressive and there are statements that are allowed in predicate logic and not in the graphical language, such as those labeled with 'no representation' in the table above. In those cases it can be convenient to use a combination of graphical language and predicate logic as is depicted below. The phrase in predicate logic is an example of a cardinality expression. It expresses that cars cannot be owned by more than one person.
**Representation of Concepts and Knowledge**

With the introduction of a document to represent conceptual knowledge, two types of relationships between Reality and Conceptual World can be distinguished. This treatment of the relationship between real- and conceptual world is in the tradition of e.g. Ogden et al (1923) and Morris (1955). Figure 5-7 shows the two types of relationships that can exist between reality and conceptual model:

- The *correspondence relationship* between objects in reality and individual concepts in the conceptual model, denoted by the solid arrow in Figure 5-7. This relationship is a conceptualization of a real thing that is known to exist in reality. Only individual concepts qualify for this relationship, since generic concepts are the result of an abstraction and have no direct relationship with reality except via their instances. Furthermore, only real things that can be recognized by means of empirical observations qualify as objects in this relationship. Institutional concepts that are not physical such as marriages and conceptual models, have no corresponding counterpart in empirical reality. An example is the concept 'The car with license plate 123456' in the mind of an individual observer, corresponding to the real car in reality;

- The *representation relationship* between objects in reality and concepts and knowledge in the conceptual model, denoted by the dotted arrow in Figure 5-7. It is a relationship between a special kind of real thing, called document and the concepts and knowledge that the document is assigned to represent. Because representation is by human convention, this relationship in not restricted to include individual concepts but all kinds of other concepts and knowledge (individual, generic, physical, institutional, etc.) as well. An example is the generic knowledge that 'one should not smoke' that is represented by a physical non-smoking sign.

**Figure 5-6: The Combination of Predicate Logic and Graphical Language**
Figure 5-7: Two types of Relationships between Reality and Model

Note that one single concept can have both of the above relationships with two different objects in reality as for instance the concept 'Plato' that corresponds to the real thing 'Plato' (the person) and could be represented by a written word 'Plato'. This illustrated in Figure 5-8.

Figure 5-8: Concept corresponding to and represented by a real thing

On the other hand, one single object, in particular a document, can have both of the above relationships with two different concepts as for instance the real thing 'marriage certificate' that represents the concept 'marriage' and corresponds to the concept 'marriage certificate'. This second possibility is illustrated below.
Describing Conceptual Modeling Techniques

As said in chapter 2, modeling techniques offer generic concepts and knowledge that can be utilized to model phenomena. An important aspect of modeling techniques is the modeling language that serves as a vehicle with which individual concepts and knowledge can be represented. Figure 5-10 shows an example of a business process modeling technique and how the generic concepts and knowledge used can be described using the C-Me language. First of all, the modeling technique uses boxes to describe the tasks that have to be carried out (e.g. to ‘write down order’ or to ‘deliver order’). In the C-Me model, the concept ‘task’ is used to denote this. Furthermore we see that the technique uses diamonds to denote choices to be made. Via transformation (b) the concept ‘choice’ appears in the C-Me model. ‘Task’ and ‘choice’ are modeled as specializations of ‘activity’. Via transformation (c) the positions that carry out activities are modeled. The modeling technique allows to model precedence between activities, explicitly by means of arrows and implicitly by the vertical ordering of activities in the diagram. Via transformation (d) this precedence is modeled as a knowledge primitive referring to two activities. Finally, tasks can be assigned to positions. For example, the cook is responsible for ‘prepare order’ and ‘deliver?’. Via transformation (e) this responsibility between position and task is captured in the C-Me model.
In order to model different perspectives and system categories, different aspect modeling techniques are used, resulting in different aspect models. For instance, numerous aspects of a single car can be modeled requiring different generic modeling knowledge in advance. Conceptual models might focus on e.g. the electrical system, the ignition system, the fuel system, the transmission system or the breaking system. Whereas the electrical system of a car can be conceived as existing out of instances of the generic concepts dynamo, bobbin, battery, wire, power, amperage, charge, etc., these concepts are not of any use when a car's fuel system is modeled.

Two aspect modeling techniques are *conceptual coherent* if and only if there are generic concepts and knowledge primitives present in both techniques that allow the expression of the same individual concepts and knowledge primitives. Figure
5-11 shows an example. It is important to note that to serve as an example, the classical DFD and ERD techniques have been combined and further formalized than originally described by e.g. Yourdon (1989). The example describes a minimal coherency relationship that exists of only one generic concept, viz. the Entity concept. Next to these minimal coherency relationships there are coherency relationship thinkable of more complex structures of concepts and knowledge primitives (so-called modeling constructs). In this case, structures of modeling concepts and primitives in two aspect modeling techniques allow the expression of the same knowledge.

![Diagram](image)

**Figure 5-11: Example application on Data Flow Modeling**

The following conceptual coherency relationships are distinguished:

- **Elementary Coherency Relationship** - A single generic concept is shared by two aspect modeling techniques, as is the case with the Entity concept of the example above. Three types of elementary coherency relationships can be distinguished. In line with the example above, it can be the case that each entity described in a DFD should also appear in the ERD. In that case the extension of the generic concept Entity, viz. the set of individual concepts, in the ERD is a subset of the extension of the same concept in the DFD, denoted as \( \text{Ext(ERD}_{\text{Entity}}) \subseteq \text{Ext(DFD}_{\text{Entity}}) \) or in general, \( \text{Ext(A)} \subseteq \text{Ext(B)} \). Furthermore it can be the case that all entities should be mentioned as well in the DFD as in the ERD. In this case, \( \text{Ext(A)} = \text{Ext(B)} \). The third possibility is that there is an overlap in entities mentioned in both diagrams, but not necessarily so that each entity should appear in both of the diagrams. In that case, \( \text{Ext(A)} \cap \text{Ext(B)} \neq \emptyset \) such that \( \text{Ext(A)} \cap \text{Ext(B)} \subseteq \text{Ext(A)} \) and \( \text{Ext(A)} \cap \text{Ext(B)} \subseteq \text{Ext(B)} \). The three types of coherency relationships will respectively be called *subset relationship*, *equality relationship* and *intersection relationship*.
Complex Coherency Relationship - Two aspect modeling techniques contain complex structures of generic concepts and knowledge primitives that allow the expression of the same knowledge. In that case, the knowledge expressed in both diagrams is semantic equivalent.

5.3 Formalization of the Description of Modeling Techniques

Core Definition of a Multi Modeling Technique
This section deals with the formalization of the C-Me language for capturing modeling techniques. The formalization is limited to that part of the language that is necessary for capturing the generic modeling knowledge used in modeling techniques.

A multi modeling technique is defined as an 8-tuple, as follows:

Definition 5-1: A multi modeling technique is a 8-tuple $M = < S, C, K, r, o, G, A, O >$, where:

Definition 5-2: $S$ is a set of aspect modeling techniques.

Definition 5-3: $C$ is a set of generic modeling concepts.

Definition 5-4: $K$ is a set of generic knowledge primitives.

In this tuple, $S$ is the set of aspect models. In the example that was depicted in Figure 5-11, this would imply that $S=\{\text{DataFlowModel, EntityRelationshipModel} \}$. $C$ is the set of all the generic modeling concepts in the language, regardless of the aspect model in which they appear. In the example above, this would imply that $C = \{\text{Process, Flow, Entity, Relationship} \}$. $K$ is the set of knowledge primitives, e.g. $K = \{\text{IsOutput, IsInput, SubProcess, FlowEntity, Role} \}$ in line with the example above.
Definition 5-5: \( r : K \rightarrow \wp(C \times \mathbb{N}^+) \) is a total function called the referent mapping function. \( \mathbb{N}^+ \) will be used as a notation for \( \mathbb{N} \setminus \{0\} \) in the remainder of this thesis. The referent mapping function maps the generic knowledge primitives to a set of Cartesian products specifying the concepts to which the knowledge primitive refers together with a role number to distinguish concepts that occur more than once in a knowledge primitive. E.g. the element \(<\text{Marriage}, \{<\text{Man},1>, <\text{Woman},2>\}>\) of \( r \) indicates that the generic knowledge primitive Marriage refers once to the generic concept Man and once to the generic concept Woman. E.g., \( r(\text{Marriage}) = \{<\text{Man},1>, <\text{Woman},2>\} \). The function is surjective since distinct knowledge primitives can be mapped to the same value. E.g., \( r(\text{Marriage}) = r(\text{Engagement}) = \{<\text{Man},1>, <\text{Woman},2>\} \). Based on the \( r \) function, the auxiliary function \( \text{card}(k \in K) = |r(k)| \) measures the cardinality or number of roles in a knowledge primitive \( k \).

The auxiliary function \( \text{plays_role} : K \times \mathbb{N}^+ \rightarrow C \) returns the concept that plays a certain role \( n \in \mathbb{N} \) in a certain knowledge primitive \( k \in K \). \( \text{plays_role} : K \times \mathbb{N} \rightarrow C \) is defined as follows: \( \text{plays_role}(k,n) = c \in C \mid \exists s \in \wp(C \times \mathbb{N}^+) : <k, s> \in R \wedge <c, n> \in s. \)

For the example above, this would imply that \( R = \{<\text{isOutput}, \{<\text{Flow}, 1>, <\text{Process}, 2>\}, <\text{isInput}, \{<\text{Flow}, 1>, <\text{Process}, 2>\}, <\text{SubProcess}, \{<\text{Process}, 1>, <\text{Process}, 2>\}, <\text{FlowEntity}, \{<\text{Flow}, 1>, <\text{Entity}, 2>\}, <\text{RoleIn}, \{<\text{Entity}, 1>, <\text{Relationship}, 2>\}\} \}. Example illustrating \( r, \text{card} \) and \( \text{plays_role} \) are \( r(\text{FlowEntity}) = \{<\text{Flow}, 1>, <\text{Entity}, 2>\}, \text{card}(\text{FlowEntity}) = 2 \) and \( \text{plays_role}(\text{FlowEntity}, 2) = \text{Entity} \).

Definition 5-6: \( o : K \rightarrow C \) is a partial function called the objectification function. It maps generic knowledge primitives to generic concepts that are objectifications of these knowledge primitives. The function is partial since not all \( k \in K \) need to be objectified by a generic concept. The function is bijective, therefore it has an inverse function \( o^{-1} \).

Definition 5-7: \( G \subseteq C \times C \) is called the generalization relation. For all elements \(<a, b>\) of \( G \), the concept \( a \) is the super type of \( b \) and the other way round, \( b \) is the sub type of \( a \). In the remainder of this thesis, generalizations are always total and disjunct. Based on the generalization relation we can define the auxiliary set \( C_{\text{Base}} = \{c \in C \mid \exists d \in C : <c,d> \in G\} \) which contains all base concepts, i.e. concepts that do not play a super type role. Furthermore, the auxiliary function \( \text{is_subtype_of}_C : C \times C \rightarrow \mathbb{B} \) is defined as follows: \( \text{is_subtype_of}_C(c_1,c_2) = <c_2,c_1> \in G \). This function returns \( \text{true} \in \mathbb{B} \) when \( c_1 \) is a subtype of \( c_2 \) and \( \text{false} \in \mathbb{B} \) otherwise. The function is transitive, i.e., \( \forall a, b, c \in C : \text{is_subtype_of}(a, b) \wedge \text{is_subtype_of}(b, c) \rightarrow \text{is_subtype_of}(a, c) \).

Based on the \( \text{is_subtype_of}_C \) function and the \( \text{plays_role} \) function defined earlier, the function \( \text{all_subroles} \) returns the set of all base concepts \( c \in C_{\text{Base}} \) that are sub concepts of the concept that plays a certain role \( n \in \mathbb{N}^+ \) in a certain knowledge...
primitive \( k \in K \). The function \( all\_subroles \ N^* \to \wp(C_{\text{Base}}) \) is defined as
\[
all\_subroles(k,n) = \{ c \in C_{\text{Base}} \mid is\_subtype\_of(c, \text{plays\_role}(k,n)) \}.
\]

**Definition 5-8:** \( A \subseteq (C \cup K) \times S \) is called the *concept aspect* relation. It defines in which aspect model the generic concepts and knowledge primitives appear. Distinct concepts and knowledge primitives can be mapped to the same aspect model, e.g. the generic concepts 'state' and 'transition' appear both in the aspect model 'state-transition-model'. On the other hand, there are concepts that appear in several aspect models, such as the UML concept 'class' that appears both in the aspect model 'class-model' as well as in the aspect model 'sequence-model' (OMG, 2001).

The last element of the C-Me tuple is the set \( O \) that may contain coherency relationships between modeling constructs in different aspect models. In order to define this set, some auxiliary sets and functions will be introduced first.

**Definition 5-9:** \( cofa : S \to \wp(C \cup K) \) is an auxiliary function. It is a total function that maps aspect models to the set of concepts and knowledge primitives that appear in the aspect model. The definition of \( cofa(s) = \{ c \in (C \cup K) \mid <c, s> \in A \} \).

The next five auxiliary sets (Definition 5-10 - Definition 5-14) contain pairs of concepts or knowledge primitives that are somehow connected, either by a sub type relationship, referent mapping or objectification.

**Definition 5-10:** \( CE_{\text{Sub}} = \{ <a, b> \in G \mid a \in C, b \in G \} \) is an auxiliary set containing all pairs of concepts that are connected by means of a sub type relationship.

**Definition 5-11:** \( CE_{\text{Ref}} = \{ <a, b> \in K \mid \exists x \in N^+ : <a, x> \in r(b) \} \) is an auxiliary set containing all pairs of concepts and knowledge primitives that are connected by means of a referent mapping.

**Definition 5-12:** \( CE_{\text{Ref},1} = \{ <a, b> \in C \mid \exists x \in N^+ : <b, x> \in r(a) \} \) is an auxiliary set containing all pairs of knowledge primitives and concepts that are connected by means of a referent mapping.

**Definition 5-13:** \( CE_{\text{Obj}} = \{ <a, b> \in K \mid \alpha(a) = b \} \) is an auxiliary set containing all pairs of knowledge primitives and concepts that are connected by means of an objectification.

**Definition 5-14:** \( CE_{\text{Obj},1} = \{ <a, b> \in C \mid \sigma^1(a) = b \} \) is an auxiliary set containing all pairs of concepts and knowledge primitives that are connected by means of an objectification.
Definition 5.15: \( CE = CE_{\text{Sub}} \cup CE_{\text{Ref}} \cup CE_{\text{Ref-1}} \cup CE_{\text{Obj}} \cup CE_{\text{Obj-1}} \) is an auxiliary set. It is the union of the five sets that were defined above, therefore containing all pairs of concepts or knowledge primitives that are somehow connected. Note that \( CE \subseteq (C \cup K) \times (C \cup K) \).

Definition 5.16: \( SC = \{ <a, b \in \varphi(C \cup K)> | \exists x \in a, y \in b : <x, y> \in CE \} \) is an auxiliary set. The elements of this set are pairs of sets containing concepts and knowledge primitives of which at least one element of a set is connected to an element of the other set.

Definition 5.17: \( D = \{ d \in \varphi(C \cup K) | \forall e \subseteq d : <e, d \setminus e> \in SC \} \) is an auxiliary set. It contains all coherent concept structures. The elements of \( D \) are sets of concepts and knowledge primitives. For each set that is element of \( D \), all possible partitionings into two sets will result in sets of concepts and knowledge primitives that are connected to each other by at least concept or knowledge primitive, or in other words, of the sets of concepts and knowledge primitives that are elements of \( D \), there is no partitioning possible such that there is no connection between the two partitioned sets. Therefore, the elements of \( D \) are coherent sets of concepts and knowledge primitives. They will be referred to as modeling constructs.

Definition 5.18: \( O \subseteq \{ <a \in D, b \in S, c \in D, d \in S> | b \neq d \land a \subseteq \text{cofa}(b) \land c \subseteq \text{cofa}(d) \} \) is called the set of coherency relationships. Elements of this set are 4-tuples \( <a, b, c, d> \) in which \( a \) is a modeling construct that is part of aspect model \( b \) and \( c \) is a modeling construct that is part of aspect model \( d \) of which the modeling constructs represent the same knowledge. \( O \) is a subset of all possible pairs of modeling constructs in different aspect models. Usually the modeling constructs consist of only one concept, e.g. the coherence relationship \( <\{'class\}', \text{classmodel}, \{'class\}', \text{sequencemodel}> \in O \) which indicates that the concept of a 'class' is used both in a class model as well as a sequence model with a similar meaning (OMG, 2001).

Definitions on Coherency Relationships

Definition 5.19: \( ct : O \rightarrow \{\text{SUBSET}, \text{EQUAL}, \text{OVERLAP}\} \) is a total function called the coherency type function. It maps coherency relationships to an element of the set that contains the possible coherency types that were described in section 5.2. For example, the UML classes mentioned in a sequence model are a subset of the classes mentioned in the class model. The coherency relationship is an elementary coherency relationship since it concerns only one concept, viz. the concept of a class. In this example, \( <\{'class\}', \text{classmodel}, \{'class\}', \text{sequencemodel}> \) is an element of \( O \). What can be said about it is that it is a subset coherency relationship, therefore, \( ct(<\{'class\}', \text{classmodel}, \{'class\}', \text{sequencemodel}>) = \text{SUBSET} \).
Chapter 5: A Proposal for Capturing Business Process Modeling Techniques

Definitions on Constraint Modeling

Definition 5-20: $\text{RQ} \subseteq K \times \mathbb{N}^*$ is the set of required roles. It contains all roles in a knowledge primitive that are required for all instances of the generic concepts corresponding to that role. E.g. given the concepts $C = \{\text{Person}, \text{Car}\}$, the knowledge primitive $K = \{\text{Owns}\}$ and the role mapping $R = \{\langle\text{Owns}, \langle\text{Person}, 1\rangle, \langle\text{Car}, 2\rangle\rangle\}$, we can specify that every car should have an owner by defining $\langle\text{Owns}, 2\rangle \in \text{RQ}$: The second role of the knowledge primitive Owns is obligatory. The set of required roles is comparable to obligation or totality constraints other role modeling approaches.

Definition 5-21: $\text{UC} \subseteq K \times \emptyset(\mathbb{N}^*)$ is the set of combinations of roles of which the combination of corresponding instances of generic concepts corresponding to these roles may occur only once in the set of individual knowledge primitives that belong to the generic knowledge primitive. E.g. given the concepts $C = \{\text{Man}, \text{Woman}\}$, the knowledge primitive $K = \{\text{MarriedTo}\}$ and the role mapping $R = \{\langle\text{MarriedTo}, \langle\text{Man}, 1\rangle, \langle\text{Woman}, 2\rangle\rangle\}$, we can specify that a man cannot be married to more than one woman by defining $\langle\text{MarriedTo}, \{1\}\rangle \in \text{UC}$ and that a woman cannot be married to more than one man by defining $\langle\text{MarriedTo}, \{2\}\rangle \in \text{UC}$.

5.4 Describing Modeling Methods

As said in chapter 2, a modeling method is viewed as a system consisting of interrelated modeling tasks that have to be carried out to solve a particular class of problems. In the case of modeling methods, the problem to be solved is to construct a conceptual model of a certain phenomenon.

The construction of a conceptual model often involves the construction of several aspect models. To distinguish the modeling tasks within a modeling method it is proposed to make use of a one-to-one relationship between aspect model and modeling task. Each task in the modeling method focuses on the construction of a single aspect model using a single aspect modeling technique. Examples of tasks are e.g. the task of modeling a DFD or the task of modeling an ERD. The relationship that orders individual tasks in time is the binary precedence relationship. In this subsection this way of describing a modeling method is worked out in detail.

The proposed framework allows the description of the individual tasks in a method as is depicted in Figure 5-13. The large rectangle represents the concept 'method'. That fact that a method consists of a number of tasks is represented by the smaller rectangles placed within the large rectangle that will be called 'task boxes'. The example represents a method consisting of five tasks that have to be carried out in arbitrary order.
To allow a description of a possible order in which tasks are carried out, it is proposed to include a binary precedence relationship between tasks. Arrows between two tasks represent these binary relationships. An arrow drawn between two tasks means that the task that the arrowhead points to can only be carried out after the task that the arrow departs from is finished. Figure 5-14 depicts a method in which task E can only be carried out after that tasks A and B are carried out, and task D can only be carried out after task B is carried out. There are no time constraints on the beginning and ending of task C.

Figure 5-15 shows the concept of the 'aggregated task'. An aggregated task is defined as a special kind of task that consists of a grouping of two or more (elementary or aggregated) subtasks. As a consequence, an 'elementary task' is defined as a task that is not an aggregated task.
Figure 5-15: Aggregated Task

Although several - more or less subjective - reasons for grouping tasks into an aggregated task can be thought of, such as the intuition that the grouped tasks are related to the same subject or performed by the same person, these criteria do not play a role here. The only criterion for grouping tasks in the proposed framework is to allow a more economic representation as becomes clear in Figure 5-15. Here methods A and B are equivalent, however, the grouping of tasks A, B and E in Method B allows a more economic representation of the precedence relationships.

Note that the precedence relationships that are represented by the arrows are constraints on the order of execution of tasks rather than triggers that initiate one task when the other is finished. Therefore, the diagram that is depicted in Figure 5-16 (A) does represent a deadlock situation where only task C can be carried out, rather than an iterative process where B, D and E are repeatedly executed after the execution of C. Figure 5-16 (B) introduces the use of a shadow behind the task box as the only proper representation to denote the iterative execution of certain tasks.

Figure 5-16: Deadlock and Iteration
The precedence relationship is a transitive relationship, e.g. the precedence relationship between B and D and the relationship between D and E in Figure 5-16 (B) implies that there is also a precedence relationship between B and E. For the reason of representational economy, an arrow between B and E representing this relationship is omitted.

The above-sketched structure is a general one. It holds for methods varying from the problem of fixing a car problem of which tasks could be diagnosis, reparation and testing, to the problem of baking of an egg in which the tasks range from selecting the proper pan to smashing the yolk. Since this thesis is about conceptual modeling methods rather than methods for backing eggs, now, the discussion narrows to what is specific for conceptual modeling methods.

The class of conceptual modeling methods distinguishes itself from the general class of methods in that the class of problems that is addressed by the method is the construction of conceptual models. A 'modeling task' is a task within a conceptual modeling method. Each modeling task deals with the construction of a single aspect model of the modeled phenomenon. Conceptual modeling techniques support the execution of a single modeling task.

Figure 5-17 shows how the proposed framework can be applied to show the coherence between modeling tasks and the aspects models that are constructed during these modeling tasks. The example explains the process of drawing a Data Flow Diagram (DFD) and an Entity Relationship Diagram (ERD) as introduced by Yourdon (1989) and further formalized here to serve as an example. Each of these tasks deals with the modeling of a certain aspect of the phenomenon. In the conceptual world we see the modeling knowledge that the modeler requires to construct the associated aspect model. The method shown below consists of two modeling tasks that are carried out in a particular order. The first task is to construct an ERD that contains instances of the generic concepts Entity and Relationship. The second task is to construct a DFD that contains instances of the generic concepts Process, Sub process and Flow. The instances of the generic concept 'Entity' that were constructed during the previous task can now be assigned to the flows in the DFD.
Figure 5-17: Coherence between modeling task and aspect model

It is likely that conceptual coherence between aspect modeling techniques influences the way that the corresponding task are ordered in time. If there is no conceptual coherence between two techniques, then there is no reason to order the corresponding tasks in time. If there is conceptual coherence, an ordering of the corresponding tasks instead of parallel execution is a means to avoid inconsistencies between the two models. This relationship will be dealt with later when the quality of techniques is discussed. In the example above, it has been chosen to first construct the DFD and afterwards the ERD. The entities that appear in the information flows of the DFD are later on worked out in the ERD.

5.5 Formalization of the Description of Methods

The language for capturing modeling methods is defined as a 3-tuple \( <T, P, I> \) as follows:
Definition 5.22: T is a set of tasks. For instance T = \{A, B, C, D, E\} in Figure 5.16(B).

Definition 5.23: P \subseteq T \times T is called the precedence relation. For all elements <a, b> of P, it holds that task a precedes task b. In the example presented in Figure 5.16(B), P = \{<C, B>, <C, D>, <C, E>, <B, D>, <D, E>, <B, E>\}. The precedence relationship is a transitive relationship.

Definition 5.24: I \subseteq \wp(T) is the set of iteration structures. It contains all sets of tasks that form a structure that is carried out more than once. An example is I = \{\{A\}, \{B, D, E\}\} in Figure 5.16(B).

The task assignment function that is defined below, links the tasks in the set T of the <T, P, I> tuple to the aspect models in set S of the <S, C, K, r, a, G, A, O> tuple:

Definition 5.25: \( ta : T \rightarrow S \) is a total function called the task assignment function. It assigns the aspect models in S to the tasks in T. One aspect model is assigned to each task. The function is bijective, therefore it has an inverse function \( ta^{-1} \) that maps a task to each aspect model. For example in Figure 5.17, given that T = \{DrawDFD, DrawERD\} and S = \{DFD, ERD\}, \( ta = \{<DrawDFD, DFD>, <DrawERD, ERD>\} \). Consequently, \( ta(DrawDFD) = DFD \) and \( ta^{-1}(DFD) = DrawDFD \).

The auxiliary set \( S_{\text{initial}} \subseteq S \) contains the aspect modeling techniques that have to be constructed as the first step of applying the modeling method: \( S_{\text{initial}} = \{ s \in S \mid \neg \exists t \in T : <t, ta^{-1}(s)> \in P \} \).

### 5.6 A Complete Description of Modeling Techniques

A complete description of a modeling technique, based on the proposed C-Me model, should contain the following components:

- Taking all of the above definitions together, including the way of modeling, the way of working as well as constraints on the way of modeling, a multi modeling method is defined as the following tuple: \( <<S, C, K, r, a, G, A, O>, RQ, UC, <T, P, I>, ta> \). A description of a multi modeling technique, depicted in (several) C-Me diagram(s), is an instance of such a tuple;

- Additional syntactic restrictions on concepts and knowledge primitives that cannot be expressed in the proposed language must be expressed in a more expressive language such as first order predicate logic. A description by means of a C-Me model and additional constraints allows the understanding of the syntactic quality of models and the influence of techniques on this quality;
A precise description of how the concepts and knowledge primitives of the technique are represented in diagrams by means of symbolic constructs must be captured, preferably by means of a legend. Such a description aids an understanding of the empirical quality of models and the influence of techniques on this quality;

A clear definition for each generic concept or knowledge primitive should be given in terms of generally agreed upon core concepts (e.g. state, transition, system). Such a description aids an understanding of the semantic quality of models and the influence of techniques on this quality;

5.7 Conclusions

In this chapter I have proposed the C-Me language as a language for capturing multi modeling techniques and methods. The meta modeling language for capturing the concepts and knowledge primitives can be compared to existing approaches such as entity relationship diagrams and object role models. The difference is that the C-Me language is specifically tailored for multi modeling languages that cover different perspectives and their interrelationships. It has special constructs for grouping concepts into aspect modeling techniques and for modeling relationships between aspect modeling techniques.

I have proposed a schema technique for capturing the order in which aspect modeling techniques are applied, a specification of the modeling method. This approach was formalized. The formalization of the technique for describing the method perspective is integrated with the formalization of the modeling techniques themselves. The modeling language for capturing modeling methods is very limited in expressiveness, presumed suitable enough to give insight into the order in which modeling tasks are carried out.

The approach is believed to be a good starting point for the evaluation of modeling techniques since it provides a basis for defining measurements with regard to individual aspect modeling techniques as well as their interrelationships. In the next chapter I will propose a number of measurements based on the C-Me modeling language. The application of the proposed C-Me language will take place in chapter 7.
6

A PROPOSAL FOR
MEASURING THE QUALITY OF
BUSINESS PROCESS MODELING TECHNIQUES

6.1 Introduction

In this chapter I will introduce the means to measure characteristics of modeling
techniques. The approach taken is to define a suite of metrics that are, to the extent
possible, anchored in the formalized C-Me language that was discussed before. The
chapter will start with a brief introduction to the kind of metrics that will be
introduced throughout the chapter (section 6.2). After this introduction I will
commence with the definition of the proposed metrics for assuring model
correctness (section 6.3). The inquiring metrics for assuring model usefulness are
presented in section 6.4.

6.2 Kinds of Metrics

In general, a metric is a procedure for calculating a value out of available data on a
certain phenomenon. It is presumed that this value says something about a certain
property of the phenomenon. For example, if we are interested in the safety of a car,
we could carry out a front impact test. Afterwards we measure how many
millimeters the steering wheel, driver's seat, etc. are displaced as a value indicative
for the (un)safety of the car. Another, less expensive, measure could be to simply
count the number of airbags and belts as a value indicative for the safety of the car.
Yet another measure could be to ask 20 experts to give a mark for the safety of the
car and calculate an average.

In this example, the phenomenon that is studied is the car, the property that we are
interested in is the safety of the car and we have proposed three metrics: One
measure that requires an experiment to be carried out, another measure that
requires an examination of the specifications of the car and yet another that
requires an inquiry amongst experts.
In fact, the three types of metrics distinguished are representative for the types of metrics used in practice. Using the different approaches to collect data and to calculate a value out of this data as a discriminator, three types of metrics can be distinguished. First of all, inquiring metrics calculate a measure by means of an inquiry among a representative amount of experts. They are asked to express their appreciation of a certain aspect of the phenomenon by means of a mark. The value of the metric is simply the average of the marks given by the experts. Other measures such as variance amongst answers is not considered here.

Another type of metric is the experimental metric. These kinds of metrics make use of experiments that involve one or more subjects using the phenomenon. In this class of metrics, measurements are derived out of the experimental data, such as the amount of errors that a subject makes using the phenomenon. Experimental and inquiring metrics for measuring the characteristics of modeling techniques are found e.g. in the work of Bajaj et al. (1999), discussed in chapter 3. Several measures are proposed, all of them requiring an experiment or inquiry to collect data out of which the values are calculated.

A third class of metrics, called theoretical or formal metrics, make use of a (formal) description of the phenomenon modeled. Measurements are calculated out of this data, without the necessity of involving people carrying out the experiment. Examples of theoretical metrics for measuring the characteristics of modeling techniques are the metrics proposed by Rossi et al. (1996), discussed in chapter 3. Values are calculated out of the meta model of the modeling technique that is investigated. Accuracy of the values calculated lies in the quality of the meta model of the technique. It should be obvious that a complete metric scheme can include both inquiring, experimental as well as theoretical metrics.

Table 6-1: Inquiring, Experimental and Theoretical Metrics

<table>
<thead>
<tr>
<th>Measurement Technique</th>
<th>Inquiring</th>
<th>Experimental</th>
<th>Theoretical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>questionnaire</td>
<td>experiment</td>
<td>calculation</td>
</tr>
<tr>
<td>Understanding of the phenomenon</td>
<td>not required</td>
<td>limited</td>
<td>full</td>
</tr>
<tr>
<td>Objectivity determined by</td>
<td>Expertise of the individual experts, representative number of experts, setup of the inquiry</td>
<td>Representative number of subjects, setup of the experiment</td>
<td>Quality of the meta model</td>
</tr>
</tbody>
</table>

Table 6-1 shows that an advantage of the use of theoretical metrics over inquiring and experimental metrics is that the measurement procedure of the first is less time consuming than the latter. The calculation of an inquiring or experimental measure requires the selection of a representative number of experts or subjects and the
carrying out of a questionnaire or experiment, whereas the calculation of a theoretical measurement is a formal derivation out of the description of the phenomenon. On the other hand, the application of a theoretical metric requires full understanding of the phenomenon by means of a formal description whereas the application of inquiring and experimental metrics do not, e.g. in the case of an inquiring metric one could ask experts about their opinion of what they think of the 'safety' of a 'car' without understanding what 'safety' or 'car' is oneself.

Another point worth mentioning is the difference between the sources of the objectiveness of measurement for each metric class. In case of both inquiring and experimental metrics, the objectiveness of the result is determined by the extent to which the size of the population is representative and the quality of the setup of the inquiry or experiment. In the case of an experimental metric, the data analysis is more comprehensive than simply calculating average opinions, as is the case with an inquiry. Therefore, the validity of the experimental setup such as the choice of hypotheses and dependent and independent variables introduces an extra source of subjectivism. For theoretical metrics, the process of calculation is free of human judgment. However, one has to assure that the source out of which these values are calculated is valid.

Especially with respect to the evaluation of modeling techniques there is a serious problem with the application of inquiring and experimental metrics. Inquiring metrics require modeling experts. In general, modeling experts are biased towards certain modeling techniques. E.g. in order to measure the quality of the UML compared to the IDEF by means of an inquiry amongst experts, the problem is to find experts that know UML and IDEF evenly well. The application of experimental metrics is surrounded by similar problems. E.g. in order to measure the quality of techniques one could set up an experiment to measure the number of mistakes in resulting models created by the modelers. However, this number of errors is not only influenced by properties of the technique but depends also by the modeler's knowledge of the techniques beforehand. In order to carry out such an experiment one has to be sure that the modelers know as much from e.g. UML as from e.g. IDEF which is quite difficult to assure.

For this reason, the aim of this chapter is to propose, to the extent possible, theoretical metrics for the measurement on modeling techniques and their resulting models. The C-Me description language and its formalization, described in the previous chapter, form the basis in which the proposed theoretical metrics are anchored. In the chapters to come, I will also address the question to what extent these measured values actually are indicative for the properties of modeling techniques defined in section 4.4.
6.3 Evaluating Modeling Techniques: Assuring Correctness

In section 4.3, I have defined the internal correctness of a model by addressing it on three semiotic levels, viz. empirics, syntax and semantics. In section 4.4 I have presented presumptions that address the properties of a modeling technique that are presumed to influence the quality of the resulting models. In this section I will propose the means to measure these properties. The properties will be revisited in the same order as they were introduced in section 4.4. It should be emphasized that the metrics are presented here without any practical example. The practical use and significance of the metrics is further explained in chapter 8.

Notational intuitiveness

The first property of a modeling technique (mentioned in section 4.4) presumed to be influencing the empirical correctness of the resulting models is the notational intuitiveness property. As any meta modeling language, the C-Me description language focuses on the syntax of the modeling language rather than the notation used in techniques. As such, a theoretical metric that is formally anchored in the description language cannot be given. The best that can be done is an inquiry amongst modeling experts, making use of a legend that links the modeling concepts mentioned in the meta model to the prescribed symbolic notations.

<table>
<thead>
<tr>
<th>Metric Name: AMI, WMI; Metric Type: Inquiring; Scale: Ratio [0..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
</tr>
<tr>
<td>For each aspect modeling technique</td>
</tr>
<tr>
<td>1 Are the symbols used for different generic concepts well</td>
</tr>
<tr>
<td>distinguishable (e.g. the notation for a UML Actor</td>
</tr>
<tr>
<td>compared to UML Use Case)?</td>
</tr>
<tr>
<td>2 Is the notation intuitive, i.e. in line with current and</td>
</tr>
<tr>
<td>past practice?</td>
</tr>
<tr>
<td>3 Are the symbols that are used different from standard</td>
</tr>
<tr>
<td>shapes such as rectangles, circles, ovals, diamonds, et</td>
</tr>
<tr>
<td>cetera?</td>
</tr>
<tr>
<td>For the complete multi modeling technique</td>
</tr>
<tr>
<td>4 Is the notation of a concept throughout different aspect</td>
</tr>
<tr>
<td>modeling techniques used in a consistent way? (e.g. a UML</td>
</tr>
<tr>
<td>object in an object diagram and a UML object in a</td>
</tr>
<tr>
<td>collaboration diagram)?</td>
</tr>
</tbody>
</table>

Figure 6-1: Questions belonging to the AMI and WMI metric

Above, the inquiring metrics *AMI (Aspect Model Intuitiveness)* and *WMI (Whole Model Intuitiveness)* are introduced. Figure 6-1 shows the inquiry and the type of questions that are asked. These questions summarize the rules given by Sindre (1990) and Rumbough (1996). The value for the AMI_{es} metric is calculated by
firstly decoding the answers given to the questions 1-3 for a particular aspect modeling technique in S from the depicted five value ordinal scale (ranging from + to -) to a [1..0] ratio scale. For each of these questions (1-3), the average answer is calculated out of the given answers (excluding the don't knows). Finally, the AMI metric is calculated by calculating the average of the first three average answers.

The WMI metric concerns the multi modeling technique as a whole rather than a single aspect modeling technique. The value for this metric is calculated by decoding the answers given to all four questions to a [1..0] ratio scale (1-3 for each aspect modeling technique, 4 only once for the multi modeling technique) and calculating the average of the answers for these questions in a similar way as was done for the AMI metric.

Both the AMI and WMI metrics yield a value within the [1..0] interval on a rational scale. It is presumed that the higher the value for the AMI and WMI metrics for a technique is, the higher the empirical quality of the resulting models will be.

**Syntactic Freedom**

Modeling techniques that offer a modeler much syntactic freedom, in other words, little restrictions on how to form expressions out of concepts, will presumably result in models with a low number of syntactical errors. The metric for measuring syntactic freedom is introduced here. Firstly, for each knowledge primitive in an aspect model, the Knowledge primitive Syntactic Freedom (KSF) is calculated and after that, the Aspect Model Syntactic Freedom (ASF) is simply the average of the KSF values of all the knowledge primitives in an aspect model.

The calculation scheme proposed for the KSF makes use of auxiliary functions $card$ and $all\_subroles$, defined earlier in section 5.3. $Card$ is the number of roles that a knowledge primitive has, $all\_subroles$ calculates the number of base concepts (concepts that are no generalizations) that can play a specific role in a given knowledge primitive. KSF calculates for each role in a knowledge primitive how many concepts can play the role. The definition is as follows:

<table>
<thead>
<tr>
<th>Metric Name: KSF; Metric Type: Formal; Scale: Ratio [0..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition 6-1: $KSF_{k \in K, s \in S} = \prod_{i=1_card(k)}^{all_subroles(k,i)} \frac{</td>
</tr>
</tbody>
</table>

Take the example depicted in Figure 5-11. In total there are three concepts in the DFD aspect modeling technique, viz. flow, entity and process. So, $|\overline{cofa(s)} \cap C_{Base}|=3$. The first role of flowEntity can only be played by flow, so $|all\_subroles(FlowEntity,1)|=1$. The same holds for $all\_subroles(FlowEntity,2)$. Therefore, $KSF(flowEntity) = 1/3 * 1/3 = 1/9$. Later on we will encounter examples
where the KSF for a given knowledge primitive is maximal, being 1. In such cases, there is no possibility to form syntactically incorrect expressions. The measurements on C-Me models where generalizations of concepts are involved are in particular interesting. We will encounter them later. As said, the ASF for a given aspect model is simply the average of all KSF of the knowledge primitives in that aspect model.

**Metric Name:** ASF; **Metric Type:** Formal; **Scale:** Ratio [0..1]

**Definition 6-2:**

\[
ASF_{s \in S} = \sum_{k \in cofa(s) \cap K} \frac{KSF_{k,s}}{|cofa(s) \cap K|}
\]

And again, for a multi modeling technique, the WSF is the average of all ASF’s.

**Metric Name:** WSF; **Metric Type:** Formal; **Scale:** Ratio [0..1]

**Definition 6-3:**

\[
WSF = \sum_{s \in S} \frac{ASF_s}{|S|}
\]

Also the intra model consistency of models is part of the syntactic quality of a model. Intra model inconsistencies result from the violation of uniqueness constraints on a knowledge primitive. E.g. a car has only one owner. If in the same model the expressions 'Jim owns the car 12345' and 'John owns the car 12345' occur, then a uniqueness constraint is violated and the model contains two statements that are inconsistent: Who is the owner, Jim or John? It is obvious that the number of knowledge primitives in an aspect modeling technique with a uniqueness constraint divided by the total number of knowledge primitives in that aspect model is an indication for the possibility that resulting models are intra model inconsistent. The AUC measures this for a given aspect modeling technique. Its definition is as follows:

**Metric Name:** AUC; **Metric Type:** Formal; **Scale:** Ratio [0..1]

**Definition 6-4:**

\[
AUC_{s \in S} = \frac{|cofa(s) \cap \{k \in K \mid \exists n \in \mathbb{N}^+: <k, n> \in UC\}|}{|cofa(s) \cap K|}
\]

Again, the WUC is simply the average of the AUC’s:

**Metric Name:** WUC; **Metric Type:** Formal; **Scale:** Ratio [0..1]

**Definition 6-5:**

\[
WUC = \sum_{s \in S} \frac{AUC_s}{|S|}
\]
A cause for models being malformed is the violation of a requiredness constraint on a knowledge primitive. An example of such a constraint is that each process model should have a begin state. Likewise as the AUC and WUC, the ARQ and WRQ measure the number of knowledge primitives in an aspect modeling technique that have at least a requiredness constraint divided by the total number of knowledge primitives in an aspect modeling technique:

**Metric Name: ARQ; Metric Type: Formal; Scale Ratio [0..1]**

**Definition 6-6:**

\[
\text{ARQ}_{s \in S} = \frac{|cofa(s) \cap \{k \in K \mid \exists n \in \mathbb{N}^+: <k, n> \in RQ\}|}{|cofa(s) \cap K|}
\]

Again, the WRQ is simply the average of the ARQ's of all aspect modeling techniques in a multi modeling technique:

**Metric Name: WRQ; Metric Type: Formal; Scale Ratio [0..1]**

**Definition 6-7:**

\[
\text{WRQ} = \sum_{s \in S} \frac{\text{ARQ}_s}{|S|}
\]

**Coherency**

In section 4.3, consistency between the different aspect models constructed with a modeling technique (inter model consistency) and inter model well-formedness were mentioned as important qualities on the syntactic layer of the semiotic ladder. Section 4.4 mentions the hypothesis that the (nature of the) conceptual coherency between the meta models of the respective aspect modeling techniques of a multi modeling technique influences the inter model consistency and inter model well-formedness of the resulting aspect models. In this section I propose a number of metrics for measuring the conceptual coherency between aspect modeling techniques. Parts of this work have been published in Hommes (2002).

The AMC (Aspect Model Coherence) measures the number of coherency relationships between two aspect modeling techniques belonging to the same multi modeling technique. It is a theoretical metric that is anchored in the formalization of the C-Me language given in chapter 5. The idea is to count those elements of O (set coherence relationships, see section 5.3) in which i and j appear as aspect models. This gives a measure for the number of coherency relationships between the two. The definition is therefore as follows:

**Metric Name: AMC; Metric Type: Formal; Scale: Absolute**

**Definition 6-8:**

\[
\text{AMC}_{i,j,s} = |\{<a, b, c, d> \in O \mid (b=i \land d=j) \lor (b=j \land d=i)\}|.
\]

123
Whereas the AMC metric can be used to determine the number of coherence relationships between a pair of aspect modeling techniques \( i, j \in S \), the AMCI metric calculates the average number of coherency relationships of an individual aspect modeling technique \( s \in S \). It is the average of all AMC's between the aspect modeling technique and the other aspect modeling techniques.

AMCI counts the total of coherency relationships that a certain aspect modeling technique \( s \in S \) has with the rest of the aspect modeling techniques within a multi modeling technique.

**Metric Name: AMCI; Metric Type: Formal; Scale: Ratio**

**Definition 6-9:**

\[
AMCI_{s \in S} = \sum_{\{i \in S | i \neq s\}} \frac{AMC_{s,i}}{|S|-1}
\]

Applied to the example mentioned in Figure 5-17, \( AMCDFD,ERD = 1 \), since there is only one coherency relationship between the two aspect modeling techniques. Note that the metric simply counts the coherency relationships and says nothing about the nature of these relationships. This will be dealt with later.

The AMCI0 and AMCI1 count the number of coherency relationships that a certain aspect modeling technique has with the other aspect modeling techniques in a multi modeling technique. AMCI0 simply counts each aspect modeling technique that a certain aspect modeling technique \( s \in S \) has at least a coherency relationship with. AMCI1 counts each aspect modeling technique that a certain aspect modeling technique \( s \in S \) has at least two coherency relationships with.

**Metric Name: AMCI0; Metric Type: Formal; Scale: Ratio**

**Definition 6-10:**

\[
AMCI0_{s \in S} = \sum_{\{i \in S | i \neq s\}} \frac{\{1, \text{ if } AMC_{s,i} > 0 \}}{|S|-1}
\]

**Metric Name: AMCI1; Metric Type: Formal; Scale: Ratio**

**Definition 6-11:**

\[
AMCI1_{s \in S} = \sum_{\{i \in S | i \neq s\}} \frac{\{1, \text{ if } AMC_{s,i} > 1 \}}{|S|-1}
\]

The \( WMC_s \) (Whole Model Coherence) measures the conceptual coherence of a multi modeling technique as a whole, and is defined as follows:
Metric Name: WMC; Metric Type: Formal; Scale: Ratio

\[
\text{Definition 6-12: } \quad \text{WMC}_x = \left\{ \begin{array}{ll}
1, & \text{if } \text{AMC}_{ij} > x, \\
0, & \text{else}
\end{array} \right. \\
\frac{\left|i \in S \mid i \neq j\right|}{\frac{1}{2} |S|(|S|-1)}
\]

The \( \text{WMC}_0 \) measures the total coherence between all aspect modeling techniques in a multi modeling technique. It counts all pairs of aspect modeling techniques that have at least one coherence relationship between them and divides this sum by the total number of possible coherency relationships between aspect modeling techniques. This yields a measure between \([0...1]\). In the example mentioned in Figure 5-17, the \( \text{WMC}_0 = 1 \): Since the technique contains only two aspect modeling techniques, \(|S|=2\), the total number of possible coherency relationships is 1 (only between DFD and ERD). And, in fact there is a coherency relationship between the two since \( \text{AMC}_{\text{DFD,ERD}} = 1 \). Therefore, the coherence is maximal.

A multi modeling technique offers per definition a set of coherent aspect modeling techniques. The \( \text{WMC}_0 \) gives an indication of the level of this coherence. The \( \text{WMC}_1 \) measure is meant to determine \textit{unwanted} coherency relationships between aspect models. \( \text{WMC}_1 \) counts all pairs of aspect modeling techniques that have more than one coherency relationship between them. It is presumed that the application of aspect modeling techniques that have more than one coherency between them results in models that are more difficult to keep consistent since so much the same knowledge is expressed in both models.

The \( \text{AMC} \) metric yields a value on the absolute scale whereas the \( \text{WMC} \) metric yields a value on the \([0..1]\) interval of the ratio scale. \( \text{WMC}_0 \) says something about the level of coherence irrespective of it having a positive or negative influence on the quality of resulting models. The higher the \( \text{WMC}_0 \) value, the more coherence there is between the individual aspect modeling techniques. \( \text{WMC}_1 \) says something about redundant coherency relationships. The higher the \( \text{WMC}_1 \) value, the more redundant coherency relationships there are. It is presumed that modeling techniques with a high \( \text{WMC}_1 \) value are less easy to keep consistent and well-formed.

Coherency relationships that form a potential hazard for inconsistent aspect models are those coherency relationships that contain knowledge primitives that have uniqueness constraints that can be violated. This set of coherency relationships is defined as follows:

\[
\text{Definition 6-13: } \text{O}_\text{UC} = \{<a,b,c,d> \in \text{O} \mid \exists k \in K, n \in \varphi(N^+) : k \in (a \cup c) \land <k, n> \in \text{UQ}\}
\]
In the above definition, $O_{UC}$ is a subset of $O$ that contains those elements $<a, b, c, d> \in O$ where there is a knowledge primitive $k$ that is an element of $a$ (knowledge primitives of aspect modeling technique $b$) or element of $c$ (knowledge primitives of aspect modeling technique $d$) with a uniqueness constraint on it ($k$ appears in $<k, n> \in UQ$).

Coherency relationships that form a potential hazard for mall-formed aspect models are those coherency relationships that prescribe that certain concepts or knowledge expressed in one aspect model must also be expressed in another aspect model. These are coherency relationships that either prescribe that the knowledge expressed in one aspect model must be equal to or a subset of the knowledge expressed in the other aspect model. This set of coherency relationships is defined as follows:

**Definition 6-14:** $O_{NQ} = \{o \in O \ | \ c(t(o)) = \text{EQUAL} \lor c(t(o)) = \text{SUBSET} \}$

The measures $AMCUC_{i,j \in S}$ and $AMCNO_{i,j \in S}$ are variants of the earlier proposed $AMC_{i,j \in S}$ measure. In these measures, the set of coherency relationships $O$ is replaced by respectively $O_{UC}$ and $O_{NQ}$:

**Metric Name:** AMCUC; **Metric Type:** Formal; **Scale:** Absolute

**Definition 6-15:** $AMCUC_{i,j \in S} = |\{<a, b, c, d> \in O_{UC} \ | \ (b=i \land d=j) \lor (b=j \land d=i) \}|$.

**Metric Name:** AMCNO; **Metric Type:** Formal; **Scale:** Absolute

**Definition 6-16:** $AMCNO_{i,j \in S} = |\{<a, b, c, d> \in O_{NQ} \ | \ (b=i \land d=j) \lor (b=j \land d=i) \}|$.

Whereas the AMCUC and AMCNO metrics can be used to determine the number of coherency relationships between a pair of aspect modeling techniques $i, j \in S$, the AMCIUC and AMCINO metrics calculate the average number of coherency relationships of an individual aspect modeling technique $s \in S$. It is the average of all AMCUC's and AMCNO's between the aspect modeling technique and the other aspect modeling techniques:

**Metric Name:** AMCIUC; **Metric Type:** Formal; **Scale:** Ratio

**Definition 6-17:**

$$AMCIUC_{s \in S} = \sum_{i \in S \mid i \neq s} \frac{AMCUC_{s,i}}{|S|-1}$$
Chapter 6: A Proposal for Measuring the Quality of Business Process Modeling Techniques

**Metric Name:** AMCINO; **Metric Type:** Formal; **Scale:** Ratio

**Definition 6-18:**

\[
AMCINO_{i,S} = \sum_{\{i \in S | i \neq s\}} \frac{AMCNO_{i,i}}{|S|-1}
\]

Finally, the WMCUQ and WMCNO are derived from the WMC0 metric and calculate the total coherence in a whole modeling technique:

**Metric Name:** WMCUC; **Metric Type:** Formal; **Scale:** Ratio

**Definition 6-19:**

\[
WMCUC = \sum_{i,j \in S} \frac{\begin{cases} 1, & \text{if } AMCUQ_{i,j} > 0, \\ 0, & \text{else} \end{cases}}{\frac{1}{2}|S|(|S|-1)}
\]

**Metric Name:** WMCNO; **Metric Type:** Formal; **Scale:** Ratio

**Definition 6-20:**

\[
WMCNO = \sum_{i,j \in S} \frac{\begin{cases} 1, & \text{if } AMCNO_{i,j} > 0, \\ 0, & \text{else} \end{cases}}{\frac{1}{2}|S|(|S|-1)}
\]

The ACC\(i,j \in S\) (Aspect model Coherence Complexity) measure determines to what degree concepts and knowledge primitives of an aspect modeling technique participate in coherency relationships. For example, within the UML, the interaction diagram and collaboration diagram are diagrams that allow the expression of exactly the same knowledge however using different notations. In this case the ACC\(_{\text{Collaboration, Interaction}} = 1.0\), the maximum value.

**Metric Name:** ACC; **Metric Type:** Formal; **Scale:** Ratio [0..1]

**Definition 6-21:**

\[
ACC_{i,j \in S} = \frac{|\{ c \in C \cup K \mid \exists a,b \in \phi(C \cup K) : (<a, i, b, j> \in O \vee <a, j, b, i> \in O) \land c \in (a \cup b)\}|}{|cova(i) \cup cova(j)|}
\]
The WCC metric measures the coherency complexity for the whole multi modeling technique.

Metric Name: WCC; Metric Type: Formal; Scale: Ratio [0..1]

Definition 6-22:
\[
WCC = \frac{\sum_{(i,j) \in S(i,j)} ACC_{ij}}{|S||S|-1}
\]

Determinism
An experimental metric is proposed to measure the determinism of a technique. The procedure proposed is to let a number of modeling experts apply the modeling technique on a single case and to measure the semantic variance in the resulting models. The semantic variance is measured as described below.

Figure 5-11: Example application on Data Flow Modeling [revisited]

For a single modeling concept in an aspect modeling technique, e.g. the concept 'Entity' in the example depicted in Figure 5-11, construct a table in which each row represents a subject participating in the experiment and each column represents an individual concept possibly mentioned by the subjects in the experiment. Each of the individual cells contains a 1 when a subject has included this individual concept in his or her model and a 0 when a subject has not included this individual concept in the model.

Table 6-2 shows an example where four entity relationship models are compared. E.g. the entity relationship model of subject S1 contains the entities Man, Woman and Marriage.
Table 6-2: Example Determinism Table

<table>
<thead>
<tr>
<th>Concept: Entity</th>
<th>Person</th>
<th>Man</th>
<th>Woman</th>
<th>Marriage</th>
<th>δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>S3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Σ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.25</td>
</tr>
</tbody>
</table>

For each pair of rows, the difference is calculated as follows: for each pair of cells in a column, (1,0) and (0,1) is counted as different and (1,1) is counted as equal. The overall difference between two rows is the number of different-pairs divided by the number of pairs not being (0,0). For each row, its average difference with the other rows can be calculated by comparing the row with all the other rows. The row(s) with the lowest average is the called the median solution. In the example above, S1 and S4 are the median solution. Differences with the median solution are depicted in the δ column: S3 differs 0.25 (Only Entity 'Person') and S2 differs 0.75 (all entities except 'Marriage'). Finally, the average δ is the called the concept semantic variance or CSV_{c \in C, s \in S} of a given generic concept c \in C in an aspect modeling technique s \in S, which is 0.25 in the example above.

**Metric Name:** CSV; **Metric Type:** Experimental; **Scale:** Ratio [0..1]

See description above

The Aspect Semantic Variance or ASV_{s \in S} is a formal metric based on the experimental CSV metric. That calculates the semantic variance of an aspect modeling technique by calculating the average CSV for each concept in that technique:

<table>
<thead>
<tr>
<th>Metric Name: ASV; Metric Type: Formal; Scale: Ratio [0..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition 6-23:</strong> ASV_{s \in S} = \sum_{c \in cofa(s) \cap C} \frac{CSV_{c,s}}{</td>
</tr>
</tbody>
</table>

Likewise, the Whole Model Semantic Variance or WSV calculates the semantic variance or determinism of the complete multi modeling technique:

**Metric Name:** WSV; **Metric Type:** Formal; **Scale:** Ratio [0..1]
Definition 6-24: \[ \text{WSV} = \sum_{s \in S} \frac{\text{ASV}_s}{|S|} \]

Complexity
An important indicator for the complexity of a modeling technique is the size of the meta model. The more concepts a technique offers, the more difficult it is to apply the technique, which may result in all kinds of errors in the resulting models. The size of a multi modeling technique is called the Whole Model Size (WMS) and is simply defined as the total of base concepts and knowledge primitives. Anchored in the C-Me formalization, this measure can be defined as:

| Metric Name: WMS; Metric Type: Formal; Scale: Absolute |
| Definition 6-25: WMS = |C_{Base}| + |K| |

The size of meta models of individual aspect modeling techniques can be measured by applying the Aspect Model Size (AMS) metric. The AMS metric is as follows:

| Metric Name: AMS; Metric Type: Formal; Scale: Absolute |
| Definition 6-26: AMS_{s \in S} = |\{c \in C_{Base} \cup K | <c, i> \in A \}| |

Both the WMS and AMS are theoretical metrics that result in a value on an absolute scale.

The Aspect Model Knowledge Increment or AKI_{s \in S} metric is a process measure to determine the number of new concepts and knowledge that are introduced in by applying a certain aspect modeling technique s \in S. It takes into account the already introduced concepts by preceding modeling tasks and calculates the percentage of newly introduced concepts and knowledge. For initial aspect modeling techniques, techniques that are not preceded by other techniques in the process (S_{Initial} = \{s \in S | \exists t \in T: <t, ta^{\perp}(s)> \in P\}, see chapter 5) this knowledge increment is of course 100%, since all concepts are newly introduced. For aspect modeling techniques that are not initial, i.e. preceded by other techniques in the process (s \in S \setminus S_{Initial}), the percentage is calculated by determining the average newly introduced knowledge primitives or concepts for each of the preceding techniques.

| Metric Name: AKI; Metric Type: Formal; Scale: Ratio [0..1] |
System Expressiveness

Based on the findings with regard to assuring the usefulness of business process models (section 4.5), an inquiring metric will be proposed here to measure the extent to which the concepts and knowledge primitives as introduced in the discrete dynamic paradigm as introduced in section 4.5.

Table 6.3: Questions belonging to the WSE metric

Table: Metric Name, WSE; Metric Type: Inquiring, Scale: Ratio, Range: [0, 1]

1. System state space
   - System state space
   - Don't know

6.4 Evaluating Modeling Techniques: Assuring Usefulness

In this example there are two aspect modeling techniques, S = {SequenceModel}, Classtask, two tasks, \( T = \{\text{DrawSequence, DrawClass}\} \), and a precedence \( P = \langle\text{DrawSequence, DrawClasstask}\rangle \). For the process step \( \langle\text{DrawSequence, DrawClasstask}\rangle \) and a precedence \( P = \langle\text{DrawSequence, DrawClasstask}\rangle \), the number of new concepts introduced is as said less then 100%, since the concept of a Class is already introduced in the sequence diagram, therefore the set of newly introduced concepts will not contain the element Class.

The higher the AKI value for a certain aspect modeling technique, the more the accompanying process step contributes to the model. On the other hand, the lower the AKI value, the less the corresponding process step contributes to the model. An example of such a relation is e.g. the construction of a UML class diagram. The UML sequence model already introduced the concept of a class. Classes that have already been defined in the sequence diagram can be used and further specified in the succeeding step where the class model is constructed.

Definition 6.27:

\[ AKI[S] = \frac{\sum_{t \in T} \text{cofix}(s) \backslash \text{cofix}(t)}{|\text{cofix}(s)|} \]
An important requirement is that the modeling techniques are capable of modeling discrete dynamic systems in a consistent way and in addition to this the role that actors play causing transitions in the system that they act upon and inspecting the state of other systems. The proposed WSE metric should therefore inquire whether a given modeling technique is capable of modeling the concepts provided by this paradigm.

**Business Suitability**

The WBS metric (*Whole model Business Suitability*) is an inquiry to measure the degree to which a modeling technique is capable of modeling the business domain. Whereas the previous metric, the WSE metric, measure the capability to model discrete dynamic systems, the WBS inquires about the kind of discrete dynamic systems that are modeled. Here we have chosen to measure the business suitability for modeling techniques that model business systems for the purpose of implementing ICT solutions. Therefore, the capability of modeling business transitions (including all kinds of transitions mentioned in section 4.5), the capability of modeling information transitions and the capability of modeling communication transitions have been chosen as requirements together with the consistency between these systems.

**Table 6-4: Questions belonging to the WBS metric**

<table>
<thead>
<tr>
<th>Metric Name: WBS; Metric Type: Inquiring; Scale: Ratio [0..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Categories Covered</strong></td>
</tr>
<tr>
<td><strong>+</strong></td>
</tr>
<tr>
<td><strong>Business System</strong></td>
</tr>
<tr>
<td>Criterion for distinguishing transitions</td>
</tr>
<tr>
<td>Criterion for distinguishing actors</td>
</tr>
<tr>
<td><strong>Information System</strong></td>
</tr>
<tr>
<td>Criterion for distinguishing transitions</td>
</tr>
<tr>
<td>Criterion for distinguishing actors</td>
</tr>
<tr>
<td><strong>Communication System</strong></td>
</tr>
<tr>
<td>Criterion for distinguishing transitions</td>
</tr>
<tr>
<td>Criterion for distinguishing actors</td>
</tr>
<tr>
<td><strong>System Interrelationships</strong></td>
</tr>
<tr>
<td>Formally defined interrelationships</td>
</tr>
<tr>
<td><strong>Total Scores</strong></td>
</tr>
</tbody>
</table>

132
6.5 Conclusions

In this chapter I have proposed a number of formal, experimental and inquiring metrics that are presumed to be suitable for the evaluation of modeling techniques. Before we proceed with the application of the framework that has been proposed in the last three chapters I would like to present an overview of the proposed measures. This overview is an extension of the overview presented in section 4.6.

Table 6-5: Technique Properties and Metrics

<table>
<thead>
<tr>
<th>Property</th>
<th>Proposed Metric</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notational</td>
<td>AMI (Aspect Model Intuitiveness)</td>
<td>Inquiring</td>
</tr>
<tr>
<td>intuitiveness</td>
<td>WMI (Whole Model Intuitiveness)</td>
<td>Inquiring</td>
</tr>
<tr>
<td>Syntactic Freedom</td>
<td>KSF (Knowledge Primitive Syntactic Freedom)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>ASF (Aspect Model Syntactic Freedom)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WSF (Whole Model Syntactic Freedom)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AUC (Aspect Model Unicity Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WUC (Whole Model Unicity Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>ARQ (Aspect Model Requiredness Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WRQ (Whole Model Requiredness Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td>Coherence</td>
<td>AMC (Aspect Model Coherence)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCI0 (Aspect Model Individual Coherence &gt;0)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCI1 (Aspect Model Individual Coherence &gt;1)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WMC (Whole Model Coherence)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCUCE (Aspect Model Coherence with Unicity Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCN0 (Aspect Model Coherence Not of type Overlapping)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCUCE (Aspect Model Individual Coherence with Unicity Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AMCUNO (Aspect Model Individual Coherence Not of type Overlapping)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WMCUCE (Whole Model Coherence with Unicity Constraints)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WMCNO (Whole Model Coherence Not of type Overlapping)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>ACC (Aspect Model Coherence Complexity)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WCC (Whole Model Coherence Complexity)</td>
<td>Formal</td>
</tr>
<tr>
<td>Determinism</td>
<td>CSV (Concept Semantic Variance)</td>
<td>Experimental</td>
</tr>
<tr>
<td></td>
<td>ASV (Aspect Model Semantic Variance)</td>
<td>Experimental</td>
</tr>
<tr>
<td></td>
<td>WSV (Whole Model Semantic Variance)</td>
<td>Experimental</td>
</tr>
<tr>
<td>Complexity</td>
<td>AMS (Aspect Model Size)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>WMS (Whole Model Size)</td>
<td>Formal</td>
</tr>
<tr>
<td></td>
<td>AKI (Aspect Model Knowledge Increment)</td>
<td>Formal</td>
</tr>
</tbody>
</table>
The formal metrics proposed can be calculated out of the C-Me description of a modeling technique. The accuracy of these metrics is therefore determined by the accuracy of the C-Me descriptions.

In order to measure determinism, a number of experimental metrics are proposed. These experiments are based on a representative number of modelers that model the single case using a single technique. Here, the level of accuracy is determined by the extent that the population is representative.

Furthermore, the proposed experimental CSV metric requires some human judgment that might influence its accuracy. The ASV is calculated out the CSV values of all concepts in the aspect modeling technique and the WSV out of all ASV values. Since the determination of the CSV value of a single concept is expected to require a considerable amount of effort and by consequence the ASV and WSV require an even larger amount of effort, it is not realistic that the ASV and WSV values can be calculated as proposed here.

The inquiring metrics require several experts to evaluate the modeling technique under study and to fill in a questionnaire form. Here the level of accuracy depends on the expertise of the modeling experts and the number of responders.

This chapter concludes the proposal of an integrated framework for capturing and measuring the quality of modeling techniques for business modeling. In the next chapters we will proceed with the application of the proposed framework. Chapter 7 discusses the application of the C-Me description language and chapter 8 will in particular dwell upon the application of the metrics that were proposed in this chapter.
Part III

Application and Validation
7 APPLICATION AND VALIDATION OF THE C-ME DESCRIPTION LANGUAGE

7.1 Introduction

In order to demonstrate the plausibility of the theory proposed in the previous three chapters, this chapter concerns the application of the C-Me description language. It will be applied to describe four multi modeling techniques for business process modeling. The four modeling techniques are selected and to a certain extent adjusted to satisfy two important criteria: 1) the chosen techniques should as much as possible be a reflection of the current state of the art in business process modeling and 2) the chosen multi modeling techniques should be suitable for carrying out an empirical study later on (described in chapter 8) to validate the theoretical measurements obtained in this chapter. The following four multi modeling techniques are described and evaluated:

- **Event Driven Process Chains (EPC's)** - A multi modeling technique that is discussed in numerous scientific publications and used widely throughout the world, mainly for analyzing processes for the purpose of ERP systems implementation. It contains four aspect modeling techniques, called views: the process view, data view (based on Entity Relationship Modeling), functional view and organizational view. EPC's were selected for this discussion because they are widely used throughout the world.

- **Dynamic Essential Modeling of Organizations (DEMO)** - A multi modeling technique developed in Delft University of Technology (Dietz, 1996, Reijswoud et al, 1999), rooted in the language action perspective (Austin, 1962, Searle, 1969). It offers a number of aspect modeling techniques of which only four are discussed in this chapter: the business architecture model, the process phase model, the transaction-result table and the bank-fact table. DEMO was selected as a representative of a novel family of Language Action oriented techniques. Only a limited number of aspect modeling techniques is discussed to assure that the overall complexity of the technique is comparable to the other techniques.
**Petri-net Based Modeling (PBM)** - A hypothetical multi modeling technique, derived from the Dynamic Enterprise Modeling technique developed by Baan, a Dutch vendor of ERP systems. It offers a number of aspect modeling techniques of which only three are selected and to a certain extent adjusted to fit the purpose of this chapter. The basic principles of the Business Control Model and Business Process Model are taken from the original approach (Post et al. 1996), leaving out some details. The Business Information Model, which is originally based on Entity Relationship Modeling, was replaced by a hypothetical Business Information Model based on the Object Role Modeling approach. This replacement was done to allow a comparison between an Entity Relationship Modeling approach (contained in the EPC’s) and an Object Role Modeling Approach (contained in the hypothetical PBM Business Information Model). PBM was chosen since the Petri-net modeling approach is used throughout the world (see e.g. Aalst, 2002).

**Business Modeling with UML (UBM)** – An extension on the well-known UML for the modeling of software artifacts, proposed by Booch, Jacobson and Rumbough (Booch et al., 1999). The extensions discussed here are the so-called Penker extensions (Eriksson et al., 2000) for modeling business processes. Four aspect modeling techniques are discussed: the process model, split into a high level and a low level process model (UML Activity Modeling), the goal model and the conceptual model (UML Class Modeling). The UBM approach was chosen because it seems to become the modeling standard for modeling almost any phenomenon, including business processes.

The remainder of this chapter is organized as follows. Each of the four techniques will be discussed and evaluated in four consecutive sections. Each of these sections contains a description of the technique by briefly discussing the way of thinking behind the approach as well as used notations, underlying modeling concepts and example diagrams. Furthermore, each section contains the application of the proposed C-Me language.

### 7.2 Event Driven Process Chains

The version of the EPC’s that is discussed here contains four views: the process view, the data view, the functional view and the organizational view. The process view can be considered the core view of the approach; all other views are to a large extent derived from the process view. A process view is constructed for each of the business processes of an organization. A business process is treated as a chain of events (the occurrence of a state transition) and functions (actions causing state transitions). These functions and events can be combined by using OR splits, AND splits and XOR splits. Organizational units can be assigned to functions. These organizational units are further elaborated in the organizational view.
Furthermore, one or more input or output data units (comparable to Entities in Entity Relationship Modeling) can be assigned to functions. These entities are further elaborated in the data view. The functional view again depicts all functions that appear in the process view, structured by means of a decomposition relationship. Table 7-1 shows the modeling constructs that comprise the process view.

Table 7-1: EPC Process View

<table>
<thead>
<tr>
<th>Construct</th>
<th>ASPECT MODEL TABLE: PROCESS VIEW</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td><img src="image" alt="name" /></td>
<td>The occurrence of a transition from one state to another state in the business system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Each process diagram starts with an event</td>
</tr>
<tr>
<td>Function</td>
<td><img src="image" alt="name" /></td>
<td>Action that causes a state change in the business system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* The performance of a task causes a transition from one system state to another</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Functions also appear in the functional view aspect modeling technique</td>
</tr>
<tr>
<td>AND-Split/Join</td>
<td><img src="image" alt="image" /></td>
<td>Connectors are used to create parallel processes, exclusiveness, to combine functions and events.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Read more on connectors and joins in the remainder of the table</td>
</tr>
<tr>
<td>OR-Split/Join</td>
<td><img src="image" alt="image" /></td>
<td></td>
</tr>
<tr>
<td>XOR-Split/Join</td>
<td><img src="image" alt="image" /></td>
<td></td>
</tr>
<tr>
<td>Application System</td>
<td><img src="image" alt="image" /></td>
<td>Automated computer system that offers functionality in support of certain functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Entity types appear also in the Data View Aspect Modeling Technique</td>
</tr>
<tr>
<td>Entity Type</td>
<td><img src="image" alt="image" /></td>
<td>A class of things that share the same properties</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Positions appear also in the Organizational View Aspect Modeling Technique</td>
</tr>
<tr>
<td>Position</td>
<td><img src="image" alt="image" /></td>
<td>Responsibility for performing a single task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* An event triggers exactly one function and one function is triggered by exactly one event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* A is called the triggering event of B</td>
</tr>
<tr>
<td>Control Flow</td>
<td><img src="image" alt="image" /></td>
<td>Event A triggers function B</td>
</tr>
</tbody>
</table>
Figure 7.1 shows an example process view. The function 'assign product to sales offices' starts when either the event 'enter new products' or the event 'enter products provided by new suppliers' occurs. The result of this function is the occurrence of the event 'product assigned'. The function reads from the entity 'product' and writes to the entity 'order record'. The organizational unit responsible for the function is the unit 'distribution'. In order to perform the function, 'merchandise handling software' is used.
Figure 7-1: Example EPC Process View

In the function view, all functions mentioned in the process view are further elaborated by means of a functional decomposition.

Table 7-2: EPC Functional View

<table>
<thead>
<tr>
<th>Construct</th>
<th>ASPECT MODEL TABLE: FUNCTIONAL VIEW</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td></td>
<td>A task that must be fulfilled as part of a business process</td>
<td>* Functions also appear in the process view aspect modeling technique</td>
</tr>
<tr>
<td>Function Decomposition</td>
<td></td>
<td>B and C are sub functions of A. A is the super function of B and C.</td>
<td>* A function can be decomposed into a number of sub functions. One sub function has only one super function. Functions that are super functions do not appear in the process view aspect modeling technique</td>
</tr>
</tbody>
</table>

Figure 7-2 shows an example of such a functional decomposition.
Figure 7-2: Example EPC Functional View

All entity types mentioned in the process view are worked out in a traditional Entity Relationship Diagram (ERD) in which all entities mentioned in the process view appear.

Table 7-3: EPC Data View

<table>
<thead>
<tr>
<th>Construct</th>
<th>ASPECT MODEL TABLE: DATA VIEW</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Type</td>
<td></td>
<td>Entity Type. A class of things that share the same properties</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>* Entity types appear also in the Process View Aspect Modeling Technique</td>
</tr>
<tr>
<td>Relationship Type</td>
<td>name</td>
<td>A class of relationships between two or more entity types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* Read more below</td>
</tr>
</tbody>
</table>
Figure 7-3 shows an example of such an ERD.

![ERD Diagram]

**Figure 7-3: Example EPC Data View**

Finally, all positions that appear in the process view are worked out by means of the organizational view, which can best be compared to an organigram. All positions mentioned in the process view appear in this diagram. Furthermore, individual positions are assigned to larger organizational units.

**Table 7-4: EPC Organizational View**

<table>
<thead>
<tr>
<th>Construct</th>
<th>Notation</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Unit</td>
<td>name</td>
<td>Part of the organization responsible for performing certain tasks</td>
<td></td>
</tr>
</tbody>
</table>
Figure 7-4 shows an example in which the organizational unit 'accounting' is assigned a sub organizational unit 'external accounting'. This latter organizational unit contains the positions 'G/L Account manager' and 'Personnel Account Manager'.

![Diagram](image)

**Figure 7-4: Example EPC Organizational View**

**Proposed Way of Working**

The way of working that is proposed here, and followed later on in the empirical study reported in chapter 8, is to start by constructing several process views. Each process view addresses a particular business process in the organization that is modeled. These process views offer a starting point for constructing the data-, organizational- and functional views. There is no particular order in which these three views must be constructed.
C-Me Models

Figure 7-5 shows the EPC process view aspect modeling technique. The core concept of the process view is the connector, which is further specialized. A connector can be a function, an event, a split (further specialized as OR, AND and XOR split), and a join (further specialized as OR, AND and XOR join). Between all of these connectors, precedence relations can exist.

The knowledge primitive 'Performs' expresses that functions are performed by positions. The knowledge primitive 'Supports' expresses that functions are supported by application systems. The knowledge primitives 'Reads instance of' and 'Update instance of' express that functions can respectively read and update instances of a certain entity type.

Figure 7-5: C-Me Model of the EPC Process View

Figure 7-6 shows the EPC Data View. Core concept of this view is the entity type. Entity types can be involved in two types of relationships with each other: the basic
relationship denoted by the concept Relationship Type and the generalization relationship denoted by the concept Generalization. A basic relationship between entities can be binary (a relationship between two entity types, e.g. a car and its owner), ternary (a reparation performed on a car by a certain employee), quaternary, and so on. The role that an entity plays in a relationship e.g. being the car in the ternary relationship reparation on a car by an employee is denoted by the concept Role. So, a relationship can have a number of roles, each being played by an entity type. For such a role it is possible to specify the multiplicity value. E.g. the multiplicity of the role that car plays in the relationship 'car and its owner' is 1 since a car has exactly one owner. Therefore, both max as well as min multiplicity value is 1. On the other hand, the multiplicity of the role owner in the same relationship is 0..n. The min value is 0 and the max value is n, since a person can own zero, one or more cars. Multiplicity is denoted by the concept Multiplicity Value.

![EPC Data View](image)

**Figure 7-6: C-Ne Model of the EPC Data View**

With regard to the Generalization relationship, the following can be said. Again, several entities can play a role in a generalization, being a super type or a sub type in the generalization. E.g. a employee being a special kind of person or a race car being a special kind of car. In the first example, employee is the sub type and person is the super type. In the second example, race car is the sub type and car is
the super type. Generalization is denoted by the concept Generalization. A generalization can be disjunct or not and total or not. Disjunct means that a certain instance of a super type cannot be an instance of more than one of the sub types. Take for instance the entity types man and woman both being sub types of the entity type person. This generalization is disjunct since a person cannot be a man and a woman at the same time. A generalization is total when a certain entity that plays a role in the generalization is obliged to be one of the sub types. In case of the last example, a person has to be a man or a woman. A person cannot be a person without being a man or a woman. With regard to the race car example, a car does not necessarily have to be a race car, so this generalization is not total.

![EPC Process View](image)

**Figure 7-7: C-Me Model of the EPC way of modeling**

As can be seen in Figure 7-7, functions, entity types and positions are worked out in more detail in respectively the function-, data- and organizational view. The function view is rather simple. Next to the concept function, it contains the
knowledge primitive decomposition that allows the decomposition of a function into sub functions. The organizational view contains a comparable decomposition of organizational units in the form of the knowledge primitive 'supervises'. Furthermore, positions are assigned to organizational units by means of the 'in' knowledge primitive.

The application of the C-Me modeling language to depict the proposed way of working results in the diagram depicted below in Figure 7-8. We see that the first step is to construct a number of process views. After that, the organizational-, data- and functional view are constructed without any particular ordering.

![EPC Method](image)

_Figure 7-8: C-Me Model of the EPC way of working_

See appendix A for a detailed description of the formalization.

### 7.3 Dynamic Essential Modeling of Organizations

The version of the Dynamic Essential Modeling Organizations technique that is discussed here contains four aspect modeling techniques: the business architecture model, the process phase model, the transaction result table and the bank fact table. This is a subset of the aspect modeling techniques that are currently offered by the DEMO methodology (Dietz, 2002, Dietz, 2003). A limited selection has been made because the same set will be used later on in an experimental setup for validation purposes. In this experiment, the number of aspect modeling per technique was limited to 3 or 4 aspect modeling techniques.

Interesting aspect of the DEMO technique is that it makes a clear distinction between transitions in the inter subject system and transitions in the object system. Transitions in the inter subject system are communicative actions, such as to request, to promise, to state or to accept something. These transitions change the state of the system in which subjects negotiate, the inter subject system. Transitions concerning the actual delivery of a product or service to a customer take
place in the object system. In the inter subject system, actors negotiate about the transitions that take place in the object system.

Transitions in the inter subject system take place in a regular pattern (Reijswoud, 1996). A derived pattern is depicted in Figure 7-9. This negotiation pattern is called a business transaction. Part of the business transaction is also the transition in the object world that the negotiation is about. There are always two parties involved in a business transaction, the initiator and the executor. The initiator starts the transaction by requesting the transition in the object world. The executor is always the one that actually carries out this transaction. After the transition in the object system is carried out, the conversation on the acceptance of this transition is further carried out in the inter subject system. Looking at a transaction from a time perspective, we can distinguish an Order phase in which initiator and executor negotiate about the transition that has to take place in the object world, an execution phase in which the executor carries out this transition and a result phase in which initiator and executor negotiate about the result of the transaction.

Figure 7-9: DEMO Transaction Pattern

Table 7-5 shows the modeling constructs that comprise the Business Architecture Diagram.
Table 7-5: DEMO Business Architecture Diagram

<table>
<thead>
<tr>
<th>Construct</th>
<th>Notation</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| Transaction          | ![Diagram](nr) | Pattern of communicative actions between two aggregated or elementary actors, containing a production action, resulting in a production fact. | - Numbering: T1, T2, T3, etc.  
- Naming: Past Perfect, e.g. "delivery", "reparation", "payment"  
- The production fact that is the result of the transaction is specified in the transaction result table, see later on. |
| Elementary Actor     | ![Diagram](nr) | The element in an organization that is competent, authorized and responsible for performing one type of production action. | - Numbering: A1, A2, A3, etc.  
- Naming in accordance with transactions as follows: 'deliverer', 'reparator', 'payer'  
- Actors are responsibilities, no persons or organizational units! |
| Aggregated Actor     | ![Diagram](nr) | A grouping of elementary actors to form a system. | - Numbering: AA1, AA2, AA3, etc.  
- The rounded box, called the system border, is used to depict the system that is modeled in detail, see composition later on.  
The gray box is used to depict systems in the environment of the modeled system  
- Example names: Hotel, Garage, Pizzeria. |
| Bank                 | ![Diagram](nr) | Container of production facts that are not produced by the organization itself. | - Numbering: B1, B2, B3, etc.  
- Production Facts contained in it are specified in the bank-fact table, see later on. |
| Authoritative        | ![Diagram](nr) | Conversation in which one aggregated or elementary actor asks another for authorization to perform a communicative or production act. | - Numbering: AC1, AC2, AC3, etc. |
| Conversation         | ![Diagram](nr) |                                                                                   |                                                                         |
| Persuasive           | ![Diagram](nr) | Conversation in which one aggregated or elementary actor persuades another for performing a communicative or production act. | - Numbering: PC1, PC2, PC3, etc. |
| Composition | The aggregate actor depicted by the rounded gray box contains the elementary actors depicted in it | Both aggregated actors as well as elementary actors can participate in this relation  
A single transaction can be initiated by several other actors  
One actor can initiate several transactions |
|---|---|---|
| Initiator Link | The aggregated or elementary actor is the initiating party of this transaction | Again, both aggregated actors as well as elementary actors can participate in this relation  
Again, A single conversation can be initiated by several other actors  
Again, One actor can initiate several conversations |
| Responder Link | The elementary actor is responsible for carrying out the production action specified in the transaction | Numbering and naming of transaction and actor should correspond: A1/T1, A2/T2, etc.  
Naming should correspond as well: e.g. the actor responsible for carrying out T1 Delivery is called A1 Deliverer  
An elementary actor is responsible for exactly one transaction  
A transaction is performed by exactly one elementary actor |
| | An elementary actor within this aggregated actor is responsible for the production action specified in the transaction | An aggregated actor can be responsible for carrying out several transactions |
| | The elementary actor is the responding party for this conversation | Again, for elementary responding actors:  
Numbering and naming of conversation and actor should correspond: A1/AC1, A2/PC2, etc.  
An elementary actor is the responder of exactly one conversation  
A conversation has only one responding actor  
Remarks do not hold for aggregated responding actors |
Figure 7-10 shows an example of a Business Architecture Diagram.

**Figure 7-10: Example DEMO Business Architecture Diagram**

The Process Phase Diagram again depicts the transactions and conversations identified earlier. In addition to this it allows to model whether a conversation is started externally, such as e.g. transactions T01 and T02 in the example above, as well as a further specification of the initiation of internal transactions and possible wait conditions between transaction phases. Table 7-6 shows the model table containing the constructs that comprise the DEMO Process Phase Diagram.
### Table 7-6: DEMO Process Phase Diagram

<table>
<thead>
<tr>
<th>Construct</th>
<th>Notation</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction</td>
<td><img src="image" alt="Diagram" /></td>
<td>Same as in Business Architecture Diagram</td>
<td>Compared to the Business Architecture Diagram, the depicted symbol is no stretched horizontally showing the Order, Execution and Result phase of the transaction, recall page 1 of this fact sheet.</td>
</tr>
<tr>
<td>Authoritative</td>
<td><img src="image" alt="Diagram" /></td>
<td>Same as in Business Architecture Diagram</td>
<td>Again, compared to the Business Architecture Diagram, the depicted symbols are stretched horizontally</td>
</tr>
<tr>
<td>Conversation</td>
<td><img src="image" alt="Diagram" /></td>
<td>Same as in Business Architecture Diagram</td>
<td></td>
</tr>
<tr>
<td>Persuasive Conversation</td>
<td><img src="image" alt="Diagram" /></td>
<td>Same as in Business Architecture Diagram</td>
<td></td>
</tr>
<tr>
<td>External Initiation</td>
<td><img src="image" alt="Diagram" /></td>
<td>Conversation or Transaction A is initiated externally: by an actor in the environment of the aggregated actor that is modeled in detail</td>
<td>Keep in mind the consistency with the Business Architecture diagram. Externally initiated transactions or conversations are started from outside the system border in the Business Architecture Diagram</td>
</tr>
<tr>
<td>Internal Initiation</td>
<td><img src="image" alt="Diagram" /></td>
<td>Conversation or Transaction A is initiated internally during the O, E or R phase of another transaction or conversation</td>
<td>Arrows can be drawn from the O, E and R phase. In this example it is drawn from the E phase. Keep in mind the consistency with the Business Architecture Diagram. If Tx is initiated during Ty, then Tx is initiated by Ay in the Business Architecture Diagram. In case a transaction or conversation is started during a conversation, it is not possible to specify the phase in which the transaction or conversation is started.</td>
</tr>
<tr>
<td>Wait Link</td>
<td><img src="image" alt="Diagram" /></td>
<td>In order for B to proceed with the next phase, it has to wait on a phase transition in A</td>
<td>In the example, in order for B to proceed with the R (result phase), A has to be in the R phase. The start point and end point (arrowhead) of the wait link can be drawn: At the beginning of the O phase, between O and E, between E and R (see example) and at the end of the R phase.</td>
</tr>
</tbody>
</table>

Figure 7-11 shows an example of a process phase diagram. This diagram is consistent with the business architecture diagram depicted above.
As said, each transaction contains a production action. The result of this production action is a production fact. The transaction result table specifies which production fact belongs to which transaction. The table contains two columns, one that specifies the transaction, the other that specifies the resulting production fact. A production fact is described by a sentence. Variables are used to refer to the classes of things (object types) that are relevant for the process. An example diagram is depicted below.

Table 7-7: Example of a Transaction Result Table

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Production Fact</th>
</tr>
</thead>
<tbody>
<tr>
<td>T01 Start Insurance</td>
<td>Insurance I is started</td>
</tr>
<tr>
<td>T02 Disburse</td>
<td>Disbursement D for insurance I is paid</td>
</tr>
<tr>
<td>T03 Check Purchase</td>
<td>Purchase for Disbursement D is checked</td>
</tr>
<tr>
<td>T04 Check Report of Offence</td>
<td>Report of Offence for Disbursement D is checked</td>
</tr>
</tbody>
</table>

As said, each bank contains one or more production facts that are not produced by the organization itself but by an organization in the environment of the system. The Fact Bank table specifies those production facts. The table contains two columns, one that specifies the bank, the other that specifies the production facts contained in it. Production facts are described in the same way as in the transaction result table. An example is depicted below.

Table 7-8: Example of a Fact Bank Table

<table>
<thead>
<tr>
<th>Bank</th>
<th>Production Fact</th>
</tr>
</thead>
<tbody>
<tr>
<td>B01</td>
<td>Product P has been sold</td>
</tr>
<tr>
<td>B02</td>
<td>Report of Offence R has been made</td>
</tr>
</tbody>
</table>
Proposed Way of working
There is no prescribed way of working.

C-Me Models
Figure 7-12 depicts the C-Me model of the DEMO Business Architecture Model. A core concept is the concept of a conversation. There are three types of conversations: Persuasion (persuade an actor to carry out a transaction), Authorization (ask an actor for authorization to perform a transaction) and Transaction (a conversation about a certain production action that has to be carried out by an actor). Each conversation has two actors: the initiating actor that starts the conversation and the responding actor that responds to the conversation. Another core concept introduced is the concept of a fact container. The transaction concept is an example of a fact container, since the carrying out of a transaction results in a fact, viz. the fact that the transaction is carried out. E.g. the transaction 'deliver goods' results e.g. in the fact that 'goods G are delivered for a customer C'. Also the bank concept is an example of a fact container. A bank contains facts that are created outside the organization that is modeled. Next to being initiator or responder in a conversation, actors can also inspect fact containers (transaction results and facts contained in banks) to influence their future actions in conversations.
Two types of actors are distinguished: aggregated actors and elementary actors. An elementary actor is uniquely assigned to be the responder of a transaction. An example of an aggregated actor is e.g. a travel organization. As part of this organization the elementary actor 'book trip' is responsible for carrying out the transaction 'book trip'. The customer is initiator of this transaction. Aggregated actors are split into two types: those being under consideration and those not being under consideration. The actors that are not under consideration are outside the organization that is currently modeled and not further filled in by elementary actors. The aggregated actor under consideration is the aggregated actor that is studied in more detail. The aggregated actor under consideration may consist of several elementary actors.

Figure 7-13: C-Me Model of the DEMO Process Phase Model

In the process phase model (Figure 7-13), again the concepts persuasion, authorization and transaction are modeled as conversations. A conversation has different phases. The transaction conversation has an order phase, execution phase and result phase. The persuasion and authorization conversations have a single phase. It can be the case that certain phases have to wait for other phases to finish. The 'conditional wait' knowledge primitive depicts this. E.g. the execution phase of a transaction has to wait until the order phase is finished. Conversations can be started during other phases of conversations. E.g. during the execution phase of a transaction a certain authorization conversation can be started. It is also possible for a conversation to be started externally, i.e. from outside the organization.

Figure 7-14 depicts the C-Me models of the DEMO Business Architecture Model, the DEMO Process Phase Model, the DEMO Transaction Result Table and the DEMO Bank Fact Table. Both Transaction Result Table and Bank Fact Table contain facts that in their turn refer to object types. E.g. the fact 'Goods G are delivered to Customer C' refers to the object types Good and Customer.
Figure 7-14: C-Me model of part of the DEMO Way of Modeling (1/3)

There are several coherency relationships between the aspect modeling techniques, most of them concerning equality relationships regarding transactions. CR1 indicates that both Transaction Result Table and Business Architecture Model should contain the same transactions. The same holds for the banks in CR2. CR3 is a coherency relationship of type OVERLAP. There can be overlap in the object types mentioned in both tables. CR4 specifies that the Transaction Result Table and Process Phase Model should contain the same transactions.

Figure 7-15 shows an additional equality coherency relationship between Process Phase Model and Business Architecture Model, expressing that each conversation in the Architecture Model should also be mentioned in the Phase Model and vice versa. Figure 7-16 shows a more complex coherency relationship between both aspect modeling techniques in order to maintain consistency between initiation of
conversations. For example, a conversation that is started by an aggregated actor that is not under consideration in the Architecture Model should be marked as initiated externally in the Phase Model. Furthermore, when a conversation A is started during a certain phase of another conversation B, then the responding actor of that conversation B should be equal to the initiating actor conversation A. The C-Me modeling language does not allow the description of the precise details of this coherency relationship.
Figure 7-15: C-Me Model of part of the DEMO Way of Modeling (2/3)
Figure 7-16: C-Me Model of part of the DEMO Way of Modeling (3/3)

Figure 7-17 shows the DEMO method. As can be seen there are no precedence relationships between the four modeling tasks, since DEMO does not prescribe the order in which model should be constructed.
Figure 7-17: C-Me Model of the DEMO Way of Working

See appendix A for a detailed description of the formalization.

7.4 Petri-net Based Modeling

The modeling technique discussed next is a hypothetical one, based on the main principles of the technique "Dynamic Enterprise Modeling" developed by the Baan Company, a Dutch vendor of Enterprise Resource Planning Systems (Post et al, 1996). It will be called Petri-net Based Modeling (PBM). The technique has been modified to be more suitable for the empirical study later on. The Entity Relationship modeling technique has been replaced with an Object Role Modeling oriented approach to make the comparison with the other techniques more interesting.

Table 7-9: PBM Business Control Model

<table>
<thead>
<tr>
<th>ASPECT MODEL TABLE: BUSINESS CONTROL MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Construct</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Business Function</td>
</tr>
</tbody>
</table>
The first aspect modeling technique discussed here is the business control model. It contains business functions (internal) and external agents (external). These concepts are related to each other by means of so-called request feedback loops. This concept is a pattern where one party requests something from the other. The other party gives feedback when the request has been carried out.

Figure 7-18 shows a typical example of a business control model, derived from Post et al (1996). Customer and Supplier are external agents whereas support, planning, purchasing, etc., are functions. 'Order to be fulfilled' is an example of a Request feedback loop between Customer and Sales.
Figure 7-19 shows how functions in the BCM can be decomposed into sub functions.

**Figure 7-19: Example PBM Business Control Model (2)**

Next to the Business Control Model, PBM offers Process Models. Each process model specifies the behavior of a request feedback loop in the Business Control Model. Process Models are based on the Petri-net principle that is explained briefly below.

Petri-nets originate from the work of Petri (1962). These classical, low-level, Petri-nets are defined by means of a set of transitions, places and arcs. In Table 7-10, transitions are represented by boxes, places are represented by circles and arcs by means of arrows. A place that is connected to a transition by means of a directed arc from place to transition is called an input place. An output place is a place that is connected to a transition by means of a directed arc from transition to place. Places can be filled with one or more tokens, depicted by black dots in Table 7-10. The way that tokens are divided over the places in a petri-net determines the current state of the net, called marking. A transition is able to fire when all its input places are filled with at least one token. Firing means that for all input places, a token is consumed (for all input places, the number of tokens is decreased by one) and for all output places, a token is created (increased by one). This principle is depicted in Table 7-10 (A). Classical petri-nets are expressive enough to model e.g. parallelism (B), iteration (C) and exclusiveness (D). The combination of places, transitions and arcs can also result in unsound petri-nets resulting in undesired multiplication of tokens (see E) or deadlock (see F). In order to model certain constructions such as iteration and parallelism, certain transitions that do not correspond to actual business tasks are necessary. These are called 'dummies'.

163
Table 7-10: Petri-Net construction examples

<table>
<thead>
<tr>
<th>How it works (A)</th>
<th>Parallelism (B)</th>
<th>Iteration (C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram A]</td>
<td>![Diagram B]</td>
<td>![Diagram C]</td>
</tr>
</tbody>
</table>

Exclusiveness (D) | WRONG! (E) | WRONG! (F) |

Petri-nets have been applied to model business processes, especially for the purpose of simulation and execution (Aalst, 2002). When modeling business processes, each transition stands for a task to be performed. Table 7-11 depicts the Petri-net variant that is evaluated in this thesis. It includes some extensions to low-level Petri-nets. Firstly, an explicit begin and end place are introduced, which makes this variant comparable to a so-called workflow net. Secondly, the possibility is offered to decompose transitions (called sub processes) into lower-level subnets (so-called hierarchy). No extensions with regard to color or time are discussed here.

Table 7-11: PBM Business Process Model

<table>
<thead>
<tr>
<th>ASPECT MODEL TABLE: BUSINESS PROCESS MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Begin Place</td>
</tr>
<tr>
<td><strong>ASPECT MODEL TABLE: BUSINESS PROCESS MODEL</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td><strong>End Place</strong></td>
</tr>
<tr>
<td>![Object] is [Action]</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| **Intermediate Place** | Intermediate state of the request feedback loop that is worked out |
| ![Object] is [Action] | When this place is preceded by a transition that performs an action on an object, the description is 'Object' is [Action]. Otherwise the description can be formed arbitrary |

| **Transition** | Transition from one state to another by performing an action on an object |
| ![Object] to be [Action] | Description in the form 'Object' to be [Action]' describing the action that is performed on an object |
|  | Sometimes 'dummy transitions' are necessary to construct sound Petri-nets. These Transitions do not have a description 'Object' to be [Action]' and do not represent actions |

| **Sub Process** | Transition from one state to another by starting a new request feedback loop to another function in the Business Control Diagram |
| ![Object] to be [Action] | Description in the form 'Object' to be [Action]' must be in accordance with the new request feedback loop that is started |
|  | Keep in mind consistency with the Business Control Model. There has to be a request feedback loop 'Object' to be [Action]' in the Business Control Model, between the function of the request feedback loop that is currently modeled and another function or external agent |

| **Input Place** | Transition or Sub Process B occurs when all of its input places (A) have at least one token |
| ![A] | A transition can have more than one input place |
| ![B] | Transitions as well as sub processes have input places |

| **Output Place** | The result of the occurrence of the transition A is that all of its output places (B) are filled (have tokens) |
| ![A] | A transition can have more than one output place |
| ![B] | Transitions as well as sub processes have output places |
Figure 7-20 shows a typical example of a business process model, specifying the behavior of the request feedback loop 'Order to be fulfilled' mentioned earlier in the Business Control Model (see begin and end places). 'Purchase order to be placed' and 'Invoice to be sent' are sub processes that start up the request feedback loops mentioned in the Business Control Model.

![Business Process Model Diagram]

Figure 7-20: Example PBM Business Process Model

**Business Information Model Table**

Table 7-12 shows the legend of the Business Information Model. It can be compared to Fact Modeling approaches such as NIAM (Nijssen, 1993), ORM (Halpin, 2001) and FCO-IM (Bakema et al, 1996).

**Table 7-12: PBM Business Information Model**

<table>
<thead>
<tr>
<th>Construct</th>
<th>Notation</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| Workflow Object Type     | name [identifier] | The type of objects that actions are performed on in the Business Process Model | - The identifier specifies how a workflow object can be uniquely identified. E.g., a car by its license plate, an order by means of an order number  
- Keep in mind the consistency with the Business Process Model and Business Control Model: All objects that appear in the descriptions 'Object to be [Action]' appear in the Business Information Model |
### ASPECT MODEL TABLE: BUSINESS INFORMATION MODEL

<table>
<thead>
<tr>
<th>Label Type</th>
<th>Denotes lexical objects such as strings and numbers</th>
<th>Mainly used to specify lexical properties of workflow object types in combination with a binary fact type, see example below</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fact Type</td>
<td>The type of relationships that can exist between workflow object types and label types</td>
<td>Each fact type has a number of roles. Roles are depicted by the number boxes 1..n. Each box is attached to one workflow object type or label type, see later on. Fact types with only one role are called 'unary fact types'. Fact types with two roles are called 'binary fact types' and so on. Instances of a fact type are facts. A fact refers to actual instances of object types and label types. Keep in mind the consistency with the Business Process Model and Business Control Model: All objects that appear in the descriptions '[Object] to be [Action]' have a unary fact type that allows the specification of the action being carried out, see example below.</td>
</tr>
<tr>
<td>Role Assignment</td>
<td>Assigns a workflow object type or label type to a role in a fact type</td>
<td>Only one object type or label type per role. Object types and label types can be assigned to one or more different roles within a single fact type as well as in other fact types.</td>
</tr>
<tr>
<td>Generalization</td>
<td>Specifies that B is a sub type of A.</td>
<td>B inherits the type of relationships that A is part of. The set of actual instances of the object type B is a subset of the set of instances of the object type A.</td>
</tr>
<tr>
<td>Unicity rule</td>
<td>Restriction rules on combinations of roles in a fact type</td>
<td>The combination of roles in a unicity rule restricts combinations of objects and labels that correspond to the object types assigned to the roles to occur more than once in the facts that are instances of this fact type, see example below.</td>
</tr>
</tbody>
</table>

Figure 7-21 shows an example application of the modeling technique. We see for instance the object type Order. By means of the binary fact types 'has customer' and 'has date' it is specified that an order belongs to one customer and that it has a unique date. Furthermore it is specified that an Order belongs 1:1 to a quotation. The same holds for the relationship between Order and Purchase Order. We also see 4 unary fact types that are connected to the Business Process Model. The fact 'Order is Fulfilled' is the fact that is the result of the order fulfillment, i.e. when the end state in the example depicted in 7-20 is reached. The fact 'Order is Entered' refers to the achievement of an intermediate state in this same example. The same holds for the other unary fact types. We see a clear relationship between Business
Information Model and Business Process Model that will be described as a coherency relationship in the C-Me models to come.

![Diagram of Information Model](image)

**Figure 7-21: Example Business Information Model**

**C-Me Models**

The first C-Me model discussed here is that of the Business Control Model (Figure 7-22). Core concept is that of an actor initiating and responding to request feedback loops. An actor can be an internal business function or an external agent (compare with the legend of the business control model described in Table 7-9). Functions can be decomposed into sub functions. This is expressed by the binary knowledge primitive 'is sub function of'.

![Diagram of C-Me Model](image)

**Figure 7-22: C-Me Model of the PBM Business Control Model**
Figure 7-23: C-Me Model of the PBM Business Process Model

Core concepts of the C-Me model of the Business Process Model (Figure 7-23) are not surprisingly the core concepts of the Petri-net modeling technique: the concept place and the concept transition. A place can be the input place or output place of a certain transition. Since a Process Model is drawn for each Request feedback loop in a Business Control Model, the Places and Transitions of a certain process model are part of the Request feedback loop that is specified in more detail. The 'part of' knowledge primitives in the C-Me model reflect this. On the other hand, certain transitions marked as sub processes in the process model stand for other request feedback loops that are carried out. This is expressed by the 'represents' knowledge primitive. An example of such a connection is the connection of e.g. sub process 'Purchase order to be placed' in Figure 7-20, to the Request feedback loop 'Purchase order to be placed' in Figure 7-19. Three types of places are distinguished: begin places, intermediate places and end places, each with different notations and different constraints.
Another set of concepts deals with the connection between process model and information model. As has been discussed before, the result of a transition in the process model corresponds to the creation of a fact in a fact model, compare e.g. the fact 'Order is Entered' in example information model (Figure 7-21) with the place 'Order is Entered' in the example process model (Figure 7-20). In these example facts and states, 'Order' is the Object Type and 'Entered' is the Action Type. In the C-Me model, Object Type and Action Type are combined to the concept 'Action on Object'. A transition performs such an 'Action on Object' and a Place corresponds to post condition of an 'Action on Object'. Later on it will be explained how these concepts are linked to the Information Model.

![Diagram](image)

**Figure 7-24: C-Me Model of the PBM Business Information Model**

Figure 7-24 shows the C-Me model of the Business Information Model. The structure consisting of the concept 'type', 'role' and 'fact type' is comparable to the structure 'entity type', 'role' and 'relationship type' of the EPC data view depicted earlier in Figure 7-6 and requires no further explanation. Furthermore, the information model distinguishes two kinds of types, the object type referring to objects in physical or social reality and the label type referring to lexical objects such as numbers and names. Between object types there is a generalization relationship. This generalization relationship is modeled in another way than the EPC generalization concept. It is modeled as a binary knowledge primitive, since no further attributes are assigned to the concept. Multiplicity is less expressive than in the EPC data view. It is modeled by a simple unicity constraint. With this constraint it can be expressed that a certain instantiation of the role that a type plays in a fact type can only occur once. E.g. an instantiation of the object type 'car' can only occur once in an instantiation of the fact type 'person owns car' since a car can only have one owner.
Figure 7-25: C-Me Model of the PBM Way of Modeling
Figure 7-25 shows the coherency relationship between the three PBM models. CR1 expresses that each request feedback loop in the business control model should be modeled by means of a process model. CR5 expresses that request feedback loops can also occur as sub processes in the process model in a consistent way. CR2 and CR3 express that respectively object types and action types in the business process model should also occur in the information model. CR4 expresses that these object types and action types should occur in the information model in a consistent way.

**Proposed Way of working**

The first step in the modeling process is to construct a business control model. After that, each request feedback loop is worked out in a business process model. After that, all workflow objects mentioned in the diagrams are worked out in the business information model.

![PBM Method Diagram](image)

**Figure 7-26: C-Me Model of the PBM Way of Working**

See appendix A for a detailed description of the formalization.

### 7.5 Business Modeling With UML

The Unified Modeling Language (UML) offers mechanisms, so-called 'extensions' that allow the extension of the language to make it suitable for the modeling of other domains. The UBM (Business Modeling with UML) is an example of such an extension. It has been proposed in 2000 by Eriksson and Penker in their book "Business Modeling with UML, Business Patterns at Work" (Eriksson et al, 2000). The standard UML is considered known. A gray background in first column of the aspect model tables below indicates that the described concept is an extension of the standard UML.
The Business Process Model is an extension of UML Activity Model.

### Table 7-13: UBM Business Process Model

<table>
<thead>
<tr>
<th>Construct</th>
<th>Notation</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin State</td>
<td><img src="image" alt="Begin State" /></td>
<td>Initial state of the process that is modeled</td>
<td>Each process model that contains control flows has exactly one begin state</td>
</tr>
<tr>
<td>End State</td>
<td><img src="image" alt="End State" /></td>
<td>Final state of the process that is modeled</td>
<td>Each process model that contains control flows has exactly one end state</td>
</tr>
<tr>
<td>Activity</td>
<td><img src="image" alt="Activity" /></td>
<td>Elementary Activity causing a state transition</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td><img src="image" alt="Process" /></td>
<td>An aggregated structure of activities.</td>
<td>For each process, a new process model must be drawn that &quot;zooms in&quot; on the elementary activities that comprise this process</td>
</tr>
<tr>
<td>Resource</td>
<td><img src="image" alt="Resource" /></td>
<td>Classes of things that are consumed or produced by processes or activities</td>
<td>Distinction is made between: Information resources, e.g. the information in documents, orders, production plans. Physical resources, e.g. products, machines. Abstract resources that cannot be touched upon, e.g. contracts, accounts. People, e.g. employees participating in the process. Mind the consistency: All resources appear in the conceptual model</td>
</tr>
<tr>
<td>Goal</td>
<td><img src="image" alt="Goal" /></td>
<td>Desired outcome of a process</td>
<td>Mind the consistency: Goals should also appear also in the goal model and might appear in the conceptual model</td>
</tr>
</tbody>
</table>
### ASPECT MODEL TABLE: BUSINESS PROCESS MODEL

<table>
<thead>
<tr>
<th>Control Flow</th>
<th>OR Split/Join</th>
<th>AND Split/Join</th>
<th>Object Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="diagram1" alt="Diagram" /></td>
<td><img src="diagram2" alt="Diagram" /></td>
<td><img src="diagram3" alt="Diagram" /></td>
<td><img src="diagram4" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **Control Flow**
  - The finishing of A is the condition for starting B
  - Control flows can be drawn between the concepts mentioned in the notation column: states, processes and activities.
  - A process or activity starts exactly one other process or activity.
  - Generally, a process or activity is started by exactly one other process or activity, except when an OR join is modeled, see example below.

- **OR Split/Join**
  - After A and before D, either B or C is carried out
  - Or Splits can also be utilized to create iterations.
  - B and C can be replaced more complex structures of activities.

- **AND Split/Join**
  - After A and before D, B and C are carried out in parallel
  - B and C can be replaced more complex structures of activities.

- **Object Flow**
  - Resource A is consumed by process or activity B
  - Resource B is produced by activity or process A
  - When B is a process and a separate process model is drawn that "zooms in" on B, then the input resources mentioned here should also appear in the process model that "zooms in" on B.
  - Processes or activities can consume many resources.
  - When A is a process and a separate process model is drawn that "zooms in" on A, then the output resources mentioned here should also appear in the process model that "zooms in" on A.
  - Processes or activities can produce many resources.
  - Each process "produces" one goal, meaning that the process achieves the goal.

The examples below show the two types of process models that are used in UBM. The High Level Process Model depicts processes, resources and goals without modeling the control flow. The low-level process models depict (for each process in
the high level process model) the individual activities, resources, processes, goals *including* control flow, begin state and end state.

![Diagram of high-level process model]

*Figure 7-27: Example UBM High-level Process Model*

![Diagram of low-level process model]

*Figure 7-28: Example UBM Low-level Process Model*

The Business Goal Model is an extension of the UML Object Model.
## Table 7-14: UBM Business Goal Model

<table>
<thead>
<tr>
<th>Construct</th>
<th>ASPECT MODEL TABLE: BUSINESS GOAL MODEL</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td><strong>&lt;Goals</strong>&lt;br&gt;<code>name:Quantitative Goal</code>&lt;br&gt;<code>Goal_Description = &quot;description&quot;</code>&lt;br&gt;<code>Goal_Value = value</code>&lt;br&gt;<code>Current_Value = value</code>&lt;br&gt;`Unit_of_measurement = &quot;unit&quot;**&lt;br&gt;The desired, quantitatively measurable outcome of a process</td>
<td>Goals also appear in the process model and might appear in the conceptual model. Fill in a &quot;description&quot; of the goal as well as a desired value and a current value, measured in a specified unit. See examples below. It is possible to draw additional goals that do not appear in the process model.</td>
</tr>
<tr>
<td><strong>Problem</strong></td>
<td><strong>&lt;Problem&gt;</strong>&lt;br&gt;<code>description</code>&lt;br&gt;Problem holding back the achievement of a goal or explaining the contradictoriness of goals</td>
<td>Goals also appear in the process model and might appear in the conceptual model. Fill in a description of the goal. It is possible to draw additional goals that do not appear in the process model.</td>
</tr>
<tr>
<td><strong>Goal Hierarchy</strong></td>
<td><img src="image" alt="Goal Hierarchy Diagram" /></td>
<td>A goal can have more than one sub goal. One sub goal has only one super goal.</td>
</tr>
<tr>
<td><strong>Goal Dependency</strong></td>
<td><strong>Goal A</strong>&lt;br&gt;<code>&lt;dependency&gt;</code>&lt;br&gt;<code>Goal B</code>&lt;br&gt;Goals A and B are contradictory. The achievement of A hinders the achievement of B and vice versa</td>
<td>Only binary goal dependencies are allowed.</td>
</tr>
<tr>
<td><strong>Problem Assignment</strong></td>
<td><strong>Goal A</strong>&lt;br&gt;<code>&lt;problem&gt;</code>&lt;br&gt;<code>Goal B</code>&lt;br&gt;A problem associated with the contradictoriness of goals A and B</td>
<td></td>
</tr>
<tr>
<td><strong>Problem Assignment</strong></td>
<td><strong>Goal A</strong>&lt;br&gt;<code>&lt;problem&gt;</code>&lt;br&gt;<code>Goal B</code>&lt;br&gt;Problem associated with achieving goal A</td>
<td></td>
</tr>
</tbody>
</table>
Figure 7-29 shows an example of a UBM Goal Model.

![UBM Goal Model Diagram](image)

**Figure 7-29: Example UBM Goal Model**

The Conceptual model is a simplification of the UML Class Model.

**Table 7-15: UBM Conceptual Model**

<table>
<thead>
<tr>
<th>Construct</th>
<th>ASPECT MODEL TABLE: CONCEPTUAL MODEL</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>name</td>
<td>Class of things relevant for the domain that is modeled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Mind the consistency between diagrams: all resources in the process model appear in classes in the conceptual model. Goals in the goal model and processes in the process model only appear in the conceptual model. It is possible to draw additional classes that do not appear as processes and goals in the other models.</td>
</tr>
<tr>
<td>Association</td>
<td>class A &lt;name&gt; -- class B</td>
<td>Relationship between class A and B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Arrows (e.g. '&lt;&gt;') depicts the direction in which the relationship name must be read. ca and cb are multiplicity descriptions common to UML. Examples are: &quot;*&quot; (many), '1' (exactly one), '0..1' (zero or one).</td>
</tr>
<tr>
<td>Generalization</td>
<td>class A</td>
<td>B and C are subclasses of A</td>
</tr>
<tr>
<td></td>
<td>class B class C</td>
<td>- B and C inherit the properties of A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The set of objects that are instances of classes B and C are subsets of the set of instances of A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A class can have more than one sub class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A sub class has only one super class</td>
</tr>
</tbody>
</table>
ASPECT MODEL TABLE: CONCEPTUAL MODEL

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Things belonging to the class A are built up out of things that belong to the class B and C</th>
</tr>
</thead>
</table>

A class (e.g. A) can have more than one part class (e.g. B and C) One part class (e.g. B and C) can also belong to one or more whole classes (e.g. A)

The figure below shows an example of a UBM conceptual model.

![Figure 7-30: Example UBM Conceptual Model](image)

**Proposed Way of working**

The first step in the modeling process is to construct a high level process model. After that, each of the processes is worked out by means of a low level process model, keeping the resources used consistent. After that, a conceptual model is constructed.

**C-Me Models**

The C-Me model of the UBM High Level Process Model (Figure 7-31) shows the four ways that processes can make use of resources: supply, control, consume and produce. There are four kinds of resources distinguished: abstract, people, physical and information. Each process has a goal assigned to it.
Figure 7-31: C-Me Model of the UBM High Level Process Model

Figure 7-32: C-Me Model of the UBM Low Level Process Model
The UBM Low level process model (Figure 7-32) is an extension of the high level process model. In this model, low level activities supply, control, consume and produce resources. An activity can be an elementary activity or a process. An activity is a special kind of connector, together with state, join and split. Between these connectors precedence relationships exist. End state and begin state are special kinds of states. And Join and Or Join are special kinds of joins, the same holds for the concept split. Connectors are always part of a certain process.

Figure 7-33: C-Me Model of the UBM Goal Model

Core concept of the UBM Goal model (Figure 7-33) is the concept Goal. There are two kinds of goals: quantitative and qualitative goals. For quantitative goals, a current and target value can be specified. There are two relationships between goals: one goal can be a sub goal of another goal and a goal might be dependent on another goal. This latter relationship is the concept 'Goal Dependency'. Problems can concern goal dependencies and the achievement of goals.

Figure 7-34: C-Me Model of the UBM Conceptual Model
The UML conceptual model (Figure 7-34) is comparable to the EPC data view. The structure 'class', 'role', 'association' and 'multiplicity' is comparable to the EPC structure 'entity type', 'role', 'association type' and 'Multiplicity value'. The specialization relationship is comparable to the PBM generalization relationship. Furthermore there is a part-of or aggregation relationship between classes.

![UBM Method Diagram](image)

**Figure 7-35: C-Me Model of the UBM Way of Working**

Figure 7-35 shows the way of working proposed. Firstly a high level process model is drawn. Out of this process model, a goal model as well as a number of low level process models are derived and further specified. After these models are constructed, the conceptual model is derived and further specified.

The rather complex structure of coherency relationships within UBM is depicted in Figure 7-36, Figure 7-37 and Figure 7-38. Many of these relationships concern the UBM concept 'class' which can be considered the core concept of UBM. CR1 specifies that the resources mentioned in the high level process model must be a subset of the resources mentioned in the low level process model. When the way of working is considered, this means that the modeler starts by specifying resources at a high level in the high level process model and might add more specific resources later on in the low level process model. CR2 specifies that the way that these resources are consumed by processes and activities must be consistent throughout the high and low level process models. In low level process models it is allowed to refer to other high level processes. In the case this is done, CR3 guarantees that the use of resources is modeled in a consistent way. CR4 specifies that the goals mentioned in the high level process model should also appear in the goal model.
Figure 7-36: C-Me Model of UBM Model Coherence (1/3)
Figure 7-37: C-Me Model of the UBM Model Coherence (2/3)
Figure 7-37 depicts the coherency relationships concerning the concept 'class'. CR5 specifies that goals in the goal model might appear as classes in the conceptual model. CR6, 7 and 8 specify that respectively processes, resources and goals in the high level process model might also appear as classes in the conceptual model. CR9 and CR13 specify that respectively resources and processes in the low level process model might appear in the conceptual model.

Figure 7-38 depicts more complex coherency relationships between the Conceptual Model and the other aspect models. CR10 specifies that goal dependencies in the goal model might be specified as associations between two classes. These two classes must then refer to the two goals in the goal model between which the dependency exists. CR11 and 12 specify that the use of a resource by a process in respectively the low level and high level process model can also be modeled as an association in the Conceptual Model.

It must be noted that the coherency relationships CR5 up until CR13 mentioned above are of the type OVERLAP, i.e. concepts on the one hand might overlap with concepts on the other hand of a coherency relationship and vice versa. Compared to the SUBSET and EQUAL coherency relationship, forcing concepts on the one hand of the relationship also to appear at the other hand of the relationship, the OVERLAP relationship is not restricting the modeler. So, although there are many coherency relationship concerning the Conceptual Model, no errors with regard to inter model consistency or well-formedness are to be expected. See appendix A for a detailed description of the formalization.
Figure 7-38: C-Me Model of the UBM Model Coherence (3/3)
7.6 Conclusions

This chapter is the first chapter in three chapters concerning the application of the research proposition. In this chapter I have successfully applied the C-Me modeling language to capture four state of the art business process modeling techniques.

The application of the C-Me modeling language on EPC’s, DEMO, PBM and UBM shows that the C-Me modeling language is capable of modeling state of the art multi modeling techniques and their interrelationships.

C-Me models where constructed on the basis of a description of the concepts used in a modeling technique. The application shows that in general, the C-Me language is suitable for capturing multi modeling techniques. It is a valuable addition to existing meta modeling techniques that are in general not capable of modeling different aspect modeling techniques and their interrelationships. The application also shows that the formalization of the C-Me models is sound and expressive enough to capture the techniques (see appendix A).

Also the ability to model the construction of aspect models as activities that are part of the way of working concerning the application of a multi modeling technique is a valuable addition to existing meta modeling approaches that do not take this aspect into consideration.

Two modeling issues require attention. The first one is that requires attention is the criterion to decide whether a modeling construct in a technique must be classified as a C-Me concept or a C-Me knowledge primitive. As an example serves the concept of generalization, which was classified as a C-Me concept when the EPC data view was captured and on the other hand as a C-Me knowledge primitive when the UBM conceptual model was captured. The reason for this distinction is that in the EPC data view extra attributes (disjunct and overlapping) were added to the concept. In the UBM conceptual model it was modeled as a knowledge primitive since there were no extra attributes to be assigned to it.

Another modeling issue that I want to discuss here is the coherency relationship. In general, the modeling of coherency relationship must be viewed as a valuable addition to existing meta modeling techniques. In this thesis I have introduced the means to model aspect modeling techniques and their interrelationships. Simple coherency relationships, consisting of one concept at the one hand and another concept at the other hand are well defined by means of the types OVERLAP, SUBSET and EQUAL and form a good basis for keeping the models that are constructed with the technique well-formed and to identify mall-formed models. However, more complex coherency relationships, concerning a structure of concepts and knowledge primitives lack a further formal specification of what exactly the nature of the coherency relationship is. In this chapter I have described this nature
textually when the coherency relationships were explained. The lack of a formal description of the nature of these coherency relationships can also distort the identification of coherency relationships. One should be careful to identify elementary coherency relationships, relationships that concern one single consistency or well-formedness rule that has to be obeyed. The danger of the lack of further specification is that one arbitrarily chooses a large structure of concepts and knowledge primitives on the one hand of the coherency relationship and a large structure on the other hand and than textually explains that there is 'some' coherency between the two.
APPLICATION AND VALIDATION OF THE Q-ME MEASUREMENT SCHEME

8.1 Introduction

In the previous chapters I have proposed a number of properties of modeling techniques as well as means to measure them. Furthermore, I have posed a number of assumptions with regard to the influence of these properties on the quality of the resulting models. In this chapter I will apply the proposed measures on the four modeling techniques that were described in the previous chapter.

Throughout this thesis, a difference has been made between the correctness of business models and the usefulness of business models. With regard to the influence of techniques on the correctness of models, techniques should reduce the ability of the modelers making errors. A number of formal metrics have been proposed presumed indicative for the number of errors made in models. The metrics will be applied in section 8.2. It has been argued that especially the modeling concepts offered by techniques determine the usefulness of a resulting model. In order to investigate the suitability of the concepts offered by techniques, inquiring metrics were developed in section 6.4. These metrics will be applied in section 8.3.

In order to study the predictive value of the theoretical measurements that are obtained, they will be compared to empirical data that was collected in sessions where 30 modelers applied the four techniques on a case description. In section 8.4, it is investigated whether the a priori theoretical measures have a predictive value with regard to the e.g. kind and amount of errors made by the modelers in these sessions. This chapter concludes the part of the thesis that dealt with the application and validation of the proposed framework.
8.2 Application of the Metrics Assuring Correctness

Prerequisites for understanding quality
The prerequisite for understanding empirical quality was met by providing the notation of all modeling concepts and knowledge primitives of the discussed modeling techniques. The prerequisite for understanding syntactical quality was met by constructing the meta models of the four techniques by using the C-Me language. The prerequisite for understanding semantic quality was met by presenting a textual description of the meaning of each of the concepts and knowledge primitives in the meta model in the model tables in chapter 7. A more formal description is given when the system expressiveness is discussed and concepts are mapped to the discrete dynamic systems paradigm (section 8.3).

Notational intuitiveness
In order to measure the Notational intuitiveness of both aspect modeling techniques and multi modeling techniques, the AMI and WMI inquiring metrics have been proposed. Figure 8-1 shows an example application of the questionnaire belonging to the inquiring metrics applied for the four multi modeling techniques and their constituting aspect modeling techniques.
**Metric Name:** AME, WME; **Metric Type:** Inquiring

**Question**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Are the symbols used for different generic concepts well distinguishable (e.g. the notation for a UML Actor compared to UML Use Case)?</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>NA</td>
<td>NA</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2 Is the notation intuitive, i.e. in line with current and past practice?</td>
<td>1/2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1/2</td>
<td>NA</td>
<td>NA</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1/2</td>
<td>+</td>
<td>1/2</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3 Are the symbols that are used more complex than standard shapes such as rectangles, circles, ovals, diamonds, et cetera?</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1/2</td>
<td>NA</td>
<td>NA</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1/2</td>
<td>NA</td>
<td>NA</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1/2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For the complete multi modeling technique</th>
<th>EPC</th>
<th>DEMO</th>
<th>PBM</th>
<th>UBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Is the notation of a concept throughout different aspect modeling techniques used in a consistent way? (e.g. a UML object in an object diagram and a UML object in a collaboration diagram)?</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Figure 8-1: Example Application of the AMI and WMI inquiry**

Once again, it should be said that the application of the enquiry here is meant as an example, it has not been carried out amongst a representative number of experts. Summarized, the results for the AMI metric, based on this example application, are as follows:

**Table 8-1: AMI Measures**

<table>
<thead>
<tr>
<th>AMI&lt;sub&gt;es&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = .7</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL = .6</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = NA</td>
</tr>
<tr>
<td>BANKFACTTABLE = NA</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = .9</td>
</tr>
</tbody>
</table>
Rather high values are scored by e.g. the UBM aspect modeling techniques due to the fact that their notation is well distinguishable, in line with accepted symbols and making use of easy to draw symbols. Low scores are measured for the DEMO process phase diagram and the PBM Information Model, due to the complex notation of some of the symbols in this diagram. Overall, the multi modeling techniques score as follows:

Table 8-2: WMI Measures

<table>
<thead>
<tr>
<th>WMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT DRIVEN PROCESS CHAINS = .85</td>
</tr>
<tr>
<td>DYNAMIC ESSENTIAL BUSINESS MODELING = .7375</td>
</tr>
<tr>
<td>PETRI-NET BASED MODELING = .75</td>
</tr>
<tr>
<td>BUSINESS MODELING WITH UML = 0.8188</td>
</tr>
</tbody>
</table>

**Syntactic Freedom**

A number of metrics have been proposed to measure the syntactic freedom of aspect modeling techniques. The calculation is carried out by applying the measurement schemes defined in section 6.3 on the formal definitions of the techniques described in the previous chapter.

Aspect Model Syntactic Freedom (ASF) is the average value of the syntactic freedom of all knowledge primitives in an aspect model (the KSF metric). Let us start by giving some typical examples of KSF values that turned out high.

For example, three of the four base concepts in the process phase model (persuasion, authorization, and transaction, see Figure 7-14) can appear in the 'initiated externally' knowledge primitive. One of the base concepts (phase, Figure 7-14) cannot. Therefore the KSF(initiated externally) is rather high: $3/4 = 0.75$. The same holds for the 'precedes' knowledge primitives in the UBM (Figure 7-32) and the EPC(Figure 7-7). For the UBM, the KSF(Precedes) $= 0.66$ since 8 out of 12 base concepts can play the roles of this knowledge primitives. In EPC, the KSF(Precedes) is even 0.72. In the EPC Function View, the KSF(issubfunctionof)=1 since there is only one concept that plays a role in this knowledge primitive, and since this
primitive is the only knowledge primitive in this aspect model, the ASF(functionview)=1, see below.

<table>
<thead>
<tr>
<th>ASF_{a,s}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = .3428572</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL = .5</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = .3333333</td>
</tr>
<tr>
<td>BANKFACTTABLE = .3333333</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = .44</td>
</tr>
<tr>
<td>PROCESSMODELLLOW = .3402778</td>
</tr>
<tr>
<td>GOALMODEL = .2833333</td>
</tr>
<tr>
<td>CONCEPTUALMODEL = .25</td>
</tr>
<tr>
<td>PROCESSVIEW = .2181818</td>
</tr>
<tr>
<td>FUNCTIONVIEW = 1</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW = .5</td>
</tr>
<tr>
<td>DATAVIEW = .2</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL = .4444444</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL = .1875</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL = .225</td>
</tr>
</tbody>
</table>

Based on these values, it is presumable that modelers will make less syntactical mistakes in e.g. the EPC function view than in the EPC process view, since the syntactic freedom in the function view is maximal (1) and in the process view quite low (0.22). From the AUC measurements depicted below we can conclude that most uniqueness constraints can be found in the UBM conceptual model modeling technique. 83% of the knowledge primitives in this modeling technique has at least a uniqueness constraint. The DEMO bank fact table scores lowest, since a bank can contain more than one fact and a fact can be contained in more than one bank there are no constraints.

<table>
<thead>
<tr>
<th>AUC_{a,s}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = .4</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL = .5</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = .5</td>
</tr>
<tr>
<td>BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = .2</td>
</tr>
<tr>
<td>PROCESSMODELLLOW = .1666667</td>
</tr>
<tr>
<td>GOALMODEL = .5</td>
</tr>
<tr>
<td>CONCEPTUALMODEL = .8333333</td>
</tr>
<tr>
<td>PROCESSVIEW = .4</td>
</tr>
<tr>
<td>FUNCTIONVIEW = 1</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW = 1</td>
</tr>
<tr>
<td>DATAVIEW = .625</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL = 1</td>
</tr>
</tbody>
</table>
Presumed to be influencing the well-formedness of models is the number of knowledge primitives in an aspect modeling technique that have a requiredness constraint. Below we see that many techniques have a maximum score here, for example the DEMO transaction result table. Every transaction is required to have a resulting fact specified and every result should at least reference to one object type.

Table 8-5: ARQ Measures

<table>
<thead>
<tr>
<th>ARQ&lt;sub&gt;ij&lt;/sub&gt;</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL</td>
<td>.8</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL</td>
<td>.25</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>BANKFACTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSMODELHIGH</td>
<td>.2</td>
</tr>
<tr>
<td>PROCESSMODELLOW</td>
<td>0</td>
</tr>
<tr>
<td>GOALMODEL</td>
<td>0</td>
</tr>
<tr>
<td>CONCEPTUALMODEL</td>
<td>.6666667</td>
</tr>
<tr>
<td>PROCESSVIEW</td>
<td>.8</td>
</tr>
<tr>
<td>FUNCTIONVIEW</td>
<td>0</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW</td>
<td>.5</td>
</tr>
<tr>
<td>DATAVIEW</td>
<td>.75</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL</td>
<td>.6666667</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL</td>
<td>1</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL</td>
<td>.5</td>
</tr>
</tbody>
</table>

Coherency

The AMC metric simply measures the number of coherency relationships between the different aspect modeling techniques in a multi modeling technique. Remarkable is the high values for the AMC's where the UBM conceptual model is involved. This is due to the fact that many concepts in the other UBM aspect models reappear in the conceptual model.

Table 8-6: AMC Measures

<table>
<thead>
<tr>
<th>AMC&lt;sub&gt;ij&lt;/sub&gt;</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL-PROCESSPHASEMODEL</td>
<td>2</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-TRANSACTIONRESULTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-BANKFACTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-TRANSACTIONRESULTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-BANKFACTTABLE</td>
<td>0</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE-BANKFACTTABLE</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-PROCESSMODELLOW</td>
<td>2</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-GOALMODEL</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-CONCEPTUALMODEL</td>
<td>4</td>
</tr>
</tbody>
</table>
Inter model inconsistencies in resulting models occur especially in those coherency relationships that contain knowledge primitives with uniqueness constraints on them. The AMCUC simply counts those coherency relationships. Here we see that all evaluated modeling techniques score quite well. Combinations of techniques with a score>0 increase the possibility that the resulting model cycle is inter model inconsistent.

Table 8-7: AMCUC Measures

<table>
<thead>
<tr>
<th>AMCUC_{i,j,e8}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL-PROCESSPHASEMODEL = 1</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-TRANSACTIONRESULTTABLE = 0</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-TRANSACTIONRESULTTABLE = 0</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-PROCESSMODELLOW = 0</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-GOALMODEL = 0</td>
</tr>
<tr>
<td>PROCESSMODELHIGH-CONCEPTUALMODEL = 1</td>
</tr>
<tr>
<td>PROCESSMODELLOW-GOALMODEL = 0</td>
</tr>
<tr>
<td>PROCESSMODELLOW-CONCEPTUALMODEL = 1</td>
</tr>
<tr>
<td>GOALMODEL-CONCEPTUALMODEL = 1</td>
</tr>
<tr>
<td>PROCESSVIEW-FUNCTIONVIEW = 0</td>
</tr>
<tr>
<td>PROCESSVIEW-ORGANIZATIONALVIEW = 0</td>
</tr>
<tr>
<td>PROCESSVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td>FUNCTIONVIEW-ORGANIZATIONALVIEW = 0</td>
</tr>
<tr>
<td>FUNCTIONVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL-BUSINESSPROCESSMODEL = 1</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL-BUSINESSINFORMATIONMODEL = 0</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL-BUSINESSINFORMATIONMODEL = 1</td>
</tr>
</tbody>
</table>

Summarizing this for the complete multi modeling technique, the WMCUC values are as follows:
Table 8-8: WMCUC Measures

<table>
<thead>
<tr>
<th>WMCUC_{i,j}^{S}</th>
<th>EVENT DRIVEN PROCESS CHAINS = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DYNAMIC ESSENTIAL BUSINESS MODELING = .1666666</td>
</tr>
<tr>
<td></td>
<td>PETRI-NET BASED MODELING = .6666666</td>
</tr>
<tr>
<td></td>
<td>BUSINESS MODELING WITH UML = 0.5</td>
</tr>
</tbody>
</table>

Inter model mal-formedness in resulting models occurs especially when a concept mentioned in one aspect model should also appear in another model (subset relationship) and possibly also vise versa (equality relationship). In total there are three types of coherency relationships: subset, equality and overlapping. The latter one is the least restrictive and has no consequences for the mal-formedness of resulting models. Therefore, the number of coherency relationships that are not overlapping (NO) is presumed to be a helpful measure for predicting consistency problems. Based on these measures problems can be expected with regard to the consistency of statements in the DEMO architecture models and process phase models, the UBM process model high and UBM process model low, the PBM Business Control Model and the Business Process Model as well as the Business Process Model and the Business Information Model.

Table 8-9: AMCNO Measures

<table>
<thead>
<tr>
<th>AMCNO_{i,j}^{S}</th>
<th>BUSINESSARCHITECTUREMODELP-PROCESSPHASEMODEL = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BUSINESSARCHITECTUREMODEL-TRANSACTIONRESULTTABLE = 1</td>
</tr>
<tr>
<td></td>
<td>BUSINESSARCHITECTUREMODEL-BANKFACTTABL = 1</td>
</tr>
<tr>
<td></td>
<td>PROCESSPHASEMODEL-TRANSACTIONRESULTTABLE = 1</td>
</tr>
<tr>
<td></td>
<td>PROCESSPHASEMODEL-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td></td>
<td>TRANSACTIONRESULTTABLE-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td></td>
<td>PROCESSMODELMEDIUM-PROCESSMODELLOW = 2</td>
</tr>
<tr>
<td></td>
<td>PROCESSMODELMEDIUM-GOALMODEL = 1</td>
</tr>
<tr>
<td></td>
<td>PROCESSMODELMEDIUM-CONCEPTUALMODEL = 0</td>
</tr>
<tr>
<td></td>
<td>PROCESSMODELLLOW-GOALMODEL = 0</td>
</tr>
<tr>
<td></td>
<td>PROCESSMODELLLOW-CONCEPTUALMODEL = 0</td>
</tr>
<tr>
<td></td>
<td>GOALMODEL-CONCEPTUALMODEL = 0</td>
</tr>
<tr>
<td></td>
<td>PROCESSVIEW-FUNCTIONVIEW = 1</td>
</tr>
<tr>
<td></td>
<td>PROCESSVIEW-ORGANIZATIONALVIEW = 1</td>
</tr>
<tr>
<td></td>
<td>PROCESSVIEW-DATAVIEW = 1</td>
</tr>
<tr>
<td></td>
<td>FUNCTIONVIEW-ORGANIZATIONALVIEW = 0</td>
</tr>
<tr>
<td></td>
<td>FUNCTIONVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td></td>
<td>ORGANIZATIONALVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td></td>
<td>BUSINESSCONTROLMODEL-BUSINESSPROCESSMODEL = 2</td>
</tr>
<tr>
<td></td>
<td>BUSINESSCONTROLMODEL-BUSINESSINFORMATIONMODEL = 0</td>
</tr>
<tr>
<td></td>
<td>BUSINESSPROCESSMODEL-BUSINESSINFORMATIONMODEL = 3</td>
</tr>
</tbody>
</table>

Summarizing this for the complete multi modeling technique, the WMCNO values are as follows:
Table 8-10: WMCNO Measures

<table>
<thead>
<tr>
<th>WMCNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT DRIVEN PROCESS CHAINS = 0.5</td>
</tr>
<tr>
<td>DYNAMIC ESSENTIAL BUSINESS MODELING = 0.666666</td>
</tr>
<tr>
<td>PETRI-NET BASED MODELING = 0.666666</td>
</tr>
<tr>
<td>BUSINESS MODELING WITH UML = 0.333333</td>
</tr>
</tbody>
</table>

When the number of concepts and knowledge primitives that participate in these coherency relationships is counted, we see for example that 54% of the concepts and knowledge primitives in the UBM Process Model High and Conceptual Model appear in a Coherency Relationship.

Table 8-11: ACC Measures

<table>
<thead>
<tr>
<th>ACC_{ijes}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL-PROCESSPHASEMODEL = .4</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-TRANSACTIONRESULTTABLE = .0952381</td>
</tr>
<tr>
<td>BUSINESSARCHITECTUREMODEL-BANKFACTTABLE = .0952381</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-TRANSACTIONRESULTTABLE = .1428571</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL-BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE-BANKFACTTABLE = .2</td>
</tr>
<tr>
<td>PROCESSMODEL-HIGH-PROCESSMODELLOW = .4444444</td>
</tr>
<tr>
<td>PROCESSMODEL-HIGH-GOALMODEL = 8.333334E-02</td>
</tr>
<tr>
<td>PROCESSMODEL-HIGH-CONCEPTUALMODEL = .5454546</td>
</tr>
<tr>
<td>PROCESSMODELLOW-GOALMODEL = 0</td>
</tr>
<tr>
<td>PROCESSMODELLOW-CONCEPTUALMODEL = .3529412</td>
</tr>
<tr>
<td>GOALMODEL-CONCEPTUALMODEL = .3181818</td>
</tr>
<tr>
<td>PROCESSVIEW-FUNCTIONVIEW = .0952381</td>
</tr>
<tr>
<td>PROCESSVIEW-ORGANIZATIONALVIEW = 8.695652E-02</td>
</tr>
<tr>
<td>PROCESSVIEW-DATAVIEW = .0625</td>
</tr>
<tr>
<td>FUNCTIONVIEW-ORGANIZATIONALVIEW = 0</td>
</tr>
<tr>
<td>FUNCTIONVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW-DATAVIEW = 0</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL-BUSINESSPROCESSMODEL = .4</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL-BUSINESSINFORMATIONMODEL = 0</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL-BUSINESSINFORMATIONMODEL = .3214286</td>
</tr>
</tbody>
</table>

The AMC, AMCUC and AMCNO metrics mentioned above, measure coherency relationships between two aspect modeling techniques $i, j \in S$. The next four metrics measure the average coherency relationships of one aspect modeling technique $s \in S$ with the other aspect modeling techniques. AMCII_{es} counts each aspect modeling technique that has at least a coherency relationship with $i$ and divides this by the $|S|-1$ (the number of aspect modeling techniques that $s$ can have coherency relationships with). Likewise, AMCI_{es} counts each aspect modeling technique that has more than one coherency relationship with $s \in S$ and divides this...
by the \(|S|\)-1. AMCIUC and AMCINO are respectively the averages of AMCUC and AMCNO for one aspect modeling technique \(s \in S\). The results are as follows.

### Table 8-12: AMCI0 Measures

<table>
<thead>
<tr>
<th>AMCI0(_{s\in S})</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = 1</td>
</tr>
<tr>
<td>PROCPHASEMODEL = .6666667</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = 1</td>
</tr>
<tr>
<td>BANKFACTTABLE = .6666667</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = 1</td>
</tr>
<tr>
<td>PROCESSMODELLLOW = .6666667</td>
</tr>
<tr>
<td>GOALMODEL = .6666667</td>
</tr>
<tr>
<td>CONCEPTUALMODEL = 1</td>
</tr>
<tr>
<td>PROCESSVIEW = 1</td>
</tr>
<tr>
<td>FUNCTIONVIEW = .3333333</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW = .3333333</td>
</tr>
<tr>
<td>DATAVIEW = .3333333</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL = .5</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL = 1</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL = .5</td>
</tr>
</tbody>
</table>

### Table 8-13: AMCI1 Measures

<table>
<thead>
<tr>
<th>AMCI1(_{s\in S})</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = .3333333</td>
</tr>
<tr>
<td>PROCPHASEMODEL = .3333333</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = 0</td>
</tr>
<tr>
<td>BANKFACTTABLE = 0</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = .6666667</td>
</tr>
<tr>
<td>PROCESSMODELLLOW = .6666667</td>
</tr>
<tr>
<td>GOALMODEL = .3333333</td>
</tr>
<tr>
<td>CONCEPTUALMODEL = 1</td>
</tr>
<tr>
<td>PROCESSVIEW = 0</td>
</tr>
<tr>
<td>FUNCTIONVIEW = 0</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW = 0</td>
</tr>
<tr>
<td>DATAVIEW = 0</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL = .5</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL = 1</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL = .5</td>
</tr>
</tbody>
</table>

### Table 8-14: AMCIUC Measures

<table>
<thead>
<tr>
<th>AMCIUC(_{s\in S})</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = .3333333</td>
</tr>
<tr>
<td>PROCPHASEMODEL = .3333333</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = 0</td>
</tr>
<tr>
<td>BANKFACTTABLE = 0</td>
</tr>
</tbody>
</table>
Table 8-15: AMCINO Measures

<table>
<thead>
<tr>
<th>AMCINO</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL</td>
<td>1</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL</td>
<td>.6666667</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE</td>
<td>.6666667</td>
</tr>
<tr>
<td>BANKFACTTABLE</td>
<td>.3333333</td>
</tr>
<tr>
<td>PROCESSMODELHIGH</td>
<td>.6666667</td>
</tr>
<tr>
<td>PROCESSMODELLOW</td>
<td>.3333333</td>
</tr>
<tr>
<td>GOALMODEL</td>
<td>.3333333</td>
</tr>
<tr>
<td>CONCEPTUALMODEL</td>
<td>0</td>
</tr>
<tr>
<td>PROCESSVIEW</td>
<td>1</td>
</tr>
<tr>
<td>FUNCTIONVIEW</td>
<td>.3333333</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW</td>
<td>.3333333</td>
</tr>
<tr>
<td>DATAVIEW</td>
<td>.3333333</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL</td>
<td>.5</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL</td>
<td>1</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL</td>
<td>.5</td>
</tr>
</tbody>
</table>

Determinism

As a measure for determinism, the experimental metric that was proposed before has been carried out. 30 graduate students have participated in this experiment by modeling a library case, discussed in more detail in section 8.4. Variance is the degree to which aspect models made by a number of different modelers with a single aspect modeling technique on a single case differ from each other. The variance between each of models made by the modelers and a median solution was determined. For each of the modeling techniques such a median solution was determined by comparing all resulting models and determining which of these models varied the least from all others. Figure 8-2 shows an example to illustrate the procedure. In order to measure the variance of the PBM Business Information Model, an analysis has been made of the 'Object Type' modeling concept, which is one of the core concepts in the PBM Information Model. Each object type mentioned by a subject is listed. We see for example that 92% of the subjects mentioned the object type 'Book' in their Business Information Model. In this example, the median
solution (the solution varying the least with all other solutions) was mentioned by five subjects, the subjects T1, S1, D2, C2 and E2. Their Information Model contains the object types 'Book', 'Person' and 'Fine'. One could argue whether such a solution is a useful model of the information in the library case but that is not what is measured here.

Figure 8-2: Example results semantic quality PBM Information Model

Furthermore we see that e.g. subject V1 has a 0.25 variance with the median solution. Since only 1 of the 4 object types given (the object type 'card') differs from the median solution. An important overall measure is the average of all variances. In the case of the example, the variance between all the constructed PBM Business Information Models is 0.42. If each of the 30 subjects had constructed the same Business Information Model belonging to this case, the average variance would be 0.00.
Due to the intensiveness of this procedure it was not feasible to measure the CSV for all concepts and then to determine the ASV for all aspect models. What has been done is to measure some CSV for core concepts in an aspect model and to let this value be indicative for the aspect model. Below, the results of these measurements are depicted.

Table 8-16: CSV values for several modeling concepts

<table>
<thead>
<tr>
<th>Technique</th>
<th>Aspect Model</th>
<th>Concept</th>
<th>CSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPC</td>
<td>Process View</td>
<td>Function</td>
<td>0.63</td>
</tr>
<tr>
<td>PBM</td>
<td>Business Process Model</td>
<td>Transition</td>
<td>0.61</td>
</tr>
<tr>
<td>UBM</td>
<td>Low level Process Model</td>
<td>Activity</td>
<td>0.60</td>
</tr>
<tr>
<td>DEMO</td>
<td>Business Architecture</td>
<td>Transaction</td>
<td>0.36</td>
</tr>
<tr>
<td>PBM</td>
<td>Business Control Model</td>
<td>RF Loop</td>
<td>0.47</td>
</tr>
<tr>
<td>EPC</td>
<td>Data View</td>
<td>Entity Type</td>
<td>0.29</td>
</tr>
<tr>
<td>DEMO</td>
<td>Transaction Result Table</td>
<td>Object Type</td>
<td>0.46</td>
</tr>
<tr>
<td>PBM</td>
<td>Business Information Model</td>
<td>Object Type</td>
<td>0.42(*)</td>
</tr>
<tr>
<td>UBM</td>
<td>Conceptual Model</td>
<td>Class</td>
<td>0.62</td>
</tr>
</tbody>
</table>

(*) See example in Figure 8-2

Complexity

Finally, the AMS measure is calculated for each of the aspect modeling techniques as a measure for complexity. The AMS metric simply counts the number of concepts and knowledge primitives in an aspect modeling technique.

Table 8-17: AMS Measures

<table>
<thead>
<tr>
<th>AMS&lt;sub&gt;acs&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESSARCHITECTUREMODEL = 12</td>
</tr>
<tr>
<td>PROCESSPHASEMODEL = 8</td>
</tr>
<tr>
<td>TRANSACTIONRESULTTABLE = 5</td>
</tr>
<tr>
<td>BANKFACTTABLE = 5</td>
</tr>
<tr>
<td>PROCESSMODELHIGH = 10</td>
</tr>
<tr>
<td>PROCESSMODELLOW = 18</td>
</tr>
<tr>
<td>GOALMODEL = 11</td>
</tr>
<tr>
<td>CONCEPTUALMODEL = 10</td>
</tr>
<tr>
<td>PROCESSVIEW = 16</td>
</tr>
<tr>
<td>FUNCTIONVIEW = 2</td>
</tr>
<tr>
<td>ORGANIZATIONALVIEW = 4</td>
</tr>
<tr>
<td>DATAVIEW = 13</td>
</tr>
<tr>
<td>BUSINESSCONTROLMODEL = 6</td>
</tr>
<tr>
<td>BUSINESSPROCESSMODEL = 16</td>
</tr>
<tr>
<td>BUSINESSINFORMATIONMODEL = 9</td>
</tr>
</tbody>
</table>

A conclusion that can be drawn from applying the AMS metric is that EPC technique is rather unbalanced with regard to the distribution of concepts over aspect modeling techniques. This is especially interesting when also the modeling
process is considered: One starts with the EPC Process view, introduction 16 new concepts and knowledge primitives (AMS = 16) and after that in random order the Functional view (AMS=2), Organizational view (AMS=4) and Data view (AMS=12) are carried out. As can be seen, the Functional View and Organizational view add little new knowledge to the whole model.

8.3 Application of the Metrics Assuring Usefulness

In this section I will apply the theory presented in section 4.5, operationalized by means of inquiring metrics in section 6.4, to the four modeling techniques to evaluate their usefulness for modeling business processes.

System Expressiveness

System Expressiveness addresses the extent to which techniques are capable of modeling the discrete dynamic systems paradigm in general, at this point regardless of the specific requirements for modeling business processes.

Event Driven Process Chains

When the process logic of the EPC process view is considered, i.e. functions, events and logical connectors between them, leaving out the organizational units, application systems and entity types, the EPC is suitable for the description of the state space of a discrete dynamic system. The state of such a system is a vector consisting of Boolean values for each event in the process view. The Boolean value indicates whether an event has occurred or not. The state of the discrete dynamic system described in the EPC process view can therefore be described as a vector \( v \in \mathbb{B}^n \), in which \( n \) is equal to the number of events that are modeled and each element of \( v \) describes whether or not the event has occurred. The following example illustrates this principle.
Lawful transitions between states are modeled by means of functions and logical connectors that connect the events with each other. Take again the example above. As said, when event 1 'enter new products' or 2 'enter products provided by new suppliers' have occurred, then the function 'assign products to sales office' can be carried out and the result will be that the event 3 'product assigned' will occur. Taking into account that in this case, the state of the system is described by the vector $v \in \mathbb{B}^3$, an example of a lawful transition modeled here is the transition between $v = \langle \text{true, false, false} \rangle$ and $v = \langle \text{true, false, true} \rangle$. This is the case when the event 1 'enter new products' has occurred and the function 'assign products to sales office' is carried out, resulting in the event 3 'products assigned' to occur. An example of an unlawful transition is the transition between $v = \langle \text{false, false, false} \rangle$ and $v = \langle \text{false, false, true} \rangle$. In this case, the function is carried out without one of its triggering events having occurred.
In the EPC Process View, Organizational units assigned to functions model the actors causing transitions. The hierarchical relationship between organizational units is further worked out in the Organization view. EPC's do not allow the explicit modeling of actors inspecting the current state of the system to plan their future actions, actors are simply part of a deterministic machine, when certain events are true, a function must be carried out by the organization unit that is assigned to the function.

The EPC data view is capable of modeling another system's state space. This system is of another category than the system modeled by the process view as will be explained later. The state of this system is defined by the collection of instances (extension) of the defined entity types and relationship types. Each instance of an entity type is defined by its attribute values, each instance of a relationship type is defined by the instances of entity types that play a role in this relationship. A state change of this system is e.g. the addition or deletion of an instance of an entity type or the change of a property of a single instance of an entity type. Each instance of an entity is a vector \( v \in \mathbb{R}^n \), in which \( n \) denotes the number of attributes the instance has. However, important to note is that this system state space is of another system category than the state space modeled by the EPC process view. Two different categories of systems are modeled, with little relation between the two and therefore, the view presented in the EPC data view is not per se a further description of the previously mentioned states in the process view. This finding will be discussed later on when the business suitability property is addressed.

**Dynamic Essential Modeling of Organizations**

The DEMO transaction concept establishes a connection between two system categories: the coordination system and the production system. Regarding the coordination system, each transaction consists of a number of predefined states and lawful transitions between them. Each transition is the result of a coordination act between two parties, the initiating and executing actor of the transaction. On the other hand, each transaction causes exactly one transition in the production system, a transition from the initial transaction state to the state that the production act is carried out. Transitions in the coordination system reflect the negotiation about the transition in the production system.

The state space of the production system can be viewed as a vector \( v \in \mathbb{B}^n \), in which \( n \) is equal to the number of transactions that are modeled and each element of \( v \) describes whether or not the production act has occurred. Lawful transitions between these production states are further specified and restricted in the process phase model.
Figure 7-9: DEMO Transaction Pattern [revisited]

The state space of a single transaction in the coordination system can be viewed as a vector \( v \in T \), in which \( T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \). The elements of \( T \) are in accordance with the states mentioned in Figure 7-9. The element 0 is introduced as the initial state. For a single transaction, the lawful transitions are predefined and can be modeled as a mapping \( C \subseteq T \times T = \{<0, 1>, <1, 2>, <2, 4>, <4, 5>, <1, 6>, <6, 1>, <6, 9>, <2, 7>, <7, 2>, <7, 10>, <4, 8>, <8, 4>, <8, 2>, <8, 11>\} \). The state space of the complete coordination system consisting of \( n \) transactions can be viewed as a vector \( v \in T^n \). For a system of \( n \) transactions, lawful transactions can be modeled as \( C \subseteq T^n \times T^n \), encompassing the predefined transitions within transactions restricted by causal and conditional relationships between transactions as specified in the process phase model.

Take for example a business system consisting of two transactions: The delivery of goods (T1) and the payment of the delivered goods (T2). The state of the coordination system is a vector \( v \in T^2 \), the first value representing the state of transaction T1, the second value representing the state of T2. The initial state is \( <0, 0> \), indicating that both transactions are in the initial state. A lawful transition is now e.g. \( <0,0>, <1,0> \) \( \in C \), a transition that represents the initiation of transaction T1 (transition from the initial state to the Requested state). An example of an unlawful transition is e.g. \( <0,0>, <0,1> \) \( \notin C \), in which the payment transaction is started without a handled request for the delivery of goods. The payment transaction (T2) is can e.g. be started after the promise to deliver goods (T1): \( <2,0>, <2,1> \) \( \in C \).
Each transaction has an initiating and an executing actor. According to the transaction principle, the executing actor is responsible for the transition in the production system. Furthermore the executing actor is responsible for the executor part of the transitions in the coordination system (e.g. to promise to carry out the production action, to state that the production action is carried out). The initiating actor is responsible for the initiator part of the transitions in the coordination system (e.g. to request a production action, to accept the result of a production action). By means of the DEMO inspection concept, actors can inspect the current state of one ore more transactions to influence their future actions (transitions).

Also the DEMO Fact model (not described in this thesis but e.g. in Dietz (1996)) is capable of modeling a systems state space. This system is of another category than the production and coordination system discussed above. In this system, the adding or deletion of a fact causes a transition. The fact model is tightly linked to the interaction model: each transaction results in a new fact. Therefore, the state space of the production system is a subset of the state space of the system modeled by the fact model: A transition in the production system will also take place in this system because a production fact is added. Next to these facts, also other facts, not related to transactions are added to specify additional production information. However, because this model was not part of the evaluation in this thesis, it will not be mentioned in the evaluation results further on.

**Petri-net Based Modeling**

The Petri-net modeling technique is known to be a precise, formal approach for modeling a systems state space and lawful transitions. For the variant used for the evaluation in this thesis, the state of the system can be viewed as a vector \( v \in \mathbb{N}^n \), in which \( n \) is equal to the number of Petri-net places and each element of \( v \) reflecting the number of tokens in that place. Lawful transitions between these states are further specified by means of Petri-net transitions, allowing precedence, iterations, exclusiveness and parallel execution.
Figure 8-4: PBM Process Model Example

The state space and lawful transitions will be explained by means of the Petri-net example sketched above. There are three places: 'order to be fulfilled', 'order is entered' and 'order is fulfilled'. Therefore, the current state of the system can be viewed as a vector $v \in \mathbb{N}^3$. Let us presume that in the initial state of this process, one token is present in the first place. Therefore, the initial state of this system is $v = \{1, 0, 0\}$. When the transition 'Order to be entered' is carried out, the token moves from place 1 to place 2, and therefore, the next state of the system will be $v = \{0, 1, 0\}$. By this we can conclude that $\langle\{1, 0, 0\}, \{0, 1, 0\}\rangle$ is a lawful transition. The total set of lawful transitions is the following: mapping $C \subseteq \mathbb{N}^3 \times \mathbb{N}^3 = \langle\{1, 0, 0\}, \{0, 1, 0\}\rangle, \langle\{0, 1, 0\}, \{0, 1, 0\}\rangle, \langle\{0, 1, 0\}, \{0, 0, 1\}\rangle$. 

In the business control model it can be specified which actor (business function) is responsible for carrying out the transitions in a certain process that is carried out by this business function. However the assignment of transitions to actors is more coarse-grained than in the other techniques, since a function is assigned to a whole process instead of a single transition. In a way comparable to EPC, PBM does not allow the explicit modeling of actors inspecting the current state of the system to plan their future actions, actors are simply part of a deterministic machine, when certain places are filled by tokens, a transition must be carried out by the business function that is assigned to that transition.

Also the PBM Information Model is capable of modeling a systems state space. The adding or deletion of a fact causes a transition in this system. The information model is tightly linked to the process model: each transition in a process results in a new fact. Next to these facts, also other facts, not related to the process transitions
are added to specify additional business information. To conclude with, the state space of the system modeled by the information model partly overlaps with that of the business system as described above.

**Business Modeling with UML**

The UBM Goal Model can be viewed as a specification of the desired end states of the various business processes that are modeled. Together, the process model and goal model specify to a certain extent the state space of the business system. To a certain extent, because states are not explicitly modeled by means of a modeling concept and the semantics of the process model are not formally defined. We can presume that each process has a begin and end state and that transitions between implicit intermediate states take place every time an activity is carried out or a choice is being made.

In order to model the actors that cause the business transitions, one could make (mis)use of the resource concept offered by the process modeling technique. An actor-like concept is not offered. Consequently it cannot be modeled that this actor changes the history of the system nor that actors inspect the system state.

The UBM Conceptual model can be used to model the state space of another system that partially overlaps the business system. E.g. goals that were modeled in the goal model can be further specified in the conceptual model, so that the transition of adding a new goal instance to a goal class can be viewed as equivalent for achieving a goal (state) in the business system. However, next to goals, a range of other concepts can also be further specified in the conceptual model, e.g. resources and processes. In general, the coupling between conceptual model and the rest of the model cycle is open for interpretation.

The overall findings are summarized in the following table.

<table>
<thead>
<tr>
<th>Metric Name: WSE; Metric Type: Inquiring</th>
<th>EPC</th>
<th>DEMO</th>
<th>UBM</th>
<th>PBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to model ...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 System state space</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2 Lawful system transitions</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3 Actors</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>4 Actors changing the history of the system</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>5 Actors inspecting the system state</td>
<td>+/-</td>
<td>+</td>
<td>-</td>
<td>+/-</td>
</tr>
<tr>
<td><strong>Total Score</strong></td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>
Business Suitability
In the previous subsection it was investigated to what extent the modeling techniques that are evaluated throughout this thesis allow the modeling of discrete dynamic systems. The exact type of the systems, actors and transitions was to a certain degree left open. In this subsection, within the framework of evaluating the business suitability of the techniques, it is specifically studied what kind of systems, transitions and actors the techniques distinguish to model business processes.

Event Driven Process Chains
As concluded so far, the EPC modeling technique models two types of systems: the business system, in which transitions occur by the carrying out of the EPC functions in the EPC process view and the information system, in which transitions occur when an EPC entity is added, modified or deleted as part of the carrying out of an EPC function. Entities are both modeled in the EPC process view and in more detail in the EPC data view. The transition of both systems cannot be mapped to each other, e.g. in the sense that the carrying out of an EPC function is equivalent the adding an instance of an entity in the data view.

Figure 8-5: Systems Modeled by EPC and their interrelationship
Figure 8-5 shows the two system categories modeled by EPC: the business system and the information system. There is a relationship between the two in the sense that during a certain business transition, the information transitions are restricted to the entities associated with the activity. Recall for instance Figure 8-3, during
the transition 'assign product to sales office' only those information transitions concerning the entity 'order record' can take place.

As can be seen in the figure above, the actor inspects and acts upon the business system as well as on the information system. Each time a business transition occurs, this might result in transitions in the information system. Take again the example in Figure 8-3. In order to carry out the transition 'to assign product to sales office' in the business system, the actor 'distribution' might inspect the 'product' in the information system and might change the 'order record' in the information system.

Let us now further examine the exact nature of the two systems modeled by EPC. With regard to the transitions in the business system, little can be said about their nature. EPC's lack any definition on the category of systems that this business system is and thus leaves a large degree of freedom to the modeler with respect to the way to interpret business processes. Consequently, its deployment can range from low-level logistic processes where the business processes is viewed as a physical system, to the modeling of e.g. processes where information is processed, depending on the modelers idea of what a business process is. Most of the known EPC Examples (Scheer, 1994, Scheer et al, 2000) however, reveal that the transitions have an administrative character such as 'Enter order details', 'Check order limit', 'Register Payment', and so on. Thus, although EPC's are not restricted to the application of administrative process, it has been applied there often. The actors carrying out these functions are human employees as part of organizational units.

The other system modeled by EPC's is a typical example of an information system in which transitions are the adding, deleting or modification of entities. The actors carrying out these transitions are the same as the actors carrying out the transitions in the business system: human employees as part of organizational units. The different system categories modeled by EPC are summarized in the following table.
Table 8-19: EPC System Categories

<table>
<thead>
<tr>
<th>System Categories</th>
<th>Event Driven Process Chains</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business System</td>
<td>Unspecified, in practice mostly administrative</td>
<td>Human, in organizational Units</td>
</tr>
<tr>
<td>Information System</td>
<td>Informational</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td>Limited consistency between Business and Information System: Business state restricts the set of entities on which informational transitions are applicable</td>
<td></td>
</tr>
</tbody>
</table>

Dynamic Essential Modeling of Organizations

As argued before, the DEMO technique makes a clear distinction between communicative and production actions and consequently a system consisting of communicative states and transitions called coordination system and a system consisting of production states and transitions called production system. These two systems restrict each other. The coordination system models the negotiations between actors about production transitions in the production system. On the other hand, the occurrence of production transitions may restrict transitions in the coordination system.

Next to these two systems, the DEMO technique, in particular the DEMO fact model, models the state space of another system consisting of informational transitions in which transitions are the adding or deletion of facts. The DEMO fact modeling technique is not discussed in this thesis. The state space of this system partly overlaps with the production system since each transition in the production system (the carrying out of a production action) results in a new, original, fact and thus a transition in the information system. The three systems and their interrelationship are depicted below.
Figure 8-6: Systems modeled by DEMO and their interrelationship

As indicated, the actor is able to inspect as well as change the state of all the systems. When we compare the modeled systems in DEMO as well as their interrelationship with the modeled systems in EPC and their relationship, we see that the coordination system imposes restrictions on the production and information system and vice versa, in a way that the actor is prohibited to make unlawful transitions. For example, a production transition can only take place when the actor has promised to carry out this production transition. This promise is a transition in the coordination system. Vice versa, the actor can only declare that a production action was carried out after this transition has taken place in the production system. In itself, to declare the carrying out of a production action is a transition in the coordination system.

Furthermore, compared to EPC's, there are more restrictions imposed on the actor and its ability to change the state of production and information system. These systems are to a certain extent automatically kept consistent since a production transition always has a corresponding information transition, the creation of the fact that the production transition has been carried out.
Table 8-20: DEMO System Categories

<table>
<thead>
<tr>
<th>System Categories</th>
<th>Dynamic Essential Modeling of Organizations</th>
<th>Transitions</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production System</td>
<td>Essential business transitions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordination System</td>
<td>Communicative actions between two actors, together these actions constitute a conversation about the production transition to achieve</td>
<td>Responsibility, Competence, a unique actor is assigned to each essential business transition</td>
<td></td>
</tr>
<tr>
<td>Information System</td>
<td>Informational</td>
<td>Transitions in the production system impose restrictions on transitions in the coordination system and vice versa. Each transition in the production system has a corresponding transition in the information system.</td>
<td></td>
</tr>
</tbody>
</table>

Table 8-20 shows more insight in the type of transitions that take place in the mentioned systems. Some further explanation is required. Transitions in the production system are called essential business transitions. Essential transitions are transitions that add value to the business process, abstracting from the way these transitions are supported by means of administrative, logistic systems and so on. These latter systems are considered to be a matter of realization of which one wants to abstract. For example, to deliver goods is an essential transition, the filling in of the order form or the form for accepting the delivered goods, are administrative and thus no essential business transition, as well as the transport of the delivered goods. Each transition in the production system is part of a negotiation pattern in the coordination system. Take for instance the delivery of goods, which is preceded by a negotiation between customer and supplier about the conditions of delivery and is succeeded by a negotiation about the acceptance of the goods. These are more strict criteria for distinguishing transitions than other techniques offer.

DEMO leaves in the middle how responsibilities and competences of an actor are realized. Responsibilities and competences can be supported by computer systems or can be carried out by human actors, this is left open since once again this is a matter of realization. It is e.g. possible that one person performs several actors roles and vice versa.
Petri-net Based Modeling

Finally the business suitability of PBM is considered. PBM distinguishes two kinds of systems: the business system and the information system. The PBM process model captures the business system. Transitions that take place are modeled by Petri-net transitions. Little can be said about the nature of these transitions. PBM lacks any definition on the category of systems that this business system belongs to and thus leaves a large degree of freedom to the modeler with respect to the way to interpret business processes. Its deployment can range from low-level logistic processes where the business processes is viewed as a physical system, to the modeling of e.g. processes where information is processed, depending on the modelers idea of what a business process is.

The PBM information model captures the state space of the information system. Transitions are the adding and deletion of facts. Comparable to DEMO, each business transition corresponds to a transition in the information system. Take the example in Figure 8-4, the business transition 'Enter Order' results also in a new fact that the order is entered (see Figure 7-21 for the information model containing this fact type). Furthermore actors are modeled by means of business functions. Altogether this results in the following illustration.

Figure 8-7: Systems modeled by PBM and their interrelationship
The table below shows the type of transitions that are modeled.

**Table 8-21: PBM System Categories**

<table>
<thead>
<tr>
<th>System Categories</th>
<th>Petri-net based Modeling Transitions</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business System</td>
<td>Unspecified</td>
<td>Business functions, often comparable to business departments</td>
</tr>
<tr>
<td>Information System</td>
<td>Informational</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
<td>A transition in the business system has a corresponding transition in the information system</td>
</tr>
</tbody>
</table>

*Business Modeling with UML (UBM)*

The UBM modeling technique models two types of systems: the business system, in which transitions occur by the carrying out of UBM activities and the information system, in which transitions occur when an UBM classes are added, modified or deleted as part of the carrying out of an UBM activity. For those resources in the UBM process model that are also modeled in the conceptual model, there is a relationship between the business and information system: Information transitions are restricted to the resources that are associated with a certain business transition. The fact that UBM processes, goals, et cetera can also appear in the conceptual model and the lack of formalization makes it impossible to give an account of the precise interrelationship between business system and information system. In any case, transitions of both systems cannot be mapped to each other, e.g. in the sense that the carrying out of an UBM action is equivalent to adding an instance of a class in the conceptual model. The two systems and their interrelationship are depicted below. Striking is of course also the absence of the concept of an actor that inspects and changes the system's state.
Figure 8-8: Systems Modeled by UBM and their interrelationship

Having identified the two systems modeled, we can take a close look at the kind of transitions that are modeled. The table below gives an overview.

Table 8-22: UBM System Categories

<table>
<thead>
<tr>
<th>System Categories</th>
<th>Business Modeling with UML</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business System</td>
<td>Unspecified</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Information System</td>
<td>Informational</td>
<td></td>
</tr>
<tr>
<td><strong>Consistency</strong></td>
<td>Limited consistency between Business and Information System: Business state restricts the set of concepts on which informational transitions are applicable, furthermore lack of clarity due to the fact that UBM concepts such as processes and goals can also appear as concepts and formalization lacks</td>
<td></td>
</tr>
</tbody>
</table>
The evaluation above results in the following values for the inquiring business suitability measure:

<table>
<thead>
<tr>
<th>System Categories Covered</th>
<th>EPC</th>
<th>DEMO</th>
<th>UBM</th>
<th>PBM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business System</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Criterion for distinguishing transitions</td>
<td>-</td>
<td>+</td>
<td>+/-</td>
</tr>
<tr>
<td></td>
<td>Criterion for distinguishing actors</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Information System</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Criterion for distinguishing transitions</td>
<td>+/-</td>
<td>+</td>
<td>+/-</td>
</tr>
<tr>
<td></td>
<td>Criterion for distinguishing actors</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Communication System</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Criterion for distinguishing transitions</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Criterion for distinguishing actors</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Coherency Between System Categories</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Formally defined interrelationships</td>
<td>+/-</td>
<td>+</td>
<td>+/-</td>
</tr>
</tbody>
</table>

With regard to the criterion for distinguishing transitions in the information system, DEMO and PBM score higher than EPC and UBM. This is due to the fact that an information transition in DEMO and PBM is simply the adding or deletion of a fact. In EPC and UBM it can be the adding, deletion or modification of an entity (EPC) of class (UBM). In case of a modification, a transition can be the change of an attribute or a change in a relationship with another entity (EPC) or association with another class (UBM). It can be concluded that informational transitions in DEMO and PBM are simpler and more straightforward.

### 8.4 Empirical Study on the Interpretation of the Proposed Metrics

#### Collection of Empirical Data

Empirical data have been collected to contribute to the discussion on the interpretation of the theoretical measures discussed before. This has been established by carrying out a number of modeling sessions in which about 30 graduate students applied each of the four techniques discussed in the previous chapter on a single case. The case was to model the business processes of a fictive library, which were described on one page in natural language. The case description is depicted below.
Figure 8-9: Library Case Description

For each of the four modeling techniques, a two-hour session was held. During the first hour, the technique was explained to the subjects and a fact sheet containing the model tables and examples that were presented in chapter 7 was handed out to each of the subjects. During the second hour, the subjects constructed models of the library using the technique. During this phase, the subjects were allowed to make use of the fact sheet that was handed out the first hour. The same instructor supervised each of the four sections and the fact sheets describing the four techniques were comparable in size. The order in which the sessions were carried out was: 1) Event Driven Process Chains, 2) Dynamic Essential Modeling of Organizations, 3) Petri-net Based Modeling and 4) Business Modeling with UML.

The quality of each of the 120 (30 × 4) resulting models was investigated using the criteria discussed in section 4.3, e.g. the number of statements in the model that correctly used the prescribed notation (empirical quality), the number of statements in the model that were in accordance with the syntax of the modeling language (syntactical quality), the number of statements contradicting each other (consistency), et cetera. Figure 8-10 shows an example to illustrate the procedure. Subjects are made anonymous by numbering them A1, B1, C1, etc. For each of the subjects the number of statements in a particular model (size) and errors were
counted. E.g. subject E1 had 42 statements in the DEMO Business Systems Architecture of which 3 were empirically incorrect. Also inter model consistency and well-formedness errors were measured, e.g. the number of transactions appearing in the business systems architecture but not in the process phase model.

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Business Systems Architecture</th>
<th>Process Phase Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>30 empirical 5 syntactic 0 well-formedness consistency 0</td>
<td>15 empirical 9 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>B1</td>
<td>32 empirical 1 syntactic 2 well-formedness consistency 0</td>
<td>16 empirical 0 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>C1</td>
<td>29 empirical 0 syntactic 1 well-formedness consistency 0</td>
<td>13 empirical 0 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>D1</td>
<td>33 empirical 0 syntactic 0 well-formedness consistency 0</td>
<td>16 empirical 2 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>E1</td>
<td>42 empirical 3 syntactic 0 well-formedness consistency 0</td>
<td>24 empirical 0 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>F1</td>
<td>32 empirical 0 syntactic 1 well-formedness consistency 0</td>
<td>19 empirical 10 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>G1</td>
<td>32 empirical 0 syntactic 0 well-formedness consistency 0</td>
<td>17 empirical 10 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>H1</td>
<td>36 empirical 6 syntactic 0 well-formedness consistency 0</td>
<td>15 empirical 3 syntactic 0 well-formedness consistency 0</td>
</tr>
<tr>
<td>I1</td>
<td>77 empirical 0 syntactic 0 well-formedness consistency 0</td>
<td>17 empirical 0 syntactic 0 well-formedness consistency 0</td>
</tr>
</tbody>
</table>

**Figure 8-10: Example of the results for DEMO (Partly)**

It must be stressed that the modeling sessions must not be interpreted as an ideal experiment where two techniques are compared that differ on one independent variable. I have chosen for a practical approach to compare 4 existing modeling techniques that are different from each other concerning many variables (e.g. notation, syntax, order in which models are constructed, number of concepts). If it were possible to construct pairs of artificial modeling techniques that differ only on one variable, this kind of experiment would not be interesting for those participating in the modeling sessions and relevant for those reading the results. Other factors that might have influenced the outcome are the order in which modeling techniques were discussed and applied as well as the knowledge of and bias towards existing modeling techniques such as UBM and DEMO by the subjects. This heavily reduces the extent to which the achieved results can be generalized.
<table>
<thead>
<tr>
<th>Aspect Modeling Technique</th>
<th>Size</th>
<th>Errors per statement</th>
<th>Syntactical Correctness</th>
<th>Intra Model Consistency</th>
<th>Intra Model Well-formedness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Nr of statements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Driven Process Chains</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process View</td>
<td>66.07</td>
<td>0.021695</td>
<td>0.037841</td>
<td>0.004036</td>
<td>0.340909</td>
</tr>
<tr>
<td>Data View</td>
<td>12.27</td>
<td>0.067935</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Functional View</td>
<td>16.47</td>
<td>0.004049</td>
<td>0.010121</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Organizational View</td>
<td>10.60</td>
<td>0.100629</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Total</td>
<td>105.40</td>
<td>0.048577</td>
<td>0.011991</td>
<td>0.001009</td>
<td>0.085227</td>
</tr>
<tr>
<td>Dynamic Essential Modeling of Organizations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Architecture</td>
<td>30.70</td>
<td>0.060217</td>
<td>0.008885</td>
<td>0.046397</td>
<td>0.424242</td>
</tr>
<tr>
<td>Process Phase Diagram</td>
<td>14.79</td>
<td>0.178279</td>
<td>0.002049</td>
<td>0.006148</td>
<td>0.012295</td>
</tr>
<tr>
<td>Transaction Result Table</td>
<td>10.48</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.164740</td>
</tr>
<tr>
<td>Bank Fact Table</td>
<td>1.52</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.200000</td>
</tr>
<tr>
<td>Total</td>
<td>57.48</td>
<td>0.059624</td>
<td>0.002733</td>
<td>0.013136</td>
<td>0.200319</td>
</tr>
<tr>
<td>Petri-net Based Modeling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Control Model</td>
<td>16.23</td>
<td>0.035545</td>
<td>0.052133</td>
<td>0.000000</td>
<td>0.038462</td>
</tr>
<tr>
<td>Business Process Model</td>
<td>38.81</td>
<td>0.027750</td>
<td>0.041625</td>
<td>0.007929</td>
<td>0.029126</td>
</tr>
<tr>
<td>Business Information Model</td>
<td>26.42</td>
<td>0.155750</td>
<td>0.004367</td>
<td>0.001456</td>
<td>0.076923</td>
</tr>
<tr>
<td>Total</td>
<td>81.46</td>
<td>0.073015</td>
<td>0.037448</td>
<td>0.003128</td>
<td>0.048170</td>
</tr>
<tr>
<td>Business Modeling With UML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highlevel Process Model</td>
<td>32.03</td>
<td>0.037675</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.068966</td>
</tr>
<tr>
<td>Lowlevel Process Model</td>
<td>60.17</td>
<td>0.014327</td>
<td>0.008023</td>
<td>0.000000</td>
<td>0.018349</td>
</tr>
<tr>
<td>Goal Model</td>
<td>6.62</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.272727</td>
</tr>
<tr>
<td>Conceptual Model</td>
<td>14.66</td>
<td>0.171765</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>Total</td>
<td>113.48</td>
<td>0.055942</td>
<td>0.002006</td>
<td>0.000000</td>
<td>0.090010</td>
</tr>
</tbody>
</table>
Table 8-24 and Table 8-25 show the empirical data that was collected during the modeling sessions. Table 8-24 shows for each aspect modeling technique the average number of errors the modelers made with regard to empirical correctness, syntactical correctness, intra model consistency and intra model well-formedness. Table 8-25 shows for each multi modeling technique the number of errors made with regard to inter model well-formedness and inter model consistency.

<table>
<thead>
<tr>
<th>Table 8-25: Empirical results of the modeling sessions (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter model well-formedness</td>
</tr>
<tr>
<td>Errors made</td>
</tr>
<tr>
<td>Event Driven Process Chains</td>
</tr>
<tr>
<td>Dynamic Essential Modeling of Organizations</td>
</tr>
<tr>
<td>Petri-net Based Modeling</td>
</tr>
<tr>
<td>Business Modeling with UML</td>
</tr>
</tbody>
</table>

In the remainder of this section the quality properties of models, introduced in section 4.3 are taken as a means to structure the discussion. For each property, the outcomes of the empirical study are presented and discussed together with a discussion on the extent to which the proposed metrics in section 6.3 can predict this outcome.

**Empirical Correctness**

The empirical correctness of a model concerns the notation of concepts and knowledge in diagrams. Consequently, empirical errors concern errors made in the notation of concepts and knowledge in diagrams. The average percentages of empirical errors, calculated for the four multi modeling techniques do not yield significant differences: 5% for EPC, 6% for DEMO, 7% for PBM and 6% for UBM.

Taking a closer look at the individual aspect modeling techniques, a remarkably higher score than average is found for the DEMO Process Phase Diagram (18%), the UBM Conceptual Model (17%) and the PBM Business Information Model (16%). A remarkably low amount of errors was measured for the UBM Goal model (0%), the EPC Functional view (<1%), the UBM Low Level Process model (1%) and the PBM Business Process Model (3%).

The values calculated for the AMI metric earlier in this chapter predict these empirical results. Techniques with a large number of empirical errors, the DEMO Process Phase Diagram and the PBM Business Information Model, have the lowest AMI value (0.6). The high number of empirical errors in the UBM conceptual model is an exception here. Techniques with a low number of empirical errors, the EPC Functional view, the UBM Low Level Process model and the PBM Business Process Model, have a high AMI value, respectively 0.9, 0.9 and 0.8.
It can be concluded that there is correlation between the AMI metric and the empirical data. The higher the AMI value, the lower the number of empirical errors to be expected when applying the technique.

**Syntactical Correctness**

**Individual Statements in Accordance with the Syntax**

The syntactical correctness of a model concerns the accordance of statements in the model with those allowed by the modeling language. Consequently, syntactical errors are statements in the diagram that do not follow the syntax of the modeling language. The average of syntactical errors for the four multi modeling techniques are: 1% for EPC, <1% for DEMO, 4% for PBM and <1% for UBM. Striking is the 4% for PBM, caused by the PBM Business Control Model (5%) and the PBM Business Process Model (4%).

High error rates are counted for the PBM Business Control Model (5%), the EPC Process View (4%) and the PBM Business Process Model (4%). Remarkably low error rates are counted for the EPC Data View (0%), the EPC Org View (0%), the DEMO tables (0%), the UBM High-level Process Model (0%) and the UBM Conceptual Model (0%).

In chapter 6, I have proposed the Aspect Model Syntactic Freedom metric (ASF), presumed to be an indicator for the number of syntactical errors to be expected when a certain aspect modeling technique is applied to model a certain business process. To be precise, it was presumed that a high ASF value, i.e. a high level of syntactic freedom, results in a lower syntactical error rate. However, the investigation of the pre-measured ASF compared to the post-measured syntactical error rate yields interesting deficiencies that limit the applicability of the ASF metric and ask for further explanation. In the table below I have compared the results for the most remarkable outcomes as identified above.

**Table 8-26: Comparison ASF metric values and Syntactic Error Rates**

<table>
<thead>
<tr>
<th>Aspect Modeling Technique</th>
<th>ASF</th>
<th>Err Rate</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBM Business Control Model</td>
<td>0.44</td>
<td>5%</td>
<td>No</td>
</tr>
<tr>
<td>EPC Process View</td>
<td>0.22</td>
<td>4%</td>
<td>Yes, ASF&lt;Average</td>
</tr>
<tr>
<td>PBM Business Process Model</td>
<td>0.19</td>
<td>4%</td>
<td>Yes, ASF&lt;Average</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aspect Modeling Technique</th>
<th>ASF</th>
<th>Err Rate</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPC Data View</td>
<td>0.20</td>
<td>0%</td>
<td>No</td>
</tr>
<tr>
<td>EPC Org View</td>
<td>0.50</td>
<td>0%</td>
<td>Yes, ASF&gt;Average</td>
</tr>
<tr>
<td>DEMO Tables</td>
<td>0.33</td>
<td>0%</td>
<td>No</td>
</tr>
</tbody>
</table>
For example, the ASF values for the EPC Data View, the DEMO Tables and the UBM Conceptual Model are not indicative for the error rate. For all three, the error rate is 0 despite the ASF measures not indicating a high degree of freedom. For the EPC Data view and the UBM Conceptual model this can be explained as follows.

Imagine a UBM Conceptual Model that models two classes, named "Book" and "Author" and one binary association between the two, named "is_author_of". The diagram is as depicted in the figure below.

![Example Conceptual Model](image)

**Figure 8-11: C-We model with simple binary association**

The C-We model that we have drawn specifies a UBM conceptual model that allows the modeling of binary associations. Its ASF is 1.0 (maximal): It is not possible to generate syntactically incorrect statements, since there is no other concept that the knowledge primitives can refer to except to the concept class. The same holds for the diagram: you cannot draw a line other than from one class to another. In this case, the ASF is 1.0 and the number of expected syntactical errors in the resulting diagrams is 0.

However, the C-We model above does not allow the modeling of n-ary associations between classes and in addition to that, does not allow the modeling of more than one association between two classes. A C-We model that is more representative for the UBM Conceptual model is the C-We model depicted in Figure 8-12. This model
was actually used throughout this thesis. It allows for the modeling of n-ary associations as well as more than one association between the same classes.

We see now, that the drawing of a simple binary association between two classes results in much more individual knowledge primitives than was the case in the example above. In addition to that, the ASF is much lower, since now it is possible (at least in the C-Me model) to construct syntactic incorrect individual knowledge primitives, e.g. an instance of the knowledge primitive “plays” referring to two instances of the concept “class” instead of referring once to an instance of the concept “class” and referring once to an instance of the concept “association”. The drawing of a simple association between two classes, however, instantiates a number of syntactically correct individual knowledge primitives. Although the C-Me model predicts that syntactically incorrect knowledge primitives can be expressed, these will never occur due to the notation used in the diagram.

Figure 8.12: Extended C-Me Model supporting n-ary association

It can be concluded that the type of constructs discussed above, appearing in the EPC data model (the relationship concept), the UBM Conceptual Model (the association concept) as well as in the PBM Business Control Model (the request feedback loop concept), distort the predictive value of the ASF metric.
A comparable situation occurs for the DEMO tables. The notation is such that one statement in a table results in a number of individual knowledge primitives that are per definition correct although in theory, with another notation, syntactically incorrect individual knowledge primitives could exist.

Another outcome that has to be discussed is the PBM Business Control Model. Given the ASV value, the syntactic error rate is rather high. The rather high syntactic error rate is explained by taking a closer look to the kind of errors that the modelers made during this session. It turns out that the vast majority of errors made were errors regarding the syntax of the naming of request feedback loops. These should be in the form "<Object> to be <Verb>". However, the specification of this syntax is found in the C-Me model of the PBM Business Process Model, not in the Business Control Model. It would have been more correct to extend the C-Me model of the Business Control Model with these knowledge primitives and concepts and then to compare the ASV with the error rate, or, to leave these knowledge primitives and concepts out and then not to measure the errors in the syntax of the naming of the request feedback loops. It can be concluded here that the error rate is influenced by syntactic errors that were wrongly counted because the C-Me model did not restrict the syntax of these statements.

To conclude, the ASF is a measure for determining the syntactical freedom given the knowledge primitives and concepts in the C-Me model, when the notation of these primitives and concepts is disregarded. Certain notations, in which a single notation construct represents a number of knowledge primitives, reduce the predicted number of errors. A high ASF indicates a low syntactic error rate, however, a low ASF does not necessarily predict a high error rate, it may depend on the notation that is chosen.

**Intra Model Consistency**

The intra model consistency of a model concerns the possible contradiction between individual statements in an aspect model. Consequently, errors in the intra model consistency are statements in the diagram that contradict each other. The average intra model consistency error rate amongst the modelers is low: <1% for EPC, 1% for DEMO, <1% for PBM and 0% for UBM.

A significantly higher 5% error rate is found for the DEMO Business Architecture. On the other hand, zero values are calculated for a number of aspect modeling techniques: the EPC Data, EPC Functional and the EPC Organizational view, the DEMO tables, the PBM Business Control Model and all of the UBM aspect modeling techniques.

In chapter 6, I have proposed to count the percentage of knowledge primitives that have a uniqueness constraint as an indicator for these type of errors occurring in
resulting models, the AUC metric. For example, in the DEMO Business Architecture, the detailed results of the modeling sessions show that the uniqueness constraint of the knowledge primitive “execution” is violated most often. Often, it is erroneously modeled that DEMO elementary actors execute more than one transaction, which results in a contraction: which of the transactions is the actor executing.

However, contrary to what was expected there is no correlation between the AUC value and the intra model inconsistency error rate in the modeling sessions. Take for instance the EPC function view: the AUC is 1.0 since the only knowledge primitive in it has a uniqueness constraint on it (a sub function cannot have two super functions). However, this constraint is never violated during the modeling sessions, so the error rate = 0.

The cause of the absence of correlation is that the AUC is too course grained to make precise predictions on the number of consistency errors to be expected. An example is the DEMO Business Architecture. Simply counting the number of knowledge primitives with uniqueness constraints did not predict this error rate, since it is only one primitive that is violated most. An explanation for the violation of certain uniqueness constraints might be the notation used. Figure 8-13 shows three ways that binary knowledge primitives with one uniqueness constraint are represented in diagrams. Notation A shows how the decomposition into sub functions is modeled in the PBM Business Control Model, notation B shows how the same decomposition is modeled in the EPC function view. The underlying knowledge primitive is “decomposition” and the uniqueness constraint on it is that one sub function can only have one super function. Notation C is the notation belonging to the knowledge primitive “execution” in the DEMO Business Architecture. The uniqueness constraints on this primitive are that one transaction has only one executing actor and that one actor executes exactly one transaction.

![Diagram](image)

Figure 8-13: Notations for Binary Knowledge Primitives with constraints
The example above shows that in the case of the DEMO Business Architecture, the uniqueness constraint has been violated since actor A executes both transaction A and transaction B. In the case of the PBM Business Control Model, no uniqueness constraint is violated. In fact, the uniqueness constraint cannot be violated due to the nature of the notation used. The same holds to a lesser extent for notation B. It is such that it does not invite the modeler to violate the uniqueness constraint of the underlying knowledge primitive.

To conclude with, the hypothesis that uniqueness constraints cause inconsistencies is true. However, measuring the percentage of knowledge primitives that have a uniqueness constraint does not predict the consistency error rate in the modeling sessions. There is no correlation between the two. The AUC is too course grained to predict this type of errors and the notation of constructs plays a role. A positive finding is, that certain notations can be used to assure that certain uniqueness constraints are not violated. For example, it could be considered to change the notation of the “execution” primitive in the DEMO Business Architecture to a notation in the sense of Figure 8-13.

**Intra Model Well-formedness**

The intra model well-formedness of a model concerns the possibility of models not being in accordance with well-formedness rules. The average intra model well-formedness rate amongst the modelers is as follows: 9% for EPC, 20% for DEMO, 5% for PBM and 9% for UBM. Striking is the 20% counted for DEMO.

A more detailed analysis reveals high error rates for the EPC Process View (34%), the DEMO Business Architecture (30%), the UBM Goal Model (27%) and the DEMO bank fact table (20%). On the other hand, zero values are calculated for the UBM Conceptual Model, the EPC Data View, the EPC Functional View and the EPC Organization View.

In chapter 6, I have proposed to count the percentage of knowledge primitives that have a requiredness constraint as an indicator for these type of errors occurring in resulting models, the ARQ metric. An example is the presence of a DEMO elementary transaction in the model without an actor that is the executor of the transaction.

A comparison between the theoretical ARQ values and the result of the empirical study shows little correlation, implying that the ARQ value must be interpreted with caution. For the DEMO Business Architecture, the ARQ measure worked well. The ARQ measured that 80% of the knowledge primitives in this diagram have a requiredness constraint. Indeed, the detailed results of the modeling sessions show that especially requiredness constraints on the knowledge primitives “initiation”
and "execution" were violated. The same holds for the DEMO Bank Fact Table where the constraints on the knowledge primitive "refers to" (referring to the concepts "fact type" and "object type") were violated. Also the EPC Process View scores a high ARQ (0.8) and a high error rate as well. These results however made turbid by the fact that also the unsoundness of a whole EPC Process Chain was counted as error while the C-Me modeling technique does not allow expressing such constraints on models.

When we consider the technique that scored no errors with regard to well-formedness, we expect low ARQ values. This holds for the EPC function view, but other techniques show striking discrepancies such as e.g. the UBM Conceptual Model and the EPC Data Model. Again it is concluded that the same notational issue as discussed in Figure 8-12 plays a role here. In these techniques, simple notational constructs represent a number of knowledge primitives including a number of constraints that are instantiated as a whole, without the possibility of a violation.

It can be concluded that some good examples of the applicability of the ARQ metric have been found such as for instance the DEMO Business Architecture. On the other hand, the incapability of the C-Me language to capture specific soundness rules that hold for e.g. the EPC Process View and the PBM Process Model distorts the outcome. Another distorting factor is again the problem that was earlier discussed where a single notational construct correctly instantiates a number of knowledge primitives and concepts.

**Inter Model Consistency**

Whereas the intra model consistency of a model concerns the possible contradiction between individual statements in one single aspect model, the inter model consistency deals with statements in different aspect models that might contradict each other. Consequently, errors in the inter model consistency are statements in different diagrams that contradict each other. Three of the four modeling techniques score an error rate of 0%: EPC, PBM and UBM, on the other hand, DEMO scores 1%. This score might be considered insignificant, but when we calculate the number of modelers that made inconsistent models instead of calculating the average number of knowledge primitives that was inconsistent, we see a larger discrepancy between DEMO and the other techniques: 0% for EPC, PBM and UBM on the one hand and 51% for DEMO on the other hand. I will demonstrate how this can be predicted making use of the proposed metrics.

It was presumed that knowledge primitives that can be modeled in different aspect models and have uniqueness constraints might yield inconsistencies between these aspect models. The WMCUC metric counts the number of coherency relationships
between aspect modeling techniques that contain knowledge primitives with a uniqueness constraint.

For EPC, the WMCUC = 0, since the only overlap there is between the different views is a single concept. There are no knowledge primitives that refer to these concepts (see Figure 7-7). Therefore there can never be inconsistent models and the error rate measured in the modeling sessions is therefore 0.

For UBM, the WMCUC = 0.5. This is rather high given that the modelers made no errors in the inter model consistency. The WMCUC is in particular high due to some consistency relationships between the UBM Conceptual Model and other models. The AMCUC metric reveals consistency relationships between the UBM Conceptual Model and 1) the UBM Goal Model, 2) the UBM High-level process model and 3) the UBM Low-level process model. However, the knowledge primitives on the one hand of the coherency relationship (in the process models) do not contain constraints. Therefore no inconsistencies can occur in spite of this being predicted by the WMCUC metric. A less liberal metric that measures coherency relationships with knowledge primitives with uniqueness constraints on both sides of the coherency relationship should have been proposed.

For PBM, the WMCUC = 0.66, because of coherency relationships with constraints between the Business Process Model and 1) the Business Control Model and 2) the Business Information Model. Again, this measure is rather high given that the modelers made no errors in the inter model consistency. In closer detail, the coherency relationship between BPM and BCM deals with the starting of new request feedback loops within processes. For no apparent reason, no modeler made consistency errors of this kind.

For DEMO, the WMCUC=0.16, due to a single coherency relationship between business architecture model and process phase model. As a result of this coherency relationship, modelers could make a business architecture model and process phase model that are not consistent. In fact they did, the number of consistency errors is not high, but 51% of the modelers made at least a mistake regarding this coherency relationship.

**Inter Model Well-formedness**

Whereas the intra model well-formedness of a model concerns the possibility of models not being in accordance with well-formedness rules with regard to one single aspect model, the inter model well-formedness concerns statements in different aspect models not being in accordance with well-formedness rules governing these aspect modeling techniques. The rules are part of the coherency relationships. Coherency relationships of the type EQUAL or SUBSET restrict modelers in a way that respectively the same concepts in one aspect model should also appear in the

---

229
other aspect model or a subset of a certain concept in one aspect model should appear in another aspect model. Therefore, the measures of the AMCNO and WMCNO metrics were proposed in chapter 6 and applied in this chapter to measure the mentioned coherency relationships. They will now be compared with the number of mistakes made regarding inter model well-formedness.

The error rates with regard to inter model well-formedness are as follows: DEMO 0.5%, UBM 2%, EPC 6%, and PBM 10%. When these values are compared to the WMCNO metric values, it can be concluded that, leaving out the results for DEMO, there is correlation between the two. The values for the WMCNO metric are: DEMO 0.66, UBM 0.33, EPC 0.5 and PBM 0.66. In order to explain this correlation, each modeling technique is studied in more detail.

For DEMO, the WMCNO = 0.66, which means that two third of the interconnections possible between the four DEMO aspect modeling techniques have at least an equality or subset coherency relationship. Most of these coherency relationships are equality relationships with regard to DEMO transactions. They specify e.g. that a DEMO Transaction in the Business Architecture Model should also appear in the Process Phase Model, that a DEMO Transaction in the Business Architecture Model should also appear in the DEMO Transaction Result Table, and so on. In principle, all these rules can be violated during the application of the DEMO technique. However, in the empirical study that was carried out, the rules were not violated so much, possibly due to the subject's familiarity with the technique or the fact that most of the rules concern the core concept of the DEMO technique, the DEMO transaction.

For UBM, the WMCNO = 0.33, the lowest value of all techniques studied. The reason for this is, that despite the fact that there are many coherency relationships between the UBM aspect modeling techniques, the coherency relationships do not restrict the modeler. For example, goals mentioned in the goal model might appear in the conceptual model, processes mentioned in the high level process model might appear in the conceptual model, and so on. There are three coherency relationships that could cause mall-formed models: 1) resources mentioned in the high level process model should also be mentioned in the low level process model, 2) in case there is a process that is mentioned in both high level as well as low level process model, the use of resources should be mentioned consistently and 3) goals mentioned in de goal model should also appear in the high level process model. The results of the empirical study show that indeed, in line with the WMCNO measure, a low error rate with regard to well-formedness was measured. The coherency relationship that was violated most was the first one mentioned above, resources in the high level model that were not mentioned in the low level model.

For EPC, the WMCNO = 0.5. Half of the interconnections possible between the aspect modeling techniques have at least an equality or subset coherency
relationship. The coherency relationships in EPC are quite simple: There are three relationships: 1) A function in the process view should appear in the function view, 2) An entity in the process view should also appear in the data view and 3) An organizational unit in the process view should also appear in the organizational view. A model that violates one of the rules is mal-formed. Despite the fact the coherency structure is quite simple, the vast majority of modelers made errors with regard to these relationships. A possible cause is that most of the modelers suffered from a lack of time during the experiment. This lack of time is explained by the complexity of the technique and the unbalanced way that concepts are distributed over aspect modeling techniques (see the remarks made below Table 8-17).

For PBM, the WMCNO = 0.66. This is together with DEMO the highest value measured. The high error rate measured during the empirical study provides evidence for correlation between theoretical metric and resulting error rate. The rule that was violated most was the rule that prescribes that each state in the process model should appear as a fact type in the information model.

Validity of Models
In Table 8-16, an overview of several CSV values was given. The CSV metric measures variance between various models made by different modelers that model a single case description with a single aspect modeling technique. The lower this value, the more equal the various models are. If a large number of modelers comes up with equal descriptions, it can be presumed that these models are in accordance with the phenomenon that is modeled. Furthermore it can be presumed that the technique is deterministic, leading to unambiguous models that are in accordance with reality. Some remarkable CSV values will be discussed and presumptions about the reasons for these values will be given.

Remarkable are the high CSV values (>=0.6) for the EPC Process View, the PBM Business Process Model and the UBM Low level Process model. As was discussed in section 8.3, all three techniques lack a precise criterion for identifying business transitions. The lack of a precise criterion for identifying transitions, be it EPC functions, PBM transitions or UBM activities, is presumed to influence the variance in the resulting models. The DEMO Business Architecture Model, also capturing business transitions by means of the transaction concept has a lower CSV (0.36). One could argue that this CSV is lower because DEMO has a more precise criterion for identifying transactions, however, the transaction concept cannot be compared to the previously mentioned techniques since also the fact that it is more course grained (containing a complete pattern of predefined transitions as opposed to e.g. the PBM transition concept which represents only one transition) is a factor that might influence the CSV value.

Taking a look at the data view, the results are interesting. The CSV of the Entity Type in the EPC Data View is the lowest of all measured CSV values: 0.29, the CSV
of the Class in the UBM Conceptual Model is one of the highest values measured: 0.62. The high CSV of the Class can be explained by means of the many coherency relationships that the UBM conceptual model has (AMCI0CONCEPTUALMODEL=1.0) in combination with the fact that the construction of a conceptual model is the last step in the method (see Figure 7-35). The Conceptual Model is constructed subsequent to all other models and the modeler is free to choose classes from the set of processes, goals, resources that have been modeled in previous steps. This freedom results inevitably in high variance amongst different modelers. In EPC’s, the construction of a data view is the second step in the method. Due to the strictness of the coherency relationship between process view and data view, the EPC entities in the Data View are not chosen freely, the set of entity types must be equal to the entity types modeled during the previous step, the construction of the process view. It is presumable that the CSV value measured for a certain concept that is part of a modeling step that is preceded by a modeling step in combination with coherency relationships that restrict modelers in their modeling freedom, will turn out lower.

Critical Remarks On the Generalizability of the Results
The application of the proposed metrics and the investigation about the predictive value of these metrics regarding errors made by modelers in an empirical study lead to the following remarks on the extent to which the results are significant and can be generalized.

First of all, empirical studies by means of modeling sessions suffer from what I would like to call the experts paradox. To get reasonable results, the population should consist of modeling experts. However, modeling experts are familiar with and can be biased to some of the techniques that are studied, which influences the results. In the experiment conducted above, the population consisted of graduate students that were at least familiar with DEMO and parts of UBM. To guarantee more objectivity, each of the four techniques was introduced with a one hour lecture, and during the modeling session, the population made use of a description of the technique that was set up equally for each technique. As said, another factor that might have influenced the outcome is the order in which modeling techniques were discussed. It also turned out that the examples given during the one-hour lecture influenced the outcome. Some modelers prefer to start with the examples and change these to fit the case at hand rather than to start a model from scratch.

8.5 Conclusions
This chapter dealt with the application of the theoretical and inquiring metrics introduced in chapter 6. Metrics were applied both to study the correctness of
models as well as their usefulness. An empirical study was conducted to acquire support for the predictive value of the theoretical measures in terms of error rates.

It can be concluded that the theoretical measurements that are based on the C-Me descriptions of techniques can be carried out straightforward, without problems. No research was carried out regarding the application of the inquiring metrics under a population of experts. E.g. an example of an AMI metric form was given rather than the result of an inquiry amongst a representative population. It is however not to be expected that the implementation of such an inquiry would yield serious problems.

The inquiring metrics concerning usefulness covered system expressiveness and business suitability. The application of the discrete dynamic systems paradigm provided a generic framework well suited for the comparison of the four modeling techniques. The system expressiveness evaluation lays bare two important characteristics of a modeling technique. On the one hand it shows deficits in the technique (such as the concept of an actor in UBM) and on the other hand it gives an account of how exactly lawful state space and transitions are modeled by the respective modeling techniques. The business suitability evaluation lays bare the different system categories that are modeled by the modeling techniques and their interrelationships. It lays bare a clear distinction between techniques that do not pay attention to the nature of the systems modeled and techniques that do (e.g. EPC as opposed to DEMO). Furthermore it shows differences with regard to how different perspectives within a modeling technique interrelate (e.g. the EPC data view and process view are more loosely coupled than other techniques; furthermore, the nature of the coupling is different than e.g. the UBM coupling between conceptual model and process model).

The empirical study carried out has shortcomings and must not be interpreted as an ideal experiment. Some shortcomings are inherent to the carrying out of modeling sessions, such as the experts paradox mentioned in this chapter. Although in many cases it did not provide full evidence for correlation between theoretical metric and empirical result, it did offer insight in the interpretation of the theoretical metrics. In future research, other ways of gathering empirical data have to be considered. This has special importance since empirical research, as an important source of knowledge to provide evidence for research hypotheses, is carried out far too little in the business modeling, information systems and software engineering disciplines.
9

CONCLUSIONS AND DIRECTIONS
FOR FURTHER RESEARCH

9.1 Introduction

In this thesis I have addressed a number of research questions concerning the evaluation of techniques for modeling business processes. I have answered these questions by proposing a framework for evaluation and justified the answers by applying the framework and validating the results. Underlying presumption with regard to the integrated framework that was proposed is that the quality of a certain artifact can be distinguished into a number of quality properties, and that each of these properties can be made measurable by assigning metrics to that property.

9.2 Answers to Research Questions

Revisiting chapter 1, the first question to be answered concerned what was to be understood by a conceptual modeling technique. This question was answered conclusively in chapter 2, systematically introducing the terms model, conceptual modeling and modeling technique. Especially contemporary multi modeling techniques were discussed, techniques that encompass a coherent whole of different techniques for modeling different aspects of a certain phenomenon.

The second research question concerned the phenomenon to be modeled, the business process. In section 2.4, the conceptual modeling of business processes was introduced. Later on, in section 4.5, I have introduced an ontology for modeling business processes. This is an ontological view that existing modeling technique can be compared to in order to investigate differences in their suitability for modeling business processes.

The notion of quality, reflecting research question 3 was discussed extensively in chapter 4. In this chapter the quality of models as well as the quality of modeling techniques was brought up. The underlying presumption here was that quality can
be differentiated into a number of quality properties that can in their turn be quantified by means of metrics that are assigned to properties. The resulting quality framework will be discussed in more detail later on.

Business process modeling techniques, subject of the fourth research question, were treated in particular in section 2.5. Examples of both existing tools as well as techniques have been given.

Chapter 4, 5 and 6 dealt with the main proposition of this thesis, an integrated framework for evaluating business process modeling techniques. As such, chapter 5 provides the answer to research question 5, how modeling techniques can be described to make quality measurable. In this chapter, I have proposed the C-Me description language. Later on, quality measurement was anchored in this description language. The main proposition of the thesis, the Q-Me framework for evaluation, answers the final research question: How can the quality of business process modeling techniques be evaluated. In the next section I will proceed by discussing this framework in more detail.

9.3 Q-Me: An Integrated Approach

The proposition in this thesis consisted of three parts: Chapter 4 concerns the introduction of the notion of quality and further subdivision of this notion by a number of quality properties for models followed by a number of quality properties for techniques presumed to be of influence on the quality of models. In chapter 5, I have proposed a description language for capturing state of the art multi modeling techniques. In chapter 6, I have combined the notion of quality that was developed before with the description language to propose a number of measurements quantifying the proposed properties. I would like to review these three parts of the proposition in more detail.

Based on existing literature and flaws and shortcomings in existing quality frameworks, I have taken two main characteristics of a good model as a starting point: a good model should be correct and it should be useful for the purpose at hand, the modeling of business processes.

The correctness property was further differentiated into an internal and external component. Both components are used to determine the correctness of elementary propositions (individual knowledge primitives and concepts), the elements out of which aspect models are built, the aspect models themselves and relationships between aspect models in a whole model. With regard to the internal correctness of the propositions in a model, a further differentiation has been made into empirical, syntactical and semantic correctness, in line with existing literature on semiotics and linguistics. External correctness concerns the validity of a model, it being in
accordance with the phenomenon that is modeled. Several properties of a modeling technique were identified presumed to influence the correctness of resulting models: notational intuitiveness, syntactic freedom, coherence and determinism.

The usefulness of a model is determined by the generic knowledge primitives and concepts offered by the technique. These should on the one hand be expressive enough to be able to model business systems according to the discrete dynamic systems paradigm and on the other hand suitable for modeling business processes, i.e. be explicit about the system categories and consequently kind of transitions, states and actors to distinguish when modeling business processes.

In order to capture modeling techniques, the C-Me description language is proposed. The C-Me language is a meta modeling language that captures the generic concepts and knowledge primitives used by a technique. A unique research contribution is the ability to describe different aspect modeling techniques and their interrelationships as well as the modeling process, the order in which aspect modeling techniques are applied within a multi modeling technique.

The proposed metric scheme provides measures to the quality properties that were identified earlier. The formal metrics are anchored in the formalization of the C-Me description language so that calculation of values can easily be automated.

Together, the identified quality properties, the C-Me description language and the measurement scheme constitute a novel, integrated approach for describing and evaluating modeling: The Q-Me framework. This is a significant research contribution since current frameworks lack an approach that integrates the above-mentioned components.

9.4 Application and Validation

Chapter 7 dealt with the application of the proposed description language to four example modeling techniques. These techniques were chosen and adjusted to form a reflection of commonly used state of the art modeling techniques. This application proved the language to be well suited for capturing aspect modeling techniques and their interrelationships. Some modeling issues that emerged during the application, such as distinguishing concepts and knowledge primitives and the modeling of complex coherency relationships, were critically assessed in the conclusions of this chapter.

Chapter 8 concerned the application of the measurement scheme, both for those properties assuring the correctness of models as well as the usefulness of models, and an empirical study to provide support offer insight in the interpretation of the metrics.
Formal metrics were calculated straightforward out of the description language and experimental metrics were calculated by carrying out the prescribed experiments. The inquiring metrics were carried out by means of an example application. It would have been better to actually carry out the inquiries amongst a population of experts.

As indicated, the empirical study suffered from shortcomings, thereby limiting the reliability and generalizability of the results. In some cases, the results showed that the quality properties indeed influence the quality of the resulting models, in other cases, the results did not show the expected correlation. However, the study did offer more insight in the interpretation of the proposed metrics. The main conclusion that can be drawn is that the properties and metrics are an important means to better understand modeling techniques, but are in itself not reliable enough to obtain a full understanding. Understanding modeling techniques requires the interpretation of metric values. The interpretation of modeling techniques by means of the proposed measures can best be compared with the evaluation of the performance of a company by means of its liquidity, profitability and solvability ratio's. They provide an indication of some aspect of the company and require further interpretation for a good understanding.

9.5 Guidelines for Qualitative Modeling Techniques

Notational intuitiveness

The notational intuitiveness concerns the notational of the various concepts and knowledge primitives used in a modeling technique. It is presumed to influence the error rate in resulting models.

- The empirical study shows that a modeling technique with a notation that is in line with past practice and uses basic shapes to distinguish different concepts results in less mistakes being made in the construction of models.

An example of a contra-intuitive notation was given in chapter 4 (Figure 4-12).

Syntactic Freedom

It is clear that a modeling technique that does not restrict the modeler at all with regard to syntax, i.e. the modeler is free to draw anything he or she wants, results in models without any syntactical errors. However, to recommend to design modeling techniques in such a way that they do not restrict the modeler at all would be too shortsighted. Restrictions are naturally imposed due to the nature of the phenomena that are modeled. For example, a modeling technique that models
functional decompositions (as the EPC function view does) naturally restricts that a
certain sub function can only belong to precisely one super function, since this is the
nature of decomposition. It is of course out of the question to drop such a natural
restriction in favor of models with less syntactical errors. This leads to the following
guidelines:

- Critically assess each uniqueness and requiredness constraint imposed on a
  knowledge primitive to check its necessity, i.e. whether the constraint is
  naturally imposed due to the character of the phenomenon modeled. The AUC
  and ARQ metrics are indicative for the number of constraints.

- Provide those constrained knowledge primitives with a suitable notation that
  forces the modeler to obey the constraint (c.f. Figure 8-13).

**Coherency**

Coherency between aspect modeling techniques is the source of inconsistencies
between resulting aspect models. Together, the aspect modeling techniques
constitute a coherent multi modeling technique and therefore, coherency
relationships are by definition part of a multi modeling technique.

- Critically access coherency relationships on their necessity, i.e. ask the question
  whether the aspect modeling techniques still form a coherent whole without a
  certain coherency relationship. The proposed AMCI1 metric is indicative here.

- Critically access constraints on knowledge primitives involved in coherency
  relationships on their necessity, i.e. whether the constraint is naturally imposed
  due to the character of the phenomenon modeled. The AMCIUC and AMCINO
  metrics are indicative here.

- Minimize conceptual overlap between two aspect modeling techniques. For
  complex coherency relationships, consisting of structures of concepts and
  knowledge primitives, minimize the conceptual structure. The proposed ACC
  measure is indicative for this complexity.

**Determinism**

The experimental CSV metric measures the variance in the outcome of the
application of techniques. Its application resulted in a number of findings that can
be translated to the following guidelines:

- Provide clear definitions that allow the unique identification of instances of
  concepts and knowledge primitives, e.g. the high CSV for the EPC process view
  is probably caused by a lack of definition what actually an EPC function is.
• For each precedence step in the modeling process (the construction of one aspect model following the construction of another) make sure that the coherency relationships are restrictive. E.g. the high CSV value for the UBM conceptual model is the result of the fact that the conceptual model is constructed after all other models and might include processes, resources from previous models as classes.

**Complexity**

It is obvious that the less complex a technique is, the less generic concepts and knowledge primitives the modeler needs to learn before the technique can be applied. This leads to the following recommendation:

• Keep the technique as simple as possible; minimize the number of generic concepts and knowledge primitives within the aspect modeling techniques.

**Expressiveness**

Most existing modeling techniques model business processes as discrete dynamic systems. Therefore this paradigm has been chosen as the paradigm that a modeling technique must comply to. The guideline for expressiveness is simply a repetition of what the WSE metric proposes:

• Make sure that the modeling technique is capable of modeling discrete dynamic systems, their lawful state space, transitions, and actors causing transitions and inspecting the state space.

**Suitability**

The modeling of discrete dynamic systems is itself not enough to guarantee that a technique is suitable for modeling business processes given a certain purpose. As mentioned before, the discrete dynamic systems paradigm can be applied to model a variety of system categories ranging from mental states and transitions and traffic light systems to the building of a house. More differentiation is required. A modeling technique should clearly specify what system categories are modeled and how to distinguish states, transitions and actors. The modeling techniques that were studied are mostly applied to align Business and IT systems and therefore, not surprisingly, the systems that are modeled are the business system and information system. The definition of what a transition is in a business system is in many cases open for debate. The following guidelines can be mentioned:

• Make sure that the modeling technique makes a clear distinction with regard to the kind of discrete dynamic systems that are modeled. The technique should clearly specify what kind of transitions, states and actors are modeled.
- If the modeling technique covers more than one system category, the systems should be interrelated with each other in a consistent way. Compare e.g. the weak relationship between Business System and Information System in the EPC technique with e.g. the PBM technique (section 6.4).

The kind of systems that a modeling technique should be able to model is purpose dependent. In this thesis I mainly focused on techniques that model business systems and information systems and their interrelationship. But, for example, when techniques are applied for the purpose of e.g. Activity Based Costing (ABC) then the technique should also cover the financial system, et cetera. Therefore,

- The kinds of systems that a modeling technique must be able to model should be determined given the purpose at hand.

**Concluding Remarks**

An important conclusion that can be drawn is that improving the quality of modeling techniques requires more than the optimization of the individual properties mentioned above. The quality properties mentioned above influence each other in various ways. An optimal balance must be found, depending on external factors such as the purpose for which the modeling technique is used.

Business process modeling techniques offer generic concepts and factual knowledge that a modeler can use to conceptualize things in reality as business processes. An individual model consists of individual concepts and factual knowledge about this reality. A modeling language prescribes how concepts can be combined to form correct expressions. It restricts the modeler for example by prescribing which concepts can be combined in a certain expression or which expressions contradict each other. On the one hand, we want modeling languages to be as expressive as possible so that we can express everything that might occur in reality according to the underlying theory. In the ideal situation, the modeling technique allows us to use concepts any way we like (a maximal syntactic freedom and no constraints) resulting in models that cannot have any syntactical errors. On the other hand modeling languages should offer restrictions so that we cannot model what will not occur in reality according to the underlying theory. If an object A weighs more than an object B, it cannot be the case that at the same time object B weighs more than object A. The underlying theory should restrict us to model this inconsistency.

A balance between this freedom and restrictions is hard to determine. A violated restriction in a model that would typically be recorded as an error in the empirical study in chapter 8, can also be an indication that the underlying theory is not correct, a falsification, something occurring in reality that a modeling technique restricts us to model.
Each generic concept should be provided with a clear meaning. This also holds for the various ways in which concepts can be combined. Precondition is that a modeling technique is semantic accurate. There are two dimensions to semantic accuracy. On the one hand, concepts should have a meaning that is understood and shared by the stakeholders in a modeling project. Stakeholders should agree on what concepts like activity, decision and responsibility mean in order to use them properly in models. The other dimension is that models should have a formal meaning that can be mapped on computer systems that execute these models. The modeling technique should have a formal semantics. The system expressiveness evaluation shows in more detail how constructs can be mapped to the discrete systems paradigm. The degree to which these dimensions are important depends very much on the purpose for which techniques are used.

At the end, an evaluation of techniques depends very much on the underlying paradigm that is chosen. The paradigm chosen in the Q-Me framework was chosen broad enough to encompass all evaluated techniques so that differences between these techniques became visible.

9.6 Directions for Further Research

In some respects the research that has been carried out is incomplete and shortcomings can be thought of. In order to extend the research that has been conducted and to overcome the shortcomings, I would like to conclude this thesis with the proposal of the following research agenda concerning the evaluation of business process modeling techniques:

- A more detailed study should be carried out concerning the purpose dependency of techniques. As said, the kind of systems that a modeling technique should be able to model (business suitability) must be determined according to the purpose at hand. For each purpose mentioned in Figure 4-2 and possibly other purposes, it should be determined what systems should be covered.

- In this thesis, the C-Me description language has been proposed as a means to capture modeling languages for the purpose of evaluation. However, the C-Me language can be used separately from the Q-Me framework as a specification language for modeling techniques that is presumed to be suitable for tool development. This presumption must be further investigated.

- The inquiring metrics should be tested by means of more applications amongst a representative population of experts.

- Although the C-Me language allows the description of both modeling techniques and methods (defined as sequence of modeling activities in time), the proposed
metrics deal in particular with the modeling techniques. Metrics that offer insights in the application of methods such as the efficiency and effectiveness of the method can be developed in the future.

- For too few empirical studies are carried in the Information Systems discipline, more emphasis on this type of research is necessary. Instead of the empirical study mentioned earlier, one should think of sound experiments to provide evidence in support of the assumptions and hypotheses.

- The Q-Me framework is not limited to the evaluation of business process modeling techniques and can also be applied in other domains if the system expressiveness and business suitability are adjusted to these domains. A relevant future application is to apply the Q-Me framework for the evaluation of software specification techniques such as UML.

- Case studies should be carried out in order to gain more experience with regard to the application of the Q-Me framework in practice.

The obtained results published in this thesis show that it is possible to describe contemporary business process modeling techniques in a formal framework and to measure relevant properties of those techniques. Frameworks such as the one developed in this thesis are useful and will probably become even more useful in the future as a tool to distinguish qualitative and less qualitative business process modeling techniques that are rapidly emerging in both commercial as well as scientific domains.
LITERATURE REFERENCES


Kroes, P. (1996). Ideaalbeelden van wetenschap; een inleiding in de wetenschapsfilosofie, Boom, Meppel;


Leeuw, A.C.J. de (1990). Een boekje over bedrijfskundige methodologie: management van onderzoek, Van Gorcum, Assen;


Morris, C.W. (1955), Sings, Language and Behavior, George Braziller, New York;


Ogden, C. K., Richards, I.A. (1923), The meaning of meaning, Harcourt, Brace & World, New-York;


Smolander, K., (1990). OPPR - A model for modeling systems development methods, University of Jyväskylä, Jyväskylä;


Swanborn, P.G. (1986). Methoden van sociaal wetenschappelijk onderzoek, Boom, Meppel;


Wand, Y. (2000), Presentation given during the CAiSE*00 doctoral consortium, Stockholm;


Appendix A: Detailed Technique Descriptions

This appendix shows in detail the four modeling technique described in chapter 7, using the C-Me formalism described in chapter 5.

Event Driven Process Chains

Way of Modeling

S := (PROCESSVIEW, FUNCTIONVIEW, ORGANIZATIONALVIEW, DATAVIEW)

C := (CONNECTOR, FUNCTION, EVENT, JOIN, SPLIT, ORSPLIT, ANDSPLIT, XORSPLIT, ORJOIN, ANDJOIN, XORJOIN, POSITION, APPLICATIONSYSTEM, ENTITYTYPE, ORGUNIT, GENERALIZATION, MULTIPLICITYVALUE, RELATIONSHIPTYPE, ROLE)

K := (PRECEDES, Performs, SUPPORTS, reads, UPDATES, ISSUBFUNCTIONOF, SUPERVIZES, IN, ISSUPERTYPEIN, ISSUBTYPEIN, ISMINOF, ISMAXOF, HAS, PLAYS, ISDISJUNCT, ISTOTAL)

r := (PRECEDES, (CONNECTOR, 1), (CONNECTOR, 2), PERFORMS, (POSITION, 1), (FUNCTION, 2), SUPPORTS, (APPLICATIONSYSTEM, 1), (FUNCTION, 2), READS, (FUNCTION, 1), (ENTITYTYPE, 2), UPDATES, (FUNCTION, 1), (ENTITYTYPE, 2), ISSUBFUNCTIONOF, (FUNCTION, 1), (FUNCTION, 2), SUPERVIZES, (ORGUNIT, 1), (ORGUNIT, 2), IN, (POSITION, 1), (ORGUNIT, 2), ISSUPERTYPEIN, (ENTITYTYPE, 1), (GENERALIZATION, 2), ISSUBTYPEIN, (ENTITYTYPE, 1), (GENERALIZATION, 2), ISMINOF, (MULTIPLICITYVALUE, 1), (ROLE, 2), ISMAXOF, (MULTIPLICITYVALUE, 1), (ROLE, 2), HAS, (RELATIONSHIPTYPE, 1), (ROLE, 2), PLAYS, (ENTITYTYPE, 1), (ROLE, 2), ISDISJUNCT, (GENERALIZATION, 1), ISTOTAL, (GENERALIZATION, 1))

o := 0

G := (CONNECTOR, FUNCTION, CONNECTOR, EVENT, CONNECTOR, JOIN, CONNECTOR, SPLIT, ORSPLIT, ANDSPLIT, XORSPLIT, JOIN, ORJOIN, JOIN, ANDJOIN, JOIN, XORJOIN)

A := (CONNECTOR, PROCESSVIEW, FUNCTION, PROCESSVIEW, EVENT, PROCESSVIEW, JOIN, PROCESSVIEW, SPLIT, PROCESSVIEW, ORSPLIT, PROCESSVIEW, ANDSPLIT, PROCESSVIEW, XORSPLIT, PROCESSVIEW, ORJOIN, PROCESSVIEW, ANDJOIN, PROCESSVIEW, XORJOIN, PROCESSVIEW, POSITION, PROCESSVIEW, APPLICATIONSYSTEM, PROCESSVIEW, ENTITYTYPE, PROCESSVIEW, PRECEDES, PROCESSVIEW, PERFORMS, PROCESSVIEW, SUPPORTS, PROCESSVIEW, READS, PROCESSVIEW, UPDATES, PROCESSVIEW, FUNCTION, FUNCTIONVIEW, ISSUBFUNCTIONOF, FUNCTIONVIEW, POSITION, ORGANIZATIONALVIEW, ORGUNIT, ORGANIZATIONALVIEW, SUPERVIZES, ORGANIZATIONALVIEW, IN, ORGANIZATIONALVIEW, ENTITYTYPE, DATAVIEW, GENERALIZATION, DATAVIEW, MULTIPLICITYVALUE, DATAVIEW, RELATIONSHIPTYPE, DATAVIEW, ROLE, DATAVIEW, ISSUPERTYPEIN, DATAVIEW, ISSUBTYPEIN, DATAVIEW, ISMINOF, DATAVIEW, ISMAXOF, DATAVIEW, HAS, DATAVIEW, PLAYS, DATAVIEW, ISDISJUNCT, DATAVIEW, ISTOTAL, DATAVIEW)

O := (PROCESSVIEW, (FUNCTION), FUNCTIONVIEW, (FUNCTION), PROCESSVIEW, (POSITION), ORGANIZATIONALVIEW, (POSITION), PROCESSVIEW, (ENTITYTYPE), DATAVIEW, (ENTITYTYPE))

<S, C, K, r, o, G, A, O> := (PROCESSVIEW, FUNCTIONVIEW, ORGANIZATIONALVIEW, DATAVIEW), (CONNECTOR, FUNCTION, EVENT, JOIN, SPLIT, ORSPLIT, ANDSPLIT, XORSPLIT, ORJOIN, ANDJOIN, XORJOIN, POSITION, APPLICATIONSYSTEM, ENTITYTYPE, ORGUNIT, GENERALIZATION, MULTIPLICITYVALUE, RELATIONSHIPTYPE, ROLE), (PRECEDES, PERFORMS, SUPPORTS, READS, UPDATES, ISSUBFUNCTIONOF, SUPERVIZES, IN, ISSUPERTYPEIN, ISSUBTYPEIN, ISMINOF, ISMAXOF, HAS, PLAYS, ISDISJUNCT, ISTOTAL), (PRECEDES, (CONNECTOR, 1), (CONNECTOR, 2)), PERFORMS, (POSITION, 1), (FUNCTION, 2), SUPPORTS, (APPLICATIONSYSTEM, 1), (FUNCTION, 2), READS, (FUNCTION, 1), (ENTITYTYPE, 2), UPDATES, (FUNCTION, 1), (ENTITYTYPE, 2), ISSUBFUNCTIONOF, (FUNCTION, 1), (FUNCTION, 2), SUPERVIZES, (ORGUNIT, 1), (ORGUNIT, 2), IN, (POSITION, 1), (ORGUNIT, 2), ISSUPERTYPEIN, (ENTITYTYPE, 1), (GENERALIZATION, 2), ISSUBTYPEIN, (ENTITYTYPE, 1), (GENERALIZATION, 2), ISMINOF, (MULTIPLICITYVALUE, 1), (ROLE, 2), ISMAXOF, (MULTIPLICITYVALUE, 1), (ROLE, 2), HAS, (RELATIONSHIPTYPE, 1), (ROLE, 2), PLAYS, (ENTITYTYPE, 1), (ROLE, 2), ISDISJUNCT, (GENERALIZATION, 1), ISTOTAL, (GENERALIZATION, 1)>.
FUNCTION, <CONNECTOR, EVENT>, <CONNECTOR, JOIN>, <CONNECTOR, SPLIT>, <SPLIT, ORSPLIT>, <SPLIT, ANDSPLIT>, <SPLIT, XORSPLIT>, <JOIN, ORJOIN>, <JOIN, ANDJOIN>, <JOIN, XORJOIN>, <CONNECTOR, PROCESSVIEW>, <FUNCTION, PROCESSVIEW>, <EVENT, PROCESSVIEW>, <JOIN, PROCESSVIEW>, <SPLIT, PROCESSVIEW>, <ORSPLIT, PROCESSVIEW>, <ANDSPLIT, PROCESSVIEW>, <XORSPLIT, PROCESSVIEW>, <ORJOIN, PROCESSVIEW>, <ANDJOIN, PROCESSVIEW>, <XORJOIN, PROCESSVIEW>, <POSITION, PROCESSVIEW>, <APPLICATIONSYSTEM, PROCESSVIEW>, <ENTITYTYPE, PROCESSVIEW>, <PREcedes, PROCESSVIEW>, <PERFORMS, PROCESSVIEW>, <SUPPORTS, PROCESSVIEW>, <READS, PROCESSVIEW>, <UPDATES, PROCESSVIEW>, <FUNCTION, FUNCTIONVIEW>, <ISSUBFUNCTIONOF, FUNCTIONVIEW>, <POSITION, ORGANIZATIONALVIEW>, <ORGUNIT, ORGANIZATIONALVIEW>, <SUPERVISES, ORGANIZATIONALVIEW>, <IN, ORGANIZATIONALVIEW>, <ENTITYTYPE, DATAVIEW>, <LEVEL, DATAVIEW>, <MULTICITYVALUE, DATAVIEW>, <RELATIONSHIPTYPE, DATAVIEW>, <ROLE, DATAVIEW>, <ISSUPERTYPEOF, DATAVIEW>, <ISSUBTYPEOF, DATAVIEW>, <ISMINOF, DATAVIEW>, <ISMAXOF, DATAVIEW>, <HAS, DATAVIEW>, <PLAYS, DATAVIEW>, <ISDISJUNCT, DATAVIEW>, <ISTOTAL, DATAVIEW>, <PROCESSVIEW, FUNCTIONVIEW>, <FUNCTION, FUNCTIONVIEW>, <PROCESSVIEW, [POSITION], ORGANIZATIONALVIEW, [POSITION], <PROCESSVIEW, [ENTITYTYPE], DATAVIEW, [ENTITYTYPE]>

Type of Coherency Relationships

c = [[[PROCESSVIEW, (FUNCTION), FUNCTIONVIEW, (FUNCTION)], EQUAL], [[[PROCESSVIEW, (POSITION), ORGANIZATIONALVIEW, (POSITION)], EQUAL], [[[PROCESSVIEW, (ENTITYTYPE), DATAVIEW, (ENTITYTYPE)], EQUAL]]

Constraints on the way of modeling

UC = [PREcedes, (1), [PREcedes, (2)], [PERFORMS, (2)], [ISSUBFUNCTIONOF, (1)], [SUPERVISES, (2)], [IN, (1)], [ISSUPERTYPEOF, (2)], [ISMINOF, (2)], [ISMAXOF, (2)], [HAS, (2)], [PLAYS, (2)]

RO = [PERFORMS, (1), [PERFORMS, (2)], [SUPPORTS, (1), [READS, (2)], [UPDATES, (2)], [IN, (1)], [ISSUPERTYPEOF, (2)], [ISMINOF, (2)], [ISMAXOF, (2)], [HAS, (2)], [PLAYS, (2)]

Way of Working

T = [(CONSTRUCT_PROCESSVIEW, CONSTRUCT_FUNCTIONVIEW, CONSTRUCT_ORGANIZATIONALVIEW, CONSTRUCT_DATAVIEW)]

P = [(CONSTRUCT_PROCESSVIEW, CONSTRUCT_ORGANIZATIONALVIEW), (CONSTRUCT_PROCESSVIEW, CONSTRUCT_DATAVIEW), (CONSTRUCT_PROCESSVIEW, CONSTRUCT_FUNCTIONVIEW)]

f = (CONSTRUCT_PROCESSVIEW)

< T, P, f > = [(CONSTRUCT_PROCESSVIEW, CONSTRUCT_FUNCTIONVIEW, CONSTRUCT_ORGANIZATIONALVIEW, CONSTRUCT_DATAVIEW), (CONSTRUCT_PROCESSVIEW, CONSTRUCT_ORGANIZATIONALVIEW), (CONSTRUCT_PROCESSVIEW, CONSTRUCT_DATAVIEW), (CONSTRUCT_PROCESSVIEW, CONSTRUCT_FUNCTIONVIEW)]

Mapping Way of Working to Way of Modeling

f = [(CONSTRUCT_PROCESSVIEW, PROCESSVIEW), (CONSTRUCT_FUNCTIONVIEW, FUNCTIONVIEW), (CONSTRUCT_ORGANIZATIONALVIEW, ORGANIZATIONALVIEW), (CONSTRUCT_DATAVIEW, DATAVIEW)]

End Definition

Dynamic Essential Modeling of Organizations

Way of Modeling

S = (BUSINESSARCHITECTUREMODEL, PHASEMODELMODEL, TRANSACTIONRESULTTABLE, BANKFACTTABLE)

C = (CONVERSATION, FACTCONTAINER, PERSUASION, AUTHORIZATION, TRANSACTION, BANK, ACTOR, AGGREGATE_ACTOR, ELEMENT_ACTOR, AGGREGATED_FUNCTION, EVENT, PROCESSVIEW, JOIN, PROCESSVIEW, CONSTRUCT_FUNCTIONVIEW, PHASE, OBJECTTYPE, FACT)

K = (INITIATOROF, RESPONDEROF, INSPECTED, CONSISTSOF, CONDITIONALWAIT, HAS, INITIATEDURING, INITIATEDEXTERNALLY, REFERSTO, SPECIFIERSRESULTOF, ISCONTAINEDIN)

r = (INITIATOROF, (ACTOR, 1), CONVERSATION, 2), (RESPONDEROF, (ACTOR, 1), CONVERSATION, 2),
Evaluation of Business Process Modeling Techniques

(BANK), TRANSACTIONRESULTTABLE, OBJECTTYPE, BANKFACTTABLE, OBJECTTYPE>, TRANSACTIONRESULTTABLE, TRANSACTION, PROCESSPHASEMODEL, TRANSACTION>, PROCESSPHASEMODEL, TRANSACTION, BUSINESSARCHITECTUREMODEL, TRANSACTION, CONVERSATION, PERSUASION, AUTHORIZATION, TRANSACTION>, PROCESSPHASEMODEL, PHASE, CONVERSATION, INITIATEDURING, INITIATEDEXTERNALLY, BUSINESSARCHITECTUREMODEL, CONVERSATION, ACTOR, AGGREGATEDACTOR, ELEMENTARYACTOR, INITIATOREOF, RESPONDEROF>}

Type of Coherency Relationships

c = {< TRANSACTIONRESULTTABLE, TRANSACTION, BUSINESSARCHITECTUREMODEL, TRANSACTION>, EQUAL, BANKFACTTABLE, BANK, BUSINESSARCHITECTUREMODEL, BANK, TRANSACTIONRESULTTABLE, OBJECTTYPE, BANKFACTTABLE, OBJECTTYPE, OVERLAP, TRANSACTIONRESULTTABLE, TRANSACTION, PROCESSPHASEMODEL, TRANSACTION>, EQUAL, PROCESSPHASEMODEL, CONVERSATION, PERSUASION AUTHORIZATION, TRANSACTION, BUSINESSARCHITECTUREMODEL, CONVERSATION, PERSUASION, AUTHORIZATION, TRANSACTION>, EQUAL, PROCESSPHASEMODEL, PHASE, CONVERSATION, INITIATEDURING, INITIATEDEXTERNALLY, BUSINESSARCHITECTUREMODEL, CONVERSATION, ACTOR, AGGREGATEDACTOR, ELEMENTARYACTOR, INITIATOREOF, RESPONDEROF>, EQUAL}

Constraints on the way of modeling

CU = {RESPONDEROF, (1), RESPONDEROF, (2), CONSISISTOF, (2), HAS, (2), INITIATEDURING, (1), SPECIESRESULTOF, (1), SPECIESRESULTOF, (2)}
RQ = {INITIATOREOF, (2), RESPONDEROF, (1), RESPONDEROF, (2), INSPECTED, (2), CONSISISTOF, (2), HAS, (1), HAS, (2), REFERSTO, (1), REFERSTO, (2), SPECIESRESULTOF, (2), ISCONTAINEDIN, (2)}

Way of Working

T = {CONSTRUCT BUSINESSARCHITECTUREMODEL, CONSTRUCT PROCESSPHASEMODEL, CONSTRUCT TRANSACTIONRESULTTABLE, CONSTRUCT BANKFACTTABLE}

P = {}
I = {}

< T, P, I > = {CONSTRUCT BUSINESSARCHITECTUREMODEL, CONSTRUCT PROCESSPHASEMODEL, CONSTRUCT TRANSACTIONRESULTTABLE, CONSTRUCT BANKFACTTABLE, 0, (1)>

Mapping Way of Working to Way of Modeling

s = {CONSTRUCT BUSINESSARCHITECTUREMODEL, BUSINESSARCHITECTUREMODEL, CONSTRUCT PROCESSPHASEMODEL, PROCESSPHASEMODEL, CONSTRUCT TRANSACTIONRESULTTABLE, TRANSACTIONRESULTTABLE, BANKFACTTABLE, BANKFACTTABLE}

End Definition

Petri-net Based Modeling

Way of Modeling

S = {BUSINESSCONTROLMODEL, BUSINESSPROCESSMODEL, BUSINESSINFORMATIONMODEL}
C = {ACTOR, FUNCTION, EXTERNALAGENT, RFLOW, OBJECTTYPE, ACTIVITYTYPE, ACTIONOBJECT, PLACE, TRANSITION, BEGINPLACE, INTERMEDIATEPLACE, ENDPLACE, SUBPROCESS, TYPE, LABELTYPE, ROLE, UNICITYCONSTRAINT, FACTTYPE}
K = {ISSUBFUNCTIONOF, INITIATES, RESPONDESTO, ACTIONOBJECT, ISPOSTCONDITIONOF, PERFORMS, ISOUTPUTOF, ININPUTOF, PLACEPARTOF, TRANSITIONPARTOF, REPRESENTS, GENERALIZATION, PLAYS, (TYPE, ROLE, CONTAINS)}

r = {ISSUBFUNCTIONOF, (FUNCTION, 1), (FUNCTION, 2), INITIATES, (ACTOR, 1), (RFLOW, 2), RESPONDESTO, (ACTOR, 1), (RFLOW, 2), ACTIONOBJECT, (ACTIVITYTYPE, 1), (OBJECTTYPE, 2), ISPOSTCONDITIONOF, ACTIONOBJECT, (PLACE, 2), PERFORMS, (ACTIONOBJECT, 1), TRANSITION, (TRANSITION, 2), ISOUTPUTOF, (PLACE, 1), TRANSITION, (TRANSITION, 2), ININPUTOF, (PLACE, 1), TRANSITION, (TRANSITION, 2), PLACEPARTOF, (PLACE, 1), RFLow, (2), TRANSITIONPARTOF, (TRANSITION, 1), RFLOW, (2), REPRESENTS, (SUBPROCESS, 1), RFLOW, (2), GENERALIZATION, (OBJECTTYPE, 1), OBJECTTYPE, (2), PLAYS, (TYPE, ROLE, CONTAINS), (UNICITYCONSTRAINT, CONTAINS, (ROLE, 2))

258
Type of Coherency Relationships

c = {
  <BUSINESSCONTROLMODEL, (RFLOOP), BUSINESSPROCESSMODEL, (RFLOOP), EQUAL>,
  <BUSINESSPROCESSMODEL, (OBJECTTYPE), BUSINESSINFORMATIONMODEL, (OBJECTTYPE), SUBSET>
}
Evaluation of Business Process Modeling Techniques

Constraints on the way of modeling


RQ = ("<INITIATES, 2>, <RESPONDSTO, 2>, <ACTIONOBJECT, 1>, <ACTIONOBJECT, 2>, <SPOSTCONDITIONOF, 1>, <PERFORMS, 1>, <PERFORMS, 2>, <ISOUTPUTOF, 2>, <ISINPUTOF, 2>, <PLACEPARTOF, 1>, <PLACEPARTOF, 2>, <TRANSITIONPARTOF, 1>, <TRANSITIONPARTOF, 2>, <REPRESENTS, 1>, <REPRESENTS, 2>, <PLAYS, 2>, <HAS, 1>, <HAS, 2>")

Way of Working

T = ("<CONSTRUCT_BUSINESSCONTROLLMODEL, CONSTRUCT_BUSINESSPROCESSMODEL, CONSTRUCT_INFORMATIONMODEL>")

P = ("<CONSTRUCT_BUSINESSCONTROLLMODEL, CONSTRUCT_BUSINESSPROCESSMODEL, CONSTRUCT_INFORMATIONMODEL>")

I = ("<CONSTRUCT_BUSINESSPROCESSMODEL>")

<y, p, t> = ("<CONSTRUCT_BUSINESSCONTROLLMODEL, CONSTRUCT_BUSINESSPROCESSMODEL, CONSTRUCT_INFORMATIONMODEL>, <CONSTRUCT_BUSINESSCONTROLLMODEL, CONSTRUCT_BUSINESSPROCESSMODEL, CONSTRUCT_INFORMATIONMODEL>, <CONSTRUCT_BUSINESSPROCESSMODEL, CONSTRUCT_INFORMATIONMODEL>, [1]"")

Mapping Way of Working to Way of Modeling

a = ("<CONSTRUCT_BUSINESSCONTROLLMODEL, BUSINESSCONTROLLMODEL>, <CONSTRUCT_BUSINESSPROCESSMODEL, BUSINESSPROCESSMODEL>, <CONSTRUCT_INFORMATIONMODEL, INFORMATIONMODEL>")

End Definition

Business Modeling with UML

Way of Modeling

S = ("PROCESSMODELHIGH, PROCESSMODELLLOW, GOALMODEL, CONCEPTUALMODEL")

C = ("PROCESS, GOAL, RESOURCE, ABSTRACT, PEOPLE, PHYSICAL, INFORMATION, CONNECTOR, ACTIVITY, ELEMENTARYACTIVITY, STATE, ENSTATE, BEGINSTATE, JOIN, ANDJOIN, ORJOIN, SPLIT, ANDSPLIT, ORSPLIT, QUALITATIVEGOAL, QUANTITATIVEGOAL, VALUE, GOALDEPENDENCY, PROBLEM, CLASS, ROLE, MULTICOMPONENT, ASSOCIATION")

K = ("SUPPLIES, CONTROLS, CONSUMES, PRODUCES, HAS, PARTOF, PRECEDES, ISSUBGOALOF, ISDEPENDENTON, CONCERNING, CONCERNINGACHIEVEMENTOF, CURRENTVALUE, TARGETVALUE, ISPARTOF, ISPECIALIZATIONOF, PLAYS, ISINOF, ISMAXOF")

r = ("{SUPPLIES, {ACTIVITY, 1}, {RESOURCE, 2}}, {CONTROLS, {ACTIVITY, 1}, {RESOURCE, 2}}, {CONSUMES, {ACTIVITY, 1}, {RESOURCE, 2}}, {PRODUCES, {ACTIVITY, 1}, {RESOURCE, 2}}, {HAS, {ASSOCIATION, 1}, {ROLE, 2}}, {PARTOF, {CONNECTOR, 1}, {CONNECTOR, 2}}, {PRECEDES, {CONNECTOR, 1}, {CONNECTOR, 2}}, {ISSUBGOALOF, {GOAL, 1}, {GOAL, 2}}, {ISDEPENDENTON, {GOAL, 1}, {GOAL, 2}}, {CONCERNING, {PROBLEM, 1}, {GOALDEPENDENCY, 2}}, {CONCERNINGACHIEVEMENTOF, {PROBLEM, 1}, {GOAL, 2}}, {CURRENTVALUE, {QUANTITATIVEGOAL, 1}, {VALUE, 2}}, {TARGETVALUE, {QUANTITATIVEGOAL, 1}, {VALUE, 2}}, {ISPARTOF, {CLASS, 1}, {CLASS, 2}}, {ISPECIALIZATIONOF, {CLASS, 1}, {CLASS, 2}}, {PLAYS, {CLASS, 1}, {ROLE, 2}}, {ISINOF, {MULTICOMPONENT, 1}, {ROLE, 2}}, {ISMAXOF, {MULTICOMPONENT, 1}, {ROLE, 2}}")

o = ("ISDEPENDENTON, GOALDEPENDENCY")

G = ("{RESOURCE, ABSTRACT}, {RESOURCE, PEOPLE}, {RESOURCE, PHYSICAL}, {RESOURCE, INFORMATION}, {CONNECTOR, ACTIVITY}, {ACTIVITY, ELEMENTARYACTIVITY}, {ACTIVITY, PROCESS}, {CONNECTOR, STATE}")
Evaluation of Business Process Modeling Techniques

<PROBLEM, GOALMODEL>, <ISUBGOALOF, GOALMODEL>, <ISDEPENDENTON, GOALMODEL>, <CONCERNING, GOALMODEL>, <CONCERNINGACHIEVEMENTOF, GOALMODEL>, <CURRENTVALUE, GOALMODEL>, <TARGETVALUE, GOALMODEL>, <CLASS, CONCEPTUALMODEL>, <ROLE, CONCEPTUALMODEL>, <MULTIPLICITY, CONCEPTUALMODEL>, <ASSOCIATION, CONCEPTUALMODEL>, <ISPARTOF, CONCEPTUALMODEL>, <IS_SPECIALIZATIONOF, CONCEPTUALMODEL>, <PLAYS, CONCEPTUALMODEL>, <HAS, CONCEPTUALMODEL>, <IS_MINOF, CONCEPTUALMODEL>, <IS_MAXOF, CONCEPTUALMODEL>, <PROCESSMODELLOW, RESOURCE, ABSTRACT, PEOPLE, PHYSICAL, INFORMATION>, <PROCESSMODELHIGH, RESOURCE, ABSTRACT, PEOPLE, PHYSICAL, INFORMATION>, <PROCESSMODELLOW, ELEMENTARYACTIVITY, ACTIVITY, PROCESS, CONNECTOR, RESOURCE, PARTOF, SUPPLIES, CONTROLS, CONSUMES, PRODUCES>, <PROCESSMODELHIGH, GOAL, GOALMODEL, GOAL>, <PROCESSMODELLOW, (CLASS), GOALMODEL, GOAL>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (PROCESS)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (RESOURCE)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (GOAL)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (CONTROLLER)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (CONTROLS)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (CONSUMES)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELHIGH, (PROCESSES)>, <PROCESSMODELLOW, (CLASS), PROCESSMODELLOW, (HAS)>, <PROCESSMODELLOW, (CLASS), ROLE, ASSOCIATION, PLAYS, HAS>, <PROCESSMODELLOW, (CLASS), ROLE, ASSOCIATION, PLAYS, HAS>, <PROCESSMODELLOW, (CLASS), ROLE, ASSOCIATION, PLAYS, HAS>

Type of Coherency Relationships


Constraints on the way of modeling

UC = {<HAS, 1>, <HAS, 2>, <PRECEDES, 1>, <PRECEDES, 2>, <ISSUBGOALOF, 1>, <CURRENTVALUE, 1>, <TARGETVALUE, 1>, <ISPARTOF, 1>, <PLAYS, 2>, <IS_MINOF, 2>, <IS_MAXOF, 2>}

RQ = {<HAS, 1>, <HAS, 2>, <PLAYS, 2>, <IS_MINOF, 2>, <IS_MAXOF, 2>}

Way of Working

T = (CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_GOALMODEL, CONSTRUCT_CONCEPTUALMODEL)

P = {(CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_GOALMODEL), (CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_GOALMODEL, CONSTRUCT_GOALMODEL), (CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_PROCESSMODELLOW, CONSTRUCT_CONCEPTUALMODEL, CONSTRUCT_CONCEPTUALMODEL)
SAMENVATTING

Evaluatie van technieken voor het modelleren van bedrijfsprocessen

Bedrijfsprocesmodellering is een relatief jong vakgebied dat zich de laatste jaren in een groeiende belangstelling mag verheugen. Zowel in de academische wereld als binnen het bedrijfsleven is een groeiend aantal technieken voor het modelleren van bedrijfsprocessen en ondersteunende gereedschappen in omloop. Belangrijke voorbeelden zijn traditionele technieken als Data Flow Diagramming (DFD) en moderne technieken als Event Driven Process Chains (EPC's). De hieruit voortvloeiende bedrijfsmethoden worden voor zeer uiteenlopende doeleinden gebruikt, variërend van het inzichtelijk maken van bedrijfsprocessen voor werknemers tot het afleiden van specificaties met betrekking tot ondersteunende geautomatiseerde systemen of de configuratie van zogenaamde Enterprise Resource Planning (ERP) systemen.

Steeds vaker staan bedrijven voor de keuze welke techniek zij moeten kiezen voor het modelleren van hun bedrijfsprocessen. Het groeiend aantal technieken en gereedschappen alsmede het ontbreken van voldoende instrumentarium om deze technieken te beoordelen op hun geschiktheid voor het modelleren van bedrijfsprocessen, geeft aanleiding tot de volgende onderzoeksdoelstelling: Het ontwikkelen en valideren van een raamwerk voor het inzichtelijk maken en meten van de kwaliteit van technieken voor het modelleren van bedrijfsprocessen.

Vanuit een bepaalde opvatting over hoe fenomenen in de werkelijkheid kunnen worden opgevat als bedrijfsproces, wordt de modeller een aantal modellerconcepten en mogelijke onderlinge verbanden aangereikt. Deze concepten en mogelijke verbanden stellen de modeller in staat de werkelijkheid op die manier te modelleren. Zo zien we bijvoorbeeld vanuit de software-engineering de trend dat men vanuit de zogenoemde objectgeoriënteerde gedachte concepten aanreikt waarmee bedrijfsprocessen gemodelleerd kunnen worden. In dat geval worden bedrijfsprocessen opgevat als een systeem van objecten die bepaalde eigenschappen hebben en een bepaald gedrag vertonen. Vanuit een werkstroomgedachte kunnen we bedrijfsprocessen opvatten als een stroom van activiteiten, al dan niet ondersteund door technologie, die door werknemers wordt uitgevoerd. Het is belangrijk te ondernemen dat de achterliggende gedachte, de theorie, bepaalt hoe een fenomeen in de werkelijkheid als bedrijfsproces
gemodelleerd kan worden. Een individueel model is uiteindelijk het resultaat van het toepassen van een theorie op een bepaald fenomeen in de werkelijkheid.

Door middel van technieken is vastgelegd welke concepten en onderlinge verbanden de modeller kan gebruiken om bedrijfsprocessen vast te leggen. Er moet een onderscheid gemaakt worden tussen zogenoemde aspectmodelleertechnieken en multimodelleertechnieken. Aspectmodelleertechnieken richten zich op een afgebakend aspect van het bedrijfssysteem, bijvoorbeeld de structuur van de organisatie, de informatiestruktuur of de processtructuur. De jongste generatie technieken moet echter multimodelleertechnieken worden genoemd. Zij bestaan uit een samenhangend geheel van aspectmodelleertechnieken. Niet alleen worden bijvoorbeeld technieken aangeboden om de informatiestruktuur en processtructuur vast te leggen, ook de onderlinge verbanden daartussen moeten, gebruikmakend van de techniek, in kaart gebracht kunnen worden. Samenvattend kan worden gezegd dat een multimodelleertechniek bestaat uit een aantal samenhangende aspectmodelleertechnieken die op hun beurt modellerconcepten aanbieden om een bepaald aspect van het bedrijfssysteem te kunnen modelleren.

Een eerste stap in het onderzoek naar modelleertechnieken is het precies vastleggen van de geboden modellerconcepten en hun onderlinge relaties. Hiertoe is de constructie van zogenoemde metamodellen (in dit geval, modellen van modelleertechnieken) noodzakelijk. Er zijn bestaande technieken voor de constructie van deze metamodellen, maar met name in het vastleggen van de onderlinge relaties tussen aspectmodelleertechnieken en multimodelleertechnieken schieten zij momenteel tekort. In dit proefschrift is de C-Me (Capturing Models for Evaluation) modelleertaal ontwikkeld die het mogelijk maakt modellerconcepten en hun onderlinge verbanden vast te leggen op het niveau van aspectmodelleertechnieken en bovendien de onderlinge relaties tussen aspect modelleertechnieken binnen een multimodelleertechniek vastlegt.

Een tweede stap is het evalueren van technieken op basis van deze metamodellen. Als uitgangspunt is gekozen dat bedrijfsmodellen die het resultaat zijn van de toepassing van een techniek enerzijds correct en anderzijds nuttig en waardevol (Eng: useful) moeten zijn. Vanuit de linguïstische school wordt onder interne correctheid empirische, syntactische, semantische en pragmatische correctheid verstaan. Onder externe correctheid wordt validiteit, de mate waarin een model met de werkelijkheid overeenkomt verstaan. Er is onderzocht op welke manier technieken de correctheid van modellen kunnen beïnvloeden door bijvoorbeeld te kijken naar de beperkingen die een modelleertechniek aan de modeller oplegt. Voorts is er vastgesteld wat de meest geschikte modellerconcepten zijn voor de constructie van nuttige en waardevolle modellen van bedrijfsprocessen. Om de geschiktheid van technieken te evalueren is ervoor gekozen de modellerconcepten die worden geboden te beschrijven binnen het zogenoemde discrete dynamisch systeem paradigma. Als voorwaarde wordt gesteld dat bedrijfssystemen kunnen
worden gezien als een systeem met een bepaalde toestandsruimte waarbinnen bepaalde toestandsveranderingen kunnen plaatsvinden. Als eis wordt gesteld dat modeleer technieken in staat zijn een bedrijfssysteem op deze manier te modelleren.

Dit alles heeft geleid tot het Q-Me (Quality based Modeling Evaluation) raamwerk. Het raamwerk benoemt een aantal eigenschappen van modelleer technieken en biedt in de vorm van de C-Me-beschrijvingstaal en een aantal metrieken een instrumentarium om inzicht in deze eigenschappen te verkrijgen.

Ter validatie zijn vier modeleer technieken gekozen die representatief zijn voor de technieken die momenteel in de wetenschap en in het bedrijfsleven worden gebruikt. Deze technieken zijn door toepassing van het Q-Me-raamwerk geëvalueerd. De toepassing toonde aan dat de C-Me-modelleertaal een nuttig instrument is om modelleer technieken vast te leggen. Een empirische studie waarbij een aantal modellerens de vier technieken heeft toegepast op een casus toonde aan dat de metrieken weliswaar een nuttig instrument zijn om inzicht te geven in de eigenschappen van technieken, maar dat de invloed van deze eigenschappen op de correctheid van de modellen beperkt is. De geschiktheid van de technieken om bedrijfsprocessen te modelleren is geëvalueerd door de vier technieken te beschrijven binnen het discreet dynamisch systeem paradigm. Deze beschrijving toonde tekortkomingen van technieken aan en met name verschillen in de manier waarop de technieken verschillende aspecten van bedrijfssystemen en hun onderlinge relaties modelleren.

Het instrumentarium dat het Q-Me-raamwerk biedt kan in de toekomst in zijn geheel of gedeeltelijk worden gebruikt om inzicht te krijgen in de eigenschappen van technieken voor het modelleren van bedrijfsprocessen. Als zodanig kunnen zij bijvoorbeeld worden gebruikt binnen techniek selectietrajecten, evaluatieprojecten en method-engineering projecten. Bovendien is de voorgestelde C-Me-modelleertaal een zeer geschikte specificatietaal voor het ontwikkelen van gautomatiseerde gereedschappen ter ondersteuning van technieken voor het modelleren van bedrijfsprocessen.

Bart-Jan Hommes was born in Oud-Beijerland on June 10th, 1972. He attended MAVO, HAVO and VWO at the Willem van Oranje scholengemeenschap in Oud-Beijerland and obtained his VWO diploma in 1992. He continued his education at the Technical University of Delft and received his master’s degree in 1998. The thesis focused on Dynamic Enterprise Modeling (DEM), a business process modeling technique developed by the Baan Company, a Dutch vendor of ERP software. In 1998 he started with a Ph.D. research project ‘the evaluation of business process modeling techniques’ at the same university and received his Ph.D. in 2004. The research has resulted in several publications and presentations on a number of international conferences. From 2002 on, Bart-Jan is working as a lecturer and researcher at the university of Delft.