Business Process Change Analysis and Business Process Simulation in the Context of Enterprise Engineering

Thesis · March 2015

DOI: 10.13140/RG.2.1.1403.0243

Yang Liu
Shenyang Normal University

Business Process Change Analysis and Business Process Simulation in the Context of Enterprise Engineering

Yang Liu

Supervisor:
Professor Junichi Iijima

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Doctoral Program of

Department of Industrial Engineering and Management
Graduate School of Decision Science and Technology

Tokyo Institute of Technology

February 2015

ABSTRACT

To respond to and to defeat their competitors, an organization must analyze their business processes and improve or change them to become more agile, efficient and effective. Sometimes, after an organization has carefully re-designed their business model, vision, and mission, their business processes must be quickly restructured to support these upper-level changes. However, contrary to their plans, the author sees high failure rates in many real-life business processes due to re-design and reengineering.

These high failure rates are caused by limitations in the traditional workflow perspective of business processes and limitations in the research methods used to support these changes. The workflow perspective places too much emphasis on details without a broad and high-level perspective; therefore, while this method can address the "how to" question, it is less capable of answering the "why" and "what is required" questions. Moreover, most business process modeling methods are aimed at developing information systems instead of business process changes. The separation of modeling and simulation makes these models weak at describing large and complex systems; they are also not very effective for supporting business process changes.

To solve these problems, the author investigated a more effective methodology to support business process change analysis. Enterprise engineering provides us with a different organizational perspective that considers an enterprise not as a set of separated workflows but as a complete coordinated system. Using this concept, the author introduced a new perspective and a corresponding method in order to analyze enterprises for improvement, reengineering or transfer. The new method analyzes enterprises from a new perspective to answer the "why", "what is required" and "how to" questions. The new method also seeks to combine modeling and simulation to provide an executable and measurable model and simulation framework that enables to describe large and complex systems and is modularized to support these changes

and the ‘to-be’ simulation. The outcome of this research can be described in three parts.

- **How to analyze:** The first part is a qualitative research framework for analyzing business process changes; this part of the research explains how the author can consider construction changes in the context of enterprise engineering. The nine proposed types of changes are guidelines that can be used throughout this research.
- **How to simulate:** The second part is a simulation framework that includes a conceptual modeling method and libraries that can be reused to analyze business process changes via calculations, comparisons, and evaluations. The proposed DEMO++ (expanded enterprise ontology) includes both ontology model and implementation model that can be used as a conceptual modeling method for business process simulations.
- **How to apply:** The third part is an application of the proposed methodology in practice.

Compared with traditional workflow-based business process modeling and simulation methods, the enterprise engineering-based DEMO++ method is not only a simulation method but also an analysis method. This method can be well integrated with management to analyze problems, seek solutions and evaluate alternative plans for enterprise reengineering. In the context of enterprise engineering, DEMO++ provides more capabilities than simulation alone. It is better at analyzing and simulating complex business processes and collaboration activities, which other traditional simulation models cannot adequately support. Moreover, this method is a modularized, component-based simulation model with increased reusability, changeability and flexibility.

Another contribution of this research is practical. By clarifying the differences and dependencies between ontological and implementation models, our methodology was used to develop a generic framework for generating modularized, component-based simulation models with increased reusability. The proposed components were developed as an AnyLogic DEMO++ library, which can be reused in other DEMO++ based simulations.

ACKNOWLEDGMENTS

This dissertation was completed with the input and kind cooperation of several people who contributed their fruitful ideas, skills, and support throughout the process. I am very grateful to the following individuals and organizations who contributed to the success of my study.

I would like to express my greatest gratitude to Professor Junichi Iijima, my supervisor at the Tokyo Institute of Technology. His insights, advice, support, patience, and kindheartedness are inestimable. I am also grateful to Assistant Professor Jaehyun Park for his kind advice and his strong encouragement of my Ph.D. research.

My research would not have been completed without my colleagues at the Iijima Laboratory, especially Ms. Keiko Kitahara and Ms. Sho Ho, who kindly supported my Ph.D. studies.

In addition to my university affiliations, I wish to thank Professor Jan L. G. Dietz, Professor Joseph Barjis, Professor Mamadou Seck and others in the CIAO group for their valuable advices regarding my research. I especially want to thank the CAC pension group members. This dissertation could not have been completed without their help in providing fruitful information.

Last but not least, I would like to give special thanks to my beloved family, especially my mum, my dad and my husband, for their love, their strong support and their understanding during the past years. This work is conducted as a commitment and a gift for all of you.

TABLE OF CONTENTS

ABSTRACT I

ACKNOWLEDGMENTS III

TABLE OF CONTENTS IV

TABLE OF FIGURES VIII

TABLE OF TABLES XI

CHAPTER 1 Introduction 1

1.1 Research Motivation 1

1.2 Research Objectives 3

1.3 Research Questions 4

1.4 Structure of Dissertation 4

CHAPTER 2 Literature Review 7

2.1 Perspectives on Business Processes and Business Process Changes 7

2.1.1 Workflow perspective: How work is related... 7

2.1.2 Coordination perspective: How and why humans are related............................ 10

2.2 Methods for Analyzing Business Processes and Business Process Changes ... 12

2.2.1 Business process modeling 13

2.2.2 Business process simulation... 18

CHAPTER 3 Enterprise Ontology – DEMO 28

3.1 Case Description (Case C) 28

3.2 DEMO Aspect Models 30

3.2.1 DEMO construction model (CM) 30

3.2.2 DEMO process model (PM) 32

3.2.3 DEMO fact model (FM) 34

3.2.4 DEMO action model (AM) 35

CHAPTER 4 Business Process Change Analysis in the Context of Enterprise Engineering 39

4.1 Introduction 39

4.2 Types of Construction Changes 39

4.2.1 Changes in ontology level 42

4.2.2 Changes in implementation level 43

4.3 DEMO for Analyzing Business Process Improvements 45

4.3.1 Case description (Case A) 45

4.3.2 Exceptions and business process improvements in “Company A” 47

4.4 DEMO for Analyzing Enterprise Transformations 50

4.4.1 Case description (Case B) 51

4.4.2 From passive to active 52

4.4.3 From product to service 57

4.4.4 Assumptions and analysis 59

4.4.5 Transitions in the transformation process 61

4.4.6 Conclusion 64

4.5 Discussion 65

CHAPTER 5 Business Process Simulation in the Context of Enterprise Engineering **67**

5.1 Introduction **67**

5.2 Research Questions and Research Design **68**

5.3 DEMO **70**

5.4 DEMO++ **70**

5.4.1 Overview of DEMO++ 70

5.4.2 Ontological model 72

5.4.3 Implementation model 77

5.4.4 Main model 81

5.4.5 Meta-model of DEMO++ 82

5.5 DEMO++ Library for AnyLogic **82**

5.6 Model Transformation **86**

5.6.1 Model transformation inside the DEMO aspect models (T1 and T2) 86

5.6.2 Model transformation from DEMO to DEMO++ 88

5.7 Case C: DEMO++ Based Business Process Simulation **93**

5.8 Discussion **97**

CHAPTER 6 A Case Study of Business Process Simulation **100**

6.1 Introduction **100**

6.2 Case Description (Case D) **100**

6.3 DEMO Aspect Models **102**

6.3.1 CM 102

6.3.2 PM 104

6.3.3 FM 105

6.3.4 AM 106

6.4 DEMO++-based AnyLogic Model and Simulation 107

6.4.1 Main 107

6.4.2 Ontological model 108

6.4.3 Implementation model 110

6.5 Simulation 114

6.5.1 Data collection and setup 114

6.5.2 Simulation results, analysis, and suggestions 117

6.5.3 Feedback 120

6.6 Discussion 122

CHAPTER 7 Conclusions and Future Research 126

7.1 Summary of Main Contributions 126

7.2 Limitations and Future Research 128

References 130

Appendix A. Details of "Case D" 141

A.1 Work Flow 141

A.2 Implementation Models of Actor Roles 142

A.3 Action Rules 149

Appendix B. DEMO++ Library T 152

TABLE OF FIGURES

Figure 1-1 Structure of the dissertation 5

Figure 2-1 Roots of enterprise engineering (Hoogervorst, 2009)................................12

Figure 2-2 Basic transaction pattern 18

Figure 2-3 Simulation areas (Jahangirian et al., 2010) 24

Figure 3-1 DEMO aspect models 28

Figure 3-2 Meta-model of the DEMO construction model 30

Figure 3-3 OCD for Buono pizza..31

Figure 3-4 Standard transaction pattern...32

Figure 3-5 Meta-model of the DEMO PM 33

Figure 3-6 PSD of Buono pizza..34

Figure 3-7 Meta-model of the DEMO FM 34

Figure 3-8 OFD for Buono pizza...35

Figure 3-9 Meta-model of DEMO FM 37

Figure 4-1 Function and construction (ref. Dietz, 2006) 39

Figure 4-2 Key concepts in construction level 40

Figure 4-3 OCD as-is of “Company A”...47

Figure 4-4 ATD to-be of “Company A” 48

Figure 4-5 OCD of the second group: ‘as-is’ model and ‘to-be’ model......................53

Figure 4-6 OCD of the first group: ‘as-is’ model and ‘to-be’ model..........................58

Figure 4-7 Five stages of enterprise transformation 60

Figure 4-8 SECI for enterprise transformation 61

Figure 5-1 Framework for BPS..68

Figure 5-2 Meta-model of DEMO 71

Figure 5-3 DEMO++ 71

Figure 5-4 Meta-model of the DEMO++ ontological model...73

Figure 5-5 Object type component “Order”..74

Figure 5-6 Transaction type component “T1” 75

Figure 5-7 Meta-model of the DEMO++ implementation model...................................78

Figure 5-8 Examples of implementation models 79

Figure 5-9 Meta-model of DEMO++ Main 81

Figure 5-10 Example of the Main component 82

Figure 5-11 Meta-Model of DEMO++ 84

Figure 5-12 DEMO++ AnyLogic library...85

Figure 5-13 Model transformation framework 86

Figure 5-14 Meta-model mapping among the DEMO aspect models 87

Figure 5-15 Conversion from DEMO to DEMO++ 88

Figure 5-16 Meta-model mapping between DEMO aspect models and DEMO++ 89

Figure 5-17 Model transformation...91

Figure 5-18 T in the DEMO++ library 92

Figure 5-19 Main for Buono Pizza 93

Figure 5-20 State transition diagram of order instance “o5” 94

Figure 5-21 Transaction pattern of transaction T1 94

Figure 5-22 Implementation phase of [T2/ex] 95

Figure 5-23 Experimental parameter setting 95

Figure 5-24 Simulation result (4 staff, 4 preparation tables, 3 ovens, 2 motors) 96

Figure 5-25 Optimization result 96

Figure 6-1 OCD of the “Company D” proposal and estimation process 103

Figure 6-2 PSD of “Company D” proposal and estimation process 104

Figure 6-3 OFD of “Company D” proposal and estimation process 105

Figure 6-4 Main of the “Company D” proposal and estimation process 107

Figure 6-5 T in DEMO++ and AnyLogic 109

Figure 6-6 Objects of the “Company D” proposal and estimation process 110

Figure 6-7 Implementation details of the actor role “A1” 114

Figure 6-8 Skill level distribution 115

Figure 6-9 Simulation result statistics 117

Figure 6-10 Simulation results 118

TABLE OF TABLES

Table 2-1 Business process modeling methods 13

Table 2-2 Simulation methods 23

Table 3-1 Buono Pizza’s transaction product table 31

Table 3-2 Action rule table 38

Table 4-1 Types of changes 41

Table 4-2 Actor-actor role-function mapping table (Case A) 49

Table 4-3 Change type validation table for “Case A” 50

Table 4-4 Change type validation table for “Case B” 65

Table 5-1 Actor-actor role mapping table plan one (Case C) 80

Table 5-2 Actor-actor role mapping table plan two (Case C) 80

Table 5-3 Meta-model mapping table 93

Table 5-4 Change type validation table for “Case C” 99

Table 6-1 TPT of CAC proposal and estimation process 103

Table 6-2 Action rule table for actor role “A1” 106

Table 6-3 Parameter setting table for T of DEMO++ 109

Table 6-4 Actor-actor role-function mapping table (Case D) 111

Table 6-5 Processor table 113

Table 6-6 Simulation setup table 116

Table 6-7 Resource capacity table 117

Table 6-8 Change type validation table for “Case D” 120

CHAPTER 1 Introduction

1.1 Research Motivation

When mobile, internet-based new business models are created, when big data analyses cause an enterprise to change its strategy, and when customer-focused concepts lead an enterprise to change from a traditional production model to providing additional services that add value to its products, we see substantial changes in the current business world and large challenges that are associated with business processes. To respond to and to defeat their competitors, an organization must analyze their business process and improve or change it to become more agile, efficient and effective. Sometimes, after an organization carefully re-designs their business model, vision, and mission, their business processes must be quickly restructured to support these upper-level changes.

However, contrary to these goals, remarkably high failure rates occur in many real-life business process reengineering projects; these rates are based on several recent research studies (Beer, Eisenbach, & Spector, 1990; Dietz & Hoogervorst, 2014a; Kaufman, 1992). These previous studies have discovered that the current methods in business process re-design and reengineering are not effective for supporting the requirements of change. The reasons for this large gap can be considered from two aspects: perspectives and methods.

- **Perspectives:** According to Davenport et al. (1994) and Hammer et al. (1993), *business processes* are defined as structured, measured sets of input-output activities that produce value for a particular customer. In contrast to a traditional system, which emphasizes what is produced by "a black box", a business process emphasizes how activities are completed from a business logic perspective (Davenport et al., 1994). However, as Barber (2003) noted, most business process improvement projects consider only single processes without developing a holistic perspective of the enterprise. Generally speaking, most existing methods, for example BPMN, UML, EPC, analyze the enterprise from a workflow perspective that is used to represent the sequences that occur in real-world work. The advantage of the workflow perspective is clear: it can help us understand how activities are conducted in a step-by-step manner. However, this method occasionally leads us to over analyze some details such that we are not able to grasp a complete image of the system. For example, it is difficult to analyze tens of A4-paper-based flowcharts or UML models to understand why certain actions are performed, what relationships exist and how to make changes that can confirm the consistency of the system before and after the change is implemented. Furthermore, this method answers the “how to” question but is weak in analyzing the “why” and “what is required” questions. Moreover, models from a workflow perspective are typically non-modularized, although they are described as a sequence of activities. Using this perspective, it is difficult to conceive of a systematic approach that can guide a process re-designer through a series of repeatable steps. In addition, the enterprise is not considered as an entire system, and human interactions are not emphasized, considering the workflow perspective.

- **Methods:** Several researchers have argued that the high failure rate in many real-life business process reengineering projects is due to a lack of tools for evaluating the effects of the designed solutions before implementation (Paolucci *et al.*, 1997; Tumay, 1996). Static models, such as UML, BPMN, Flowchart, and IDEF, have been traditionally used in business process execution descriptions. However, these models do not add much value if they cannot assist with inspecting and analyzing the business processes. For example, they cannot provide a method for bottleneck identification and performance analysis or for the generation of alternative improved business processes in terms of specified objectives (Vergidis *et al.*, 2008). *Business process simulation (BPS)* (Scholz-Reiter *et al.*, 1999) is a powerful tool that can assist in change analyses and effectiveness evaluations due to its ability to measure performance, to test alternatives and to engage in processes (Greasley & Barlow, 1998). Based on several reviews (Aguilar-Savén, 2004; Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010; Netjes, 2006), BPS plays an important role in supporting business process-related analysis. However, current simulation methods remain weak in describing large, complex systems. As some studies (Jahangirian *et al.*, 2010; Barber *et al.*, 2003) have noted, most business process improvement projects consider only a single process without developing a holistic perspective of the enterprise, and the complexity is increased when small, individual process models are joined together into a large hierarchical construct. Thus, it is inefficient to use these methods for process change analyses of an entire enterprise. Another limitation of BPS is the complexity of changing the models that are used to simulate new designs in *business process reengineering (BPR)* (Greasley, 2003). Simulation is useful in comparing ‘as-is’ and ‘to-be’ models to validate the effects of change and to ensure the completeness of the models; however, this approach has limited ability for the design of ‘to-be’ models. Most of the business process simulation literature restricts itself to comparing the before and after conditions, providing little information regarding the redesign process itself (Reijers & Liman Mansar, 2005). Moreover, as Valiris *et al.* (2004) noted, most reengineering methodologies *“lack the formal underpinning to ensure the logical consistency of the generation of the improved business process models”*. This issue leads to a lack of a systematic approach, although a systematic approach is needed to guide a process designer through a series of repeatable steps to achieve process redesign. Vergidis *et al.* (2008) argued that *“a structural and repeatable methodology that could be generally applied to business process modeling and improvement was never established”*. This situation is problematic for business process modeling and for BPS.

1.2 Research Objectives

In summary, this research considers that the limitations in business process change analyses are caused by limitations in the traditional workflow perspective and in the available research methods. The workflow perspective places too much emphasis on details without developing a holistic, high-level perspective and answers the “how to” question without adequately addressing the questions of “why” and “what is required”. Workflow perspective-based methods are typically non-modularized, and the enterprise is not considered as an entire system in this perspective. In addition, current modeling methods lack tools for evaluating the effects of designed solutions before implementation, and current simulation methods are inadequate for describing large, complex system. These methods have limited application for the design of 'to-be' models and lack methodology to confirm the consistency of change.

To solve these problems, this research focused on finding a more effective methodology to support business process change analysis. Enterprise engineering provides a different perspective in terms of organization, which views enterprises not as separated workflows but as entire coordinated systems. Using this context, this research introduces a new perspective and a new method to analyze enterprises for improvement, reengineering, or transfer. The new perspective analyzes enterprises from a fresh angle to answer the "why", "what is required" and "how to" questions. The new method combines modeling with simulation to provide an executable and measurable model and simulation framework that can describe large and complex systems and that can be modularized to support additional changes and 'to-be' simulations.

1.3 Research Questions

To achieve the aforementioned objectives, this research restricted the scope to answer the following three questions.

- Q1: How to analyze business process changes in the context of enterprise engineering?
- Q2: How to conduct a simulation in the context of enterprise engineering to assist in business process change analysis?
- Q3: How to use this enterprise engineering-based business process simulation to assist an actual case?

1.4 Structure of Dissertation

To answer the three aforementioned questions, the research was designed as shown in Figure 1-1.

Chapter 1 briefly describes the motivation for this research: what is the problem, and what are the causes of the problem? It also defines research objectives and scope based on the corresponding research questions. These questions will be individually answered in the following chapters.

Chapter 2 reviews the related research to examine how other researchers have considered this problem and to discuss the advantages and limitations of those solutions. The review was conducted according to two aspects: (1) perspectives used in analyzing business processes and changes in business processes, including a workflow-based perspective and a coordination-based perspective, and (2) methods that can be used to analyze business processes and business process changes, including modeling and simulation.

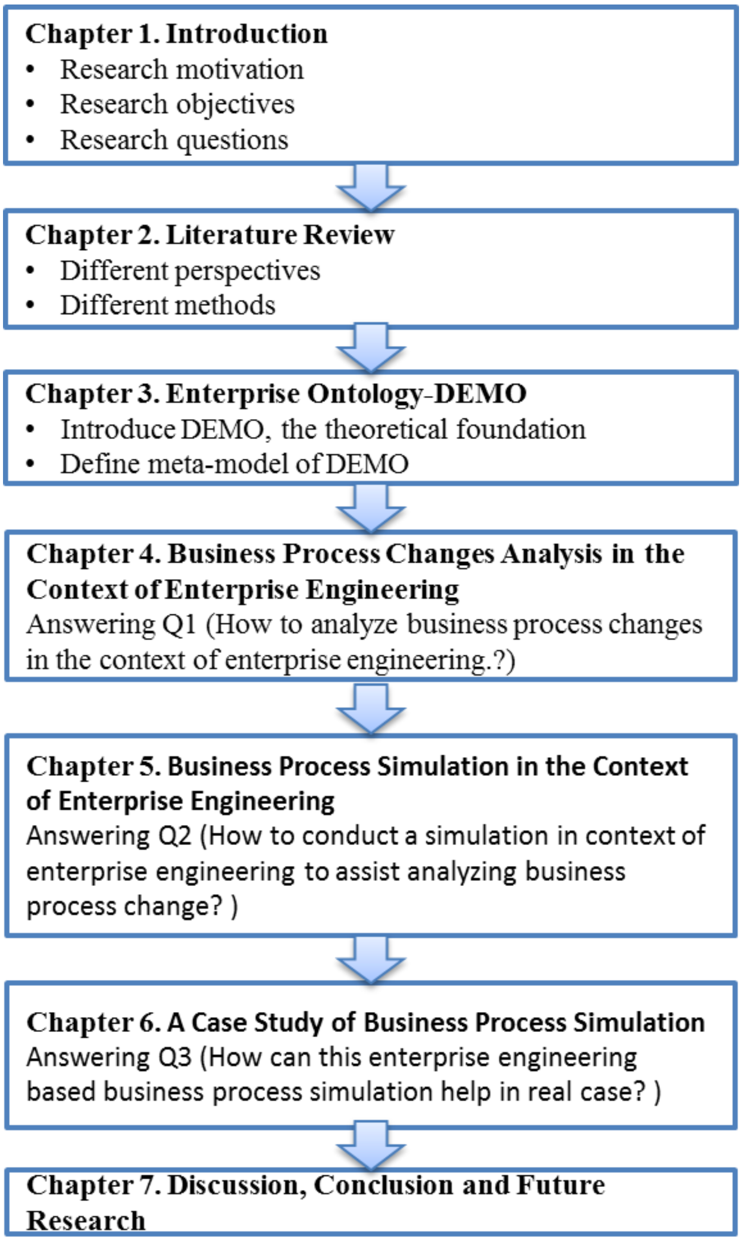


Figure 1-1 Structure of the dissertation

Chapter 3 introduces enterprise ontology-DEMO as a fundamental theory for enterprise engineering. The entire research project is based on the DEMO concepts. Meanwhile, meta-models of the DEMO aspect models are defined in this chapter.

Chapter 4 defines nine types of changes to answer the first research question: “How to analyze business process changes in the context of enterprise engineering?” Following the concepts of enterprise engineering, an enterprise is analyzed according to its function and construction. Focus is placed on construction changes; these changes are classified into six types in two levels: the ontological level and the implementation level. The types of changes are validated via two real-world case studies: one study (Case A) represents business process improvements, while the other study (Case B) corresponds to enterprise transformation.

Chapter 5 proposes a business process simulation methodology in the context of enterprise engineering, following the framework of business process change defined in chapter 4. In the proposed methodology, a DEMO-based conceptual modeling method for business process simulation is conducted, which is called DEMO++. This approach combines ontology with implementation in the construction level so that each DEMO++-based simulation is modularized, easy-to-change and reusable. Based on DEMO++, a reusable DEMO++ library is developed using the simulation platform AnyLogic; moreover, a semi-automatic model transformation method that aids in the generation of the simulation model is presented. This methodology is validated using a simple pizza store case (Case C).

Chapter 6 conducts a real-world case study to further investigate how the proposed business process simulation can assist in business process change and to evaluate the advantages, potentials and limitations of the methodology. In this chapter, a “proposal and estimation process” in an actual Japanese company (Case D) is analyzed. Meanwhile, this case study is also conducted in order to standardize and demonstrate the process of developing DEMO++ based simulation for future applications. This chapter answers the third research question.

Chapter 7 concludes with a discussion of the contributions and limitations of this research, and provides a plan for future research.

CHAPTER 2 Literature Review

Different types of studies have been conducted, describing business processes and business process changes from different perspectives and for different objectives. To ensure that this review well organized, this chapter has classified and explained the previous studies according to two aspects: (1) different perspectives regarding business processes and business process changes, mainly including studies from the workflow-based and coordination-based perspectives, and (2) different methods to support business process change analyses that primarily include modeling and simulation methods.

2.1 Perspectives on Business Processes and Business Process Changes

2.1.1 Workflow perspective: How work is related

The workflow story began when Taylor published his theory of management in the book *Principles of Scientific Management* (Taylor, 1911), which discusses how organizations are considered to be a system with clear relationships between inputs and outputs. This system-wide perspective emphasizes that everything is connected to everything else and that businesses are often modeled as *workflows*, defining how work activities are related. This opinion is the foundation of modern business process research.

Current business process research is based on value chains that Porter introduced in 1985 (Harmon, 2007). Applying a system-wide perspective, business processes are considered to be a set of activities that cross boundaries between departments throughout a value chain (Rummler, 1984). This holistic perspective takes the stance activities in an organization that can work together to achieve a common goal instead of working separately. This concept was later inherited by other researchers.

The most cited definition of a business process is that given by Davenport *et al.* (1994), who stated that a business process is a "*structured, measured set of activities designed to produce a specific output for a particular customer or market*". He also stated that a business process emphasizes how activities are conducted based on a business logic perspective rather a traditional system-wide perspective. According to this definition, a process must have clearly defined boundaries, incomes and outcomes that consist of activities ordered in time. Therefore, a business is structured by "cross-function processes" that are required to produce the necessary value. Irani *et al.* (2002), in accordance with Davenport, emphasized that businesses should be analyzed in terms of the key business processes that they use and not in terms of functions that they can be decomposed into or in terms of the products they produce.

Another popular definition was proposed by Hammer and Champy (1993), who stated that "*a business process is a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer*". This definition is different from Davenport`s definition, because it emphasizes inputs, outputs, activity sets and values and ignores the "cross-function" and "structure" perspectives of a business process.

Rummler & Brache (1995) stated that a business process from an external customer perspective is "*a series of steps designed to produce a product or service*". This definition follows Porter's value chain model. Porter's model classifies business processes into two types: a *"primary process"* that results in a product or service being provided to an external customer, and a *"support process"* that is invisible to the external customer, although it is essential for the effective management of the business. Most other definitions (Guha, Kettinger, & Teng, 1993; Strnadl, 2006) have followed one of these three models with some amendments.

The enterprise advocated by Taylor strongly resembles reductionist and deterministic thinking (Hoogervorst, 2009). Thus, the aforementioned definitions have a common characteristic: they all employ a workflow perspective of business processes. A typical interpretation of a *workflow perspective* is that it has clearly definable inputs and outputs that a clearly causal relationship exists between the inputs and outputs.

Several concepts related to business process changes were introduced in the early 1990s from the workflow perspective. Davenport and Stoddard stated that *business process changes* should transform only work processes without considering any other aspect of an organization (Davenport et al., 1994). Therefore, understanding and analyzing business processes help us discovering the source of problems, which is valuable for ensuring that these problems are not repeated in the new process when proposing changes (Irani et al., 2002). Total quality management (TQM) (Deming, 1986) is a method that can be used to make continual changes to a routine undertaking that follows this understanding. Davenport's opinion is founded in the belief that information technology is capable of creating major improvements in business processes. In the mid-1990s, information system-driven business process redesigns and improvements were initiated (Harmon, 2007). Companies began to use software programs called *workflow systems* to automate their business processes. Vendors, e.g., SAP, Oracle, and PeopleSoft began to organize their applications as business processes to provide the "best practice solutions", and these solutions enclosed the workflows. These workflow and Enterprise Resource Planning (ERP) systems, or E-business suites, were widely used in companies to rebuild and improve their processes.

Hammer and Champy (1993; 2006), who hold an opposite perspective, proposed BPR as fundamental changes that cause "*the fundamental rethink and radical redesign of business processes to generate dramatic improvements in critical performance measures — such as cost, quality, service and speed*". Hammer's key point was that even though information systems can improve the functions of a department, these functions are not redesigned by this process and are instead automated (Harmon, 2007). Furthermore, limited improvements in a department may unexpectedly increase problems in other departments. Thus, changes should be performed from the ground up and should consider not only the business processes but also other aspects of an organization, such as the structure, strategy, resource allocations and business capabilities. However, the foundational changes of BPR are considered by these researchers to be a "myth" due to the challenges in creating a *"clean slate*", performing "*top-down design*", and preparing to "*make transformation based reengineering*". BPR projects frequently attempt *"revolutionary"* (radical) changes that for the limitations of political, organizational and resource constraints, take on *"evolutionary"* (incremental) implementations (Davenport et al., 1994; Stoddard & Jarvenpaa, 1995). In practice, there are several drawbacks that caused the BPR concept to become unpopular during the late 1990s.

2.1.2 Coordination perspective: How and why humans are related

In contrast to the workflow perspective, Keen and Knapp (1995) argued that there are some processes that have no clear inputs, flows, and outputs, such as governing, management succession, acquisitions, and incentives. The workflow perspective can also cause management to ignore other aspects of an organization that may require improvements. Moreover, a workflow perspective focuses more on the tasks that are being executed rather than on the employees involved in the cooperation activities required to complete these tasks. However, an enterprise is far more complex than any workflows that are defined following a mechanistic approach. The employees involved in activities requiring cooperation and the ignored processes occasionally increase the complexity of an enterprise, which may require additional improvements (Hoogervorst, 2009).

Furthermore, although the workflow perspective has been widely accepted in supporting business process changes, the high failure rate of business process reengineering, TQM, six sigma, and ERP projects, i.e., the majority of failures in strategic initiatives, demonstrates that enterprises continue to have difficulty in changing their processes to derive success from their strategies (Dietz & Hoogervorst, 2012). As Dietz et al. (Dietz & Hoogervorst, 2012) indicated, 70% to 90% of all strategic initiatives appear to fail. The primary reason for these failures is *"inadequate strategy implementation"* because *enterprises* are an overall product that is intended to build *"an intentionally created cooperative of human beings with a certain societal purpose"* (Dietz et al., 2013). Enterprises are highly complex units with highly organized entities (Hoogervorst, 2009). Traditionally, the organizational sciences do not verify that enterprises, from their design to their implementation, are coherently and consistently integrated entities. Thus, it is difficult to implement successful changes in a system based on a strategy change. Many authors have argued that using a system-wide approach is the only meaningful way to address the core problem of organized complexity; they have also argued that it is the only meaningful way to study and develop enterprises (Dietz, 2006).

Keen et al. (Keen & Knapp, 1995) suggested that a business process can be defined from the *coordination perspective* as follows: "*It is recurrent; it affects some organizational capabilities; it can be accomplished in different ways, which affects* *the contribution it generates in terms of cost, value, service or quality; it involves coordination"*. Based on this perspective, the core of a business process is *"coordination through language"* (Flores, 1991). Thus, cooperation and collaboration occur primarily through informational relationships in which the work need to be not only *automated* but also the content must be *informated* (Hoogervorst, 2009). The difference was distinguished by Dietz (Dietz, 2005), who stated that the "automation" concerned with transitions in the "form of information", while the "information" is concerned with transitions in the "content of information". Furthermore, Dietz (2006) explained that *cooperation* occurs through a completed communication pattern during which social actors must enter into the expression of communicative acts and comply with commitment (Hoogervorst, 2009). *Basic communicative acts* include a *request* (to produce a product or service), *promise* (to honor the request), *state* (when the product or service has been produced) and *acceptance* (of the statement). The content of this information is informed to accomplish the communicative acts. Thus, these activities are related by the intention of information; this intention should be placed above the content and format of the information.

- By focusing on the *form of the information*, we can address the "what is the information" question; this is "data system engineering";
- By focusing on the *content of the information*, we can address the "how do we use the information" question; this is "information system engineering".
- By focusing on the *intention of the information*, the "why do we need the information" question can be addressed; this is "enterprise engineering". The "why do we need the information" question is more qualitative because it is closely related to how and why humans are related in terms of cooperation and collaboration.

As described in Figure 2-1, *Enterprise engineering* is a combination of information system engineering and traditional organizational science. A traditional system philosophy focuses on what an enterprise should achieve; business processes are interested in how this work should be accomplished. Enterprise engineering concerns both of these aspects, intending to combine design and implementation to address design perspectives in a formal, methodological manner (Dietz et al., 2013). On one hand, enterprise engineering is concerned with an understanding of the strategic intentions that are to be operationalized; on the other hand, the required arrangements are also of interest.

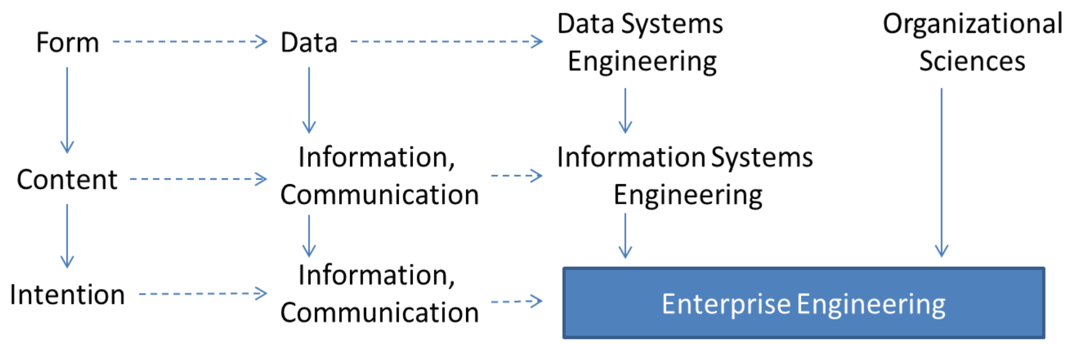


Figure 2-1 Roots of enterprise engineering (Hoogervorst, 2009)

Enterprise engineering is an integrated set of disciplines for building or changing an enterprise, its processes, and its systems (Martin, 1995). CIAO! group members published a paper regarding enterprise engineering in which they considered the reasons for the low BPR success rate as "*enterprise phenomena are not comprehensively understood, cannot be addressed adequately, and hence the nature of necessary changes cannot be determined*" (Dietz et al., 2013). According to this perspective, an enterprise is an intentionally created cooperative of human beings with a certain societal purpose. "*In order to perform optimally and to implement changes successfully, enterprises must operate as a unified and integrated whole, taking into account all aspects that are deemed relevant*" (Dietz et al., 2013). Although its analysis of enterprises is top-down, enterprise engineering takes a human-centered perspective. This perspective is contrary to Taylor's mechanistic perspective of organizations.

2.2 Methods for Analyzing Business Processes and Business Process Changes

There are numerous techniques for analyzing business processes with approaches that capture different aspects of a business process; each of these techniques has distinctive advantages and disadvantages. Aguilar-Saven (2004) classified the main approaches into two categories: active and passive.

- *Active models* are models that allow the user to interact with them (dynamic models).

- *Passive models* are models that do not provide this capability (static models).

Active models are more related to business process simulations, while passive models are more related to business process modeling. A brief review of both models is provided in the following section.

2.2.1 Business process modeling

According to Ackoff (1963), *models* are idealized representations of states, objects, and events to describe what a thing is like. They are less complicated than reality; hence, they are easier to use for research purposes. Compared with the real world, only the relevant properties of reality are represented in a model. Each model has a focus that provides an idealized representation of the aspects it is concerned with and ignores other aspects.

The purpose of *modeling* is to demonstrate, to reveal, or to show what a thing is like based on only the relevant properties (Iijima, 2014). *Business process modeling* involves the understanding and representation of business processes. Business process modeling methods are primarily used for two purposes: one method is used for traditional software development, while the other method is used for restructuring business processes (Phalp & Shepperd, 1999).

Table 2-1 Business process modeling methods

Business process models	Description	Model categories
Flowcharts	Sketch a business process using a visual diagram.	Diagrammatic models
IDEF		
RAD		
DEVS	All the elements have a mathematical or a formal underpinning	Formal/Mathematical models
Petri-net	Formally defined diagram models	
BPEL	Software-based languages that support business process modeling and most of the times process	Executable models
BPML		
UML	Diagrammatic models that support process execution in software development	
BPMN		
YAWL		
EPC		
DEMO	Formally defined diagram models describing ontology	Ontology models

Based on Vergidis's classification (Vergidis et al., 2002), the most popular business process modeling technologies are listed and classified into four categories, as presented in Table 2-1. The categories include diagrammatic models, formal/mathematical models, executable models and ontological models.

2.2.1.1 Diagrammatic Models

Diagrammatic models are the simplest business process modeling methods; they represent a business process using visual diagram. There is no semantic description for these models. Typical diagrammatic models include flowcharts, role activity diagrams (RADs) and integrated definition for function (IDEF) modeling.

- Flowcharts: According to (Aguilar-Savén, 2004), a flowchart is a type of diagram that represents a workflow or a process. The steps are called activities, which are denoted as boxes of various types; decisions are denoted as diamonds. Flowcharts are always used in describing business processes with a high level of detail; they are not used to provide an overview. Flowcharts do not define responsibilities. Moreover, they can be difficult to connect with organizational functions.
- RADs: RADs provide a graphic perspective of business processes from the perspective of individual roles that is concentrated on the responsibilities of the roles and the interactions between them (Holt, Ramsey, & Grimes, 1983). These diagrams are easy and intuitive to read and to understand; they present a detailed perspective of the process and permitting activities in parallel. Moreover, these diagrams also demonstrate how the processes interact. The disadvantage of RADs is that they are presented as a sequence of activities; this concept does not allow for a decomposition of the process. Thus, an overview of the process is difficult (Aguilar-Savén, 2004).
- IDEF modeling: IDEF modeling is a family of methods that support a paradigm capable of addressing the modeling needs of an enterprise and its business areas (IDEF). IDEFs include methods from IDEF0 to IDEF5. The most useful methods are IDEF0 and IDEF 3. IDEF0 is a modeling technique that is used for developing structural and graphical representations of processes or complex systems as enterprises. It is used to specify function models and to express "what is done within the organization models". The IDEF3 process description capture

method is used to capture the behavioral aspects of a process, providing different perspectives of how things function within an organization. (Aguilar-Savén, 2004).

2.2.1.2 Formal/Mathematical Models

Formal models are mathematical or contain a formal underpinning of a business process (Vergidis et al., 2002). A diagram model is easy to create; however, such a model can also easily cause ambiguity about a process. On the contrary, a formal description is far more difficult to create and maintain. It is also difficult to retain consistency; however, this approach ensures the accuracy of a process and can be analyzed using an extraction tool (Koubarakis & Plexousakis, 2002). A typical formal model with a diagram description is petri-net.

Petri-net was first created by Carl Adam in August 1939 to describe chemical processes. It is a mathematical modeling language for describing the state and state transitions associated with a discrete system. Petri-net also offers a graphical notation for describing workflows and contains a well-developed mathematical theory for process analyses. The key elements defined in petri-net include place, transition and arc.

- *Place*: describes the states of a system;
- *Transition*: describes the functions or activities that change the state;
- *Arc*: creates links from place to transition or vice versa.

2.2.1.3 Executable Models

Executable models use process languages in most of the XML-based cases to model and execute a business process. Such languages always have clear semantics that are used to describe business processes; these languages can be utilized for models. Moreover, such models contribute to the analysis of their structural properties (van der Aalst, ter Hofstede, & Weske, 2003)

- Business process execution language (BPEL) for web services: BPEL is not a notational language; instead, it is an XML-based executable language that was founded on the OASIS standard (https://www.oasis-open.org/). Inheriting XML attributes, BPEL specifies actions within business processes using web services.

A BPEL message is facilitated depending on the web services description language (WSDL).

- Business process modeling language (BPML): BPML is a subset of BPEL that was produced by the Business Process Management Initiative (BPMI, www.bpmi.org) for modeling business processes. This language is also based on XML and encodes the flow of a business process in an executable form.
- Business process modeling notation (BPMN): BPMN is a graphical notation that specifies a business process based on a flowchart. The first version was proposed by the BPMI in 2004, and the second version was proposed by the Object Management Group (OMG). These graphical notations can be mapped into BPEL. BPMN is able to describe process semantics; therefore, it is usually a bridge between business process design and implementation.
- Yet another workflow language (YAWL): YAWL is another graphical process language created by van der Aalst and ter Hofstede. It is a petri-net-based language that was built with the primary goal of supporting a wide range of business process patterns (ter Hofstede, van der Aalst, & Adams, 2010). YAWL extended petri-net with several operators, such as OR, AND, and XOR, which are better for supporting and describing workflows.
- Event process chains (EPCs): EPCs are workflow diagrams that originated from SPR R/3 modeling. EPCs were developed within the framework architecture of integrated information systems (ARIS) (Iijima, 2014). The key elements in EPCs are as follows.
    - *Event*: The passive elements throughout the process.
    - *Function:* The active elements, which describe tasks and activities.
    - *Organization unit:* Determines the organization that responds to the function.
    - *Input and output:* Defines any additional inputs and outputs that are required to perform a function.
    - *Logical relationships:* Defines the OR, XOR, and AND relationships.

2.2.1.4 Ontology Model

The aforementioned modeling methods are the most popular techniques used to describe business processes from a passive perspective. By comparing and analyzing each method, the author has discovered that all of these methods stem from a workflow perspective and include clear inputs, outputs, and rational relationships that generate outputs from the provided inputs.

However, as discussed in section 2.1, a process described from a workflow perspective may have difficulty in describing the cooperation that exists between humans and may also have difficulty in handling the complexity of an enterprise. Enterprise engineering, which represents another coordination perspective, considers an enterprise as a whole. This perspective analyzes business processes by considering the "why are humans related" question and defines the cooperation that exists between humans, which causes the design to happen.

The enterprise ontology DEMO is a core theory for enterprise engineering. The goal is to offer a new understanding of enterprises so that one is able to look through the actual distracting and confusing appearance of an enterprise and into its essence (Dietz et al., 2013).

DEMO is based on *ψ theory*, which considers an organization to be an interaction of individual social subjects. A *subject "enters into and complies with commitments regarding the products/services that they bring about in cooperation"* (Dietz & Hoogervorst, 2014b). A *product* is an independently existing fact (for example, "pizza order #002 has been delivered"). Subjects generate products by performing *production acts* (or Pacts for short). Meanwhile, subjects enter into and make commitments toward each other regarding products by performing *coordination acts* (Cacts for short). The state changes caused by these Cacts/Pacts are called Cfacts/Pfacts. Cacts and Pacts occur in universal patterns called *transactions*. As described in Figure 2-2, a transaction involves two subjects: an *initiator* that generates a request and an *executor* that produces the products (Dietz & Hoogervorst, 2014b). Transactions are the elementary (essential) organizational building blocks of enterprises (Perinforma, 2012). Enterprises have dozens of different processes, such as processes for production, purchasing, and logistics. Despite their different natures, all enterprises share the same underlying transaction patterns, with similar coordination and production routines (Dietz et al., 2013). A *basic transaction pattern* is described in Figure 2-2. A coordination process begins when an initiator "requests" (rq) a product. The executor responds to the request by "promising" it (pm); then, the executor produces it. After he/she "states" (st) that the product has been produced, the initiator "accepts" (ac) the produced product in response to this event. The four *intentions* (i.e., rq, pm, st, and ac) are basic steps for a successful transaction. The effect of each intention leads to some state change in the world, e.g., "proposition requested" or "result produced" (Dietz & Hoogervorst, 2014b).

Details of DEMO aspect models are explained in chapter 3 by analyzing a real-life pizza store case.

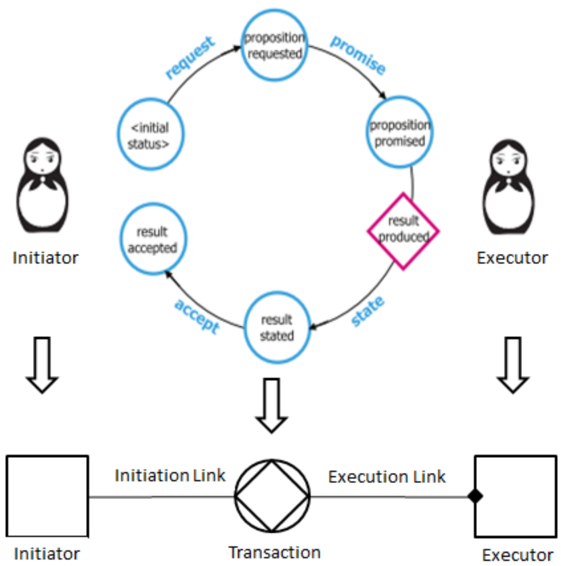


Figure 2-2 Basic transaction pattern

2.2.2 Business process simulation

Traditionally, static models, such as UML, BPMN, Flowchart, and IDEF, have been widely used to describe business processes. However, such models cannot add much value if they cannot assist in inspecting and analyzing business process performance. For example, if these models cannot provide the necessary means for bottleneck identification and for performance analysis or generate alternative improved business processes in terms of specified objectives, little value is added (Vergidis et al., 2002).

Toussaint et al. (1997) noted that a model should be functional, static and dynamic to best describe a business process.

Static modeling is widely used in describing workflow-based business processes. However, for the other perspectives of business processes, a more flexible dynamic and interactive method, such as simulation, is required.

2.2.2.1 What is simulation?

Simulation is a concept founded on systems. A *system* is a set of interrelated components that work together toward some common objective or purpose (Kossiakoff, Sweet, Seymour, & Biemer, 2011). To understand the operations of a system, a set of assumptions concerning how the system behaves are made (Law & Kelton, 2000). These assumptions, usually in the format of a physical model, a mathematical model or a combination of these models, are developed as a simulation model (Roberts et al., 1983). A *simulation model* is *"a representation of a real system that can be simulated by means of experimentation"* (Kleijnen, 2008).

Mathematical models can be used to analyze interesting questions when the relationships that compose a model are sufficiently simple. The solution can be attained via an algebraic method, calculus or mathematical theories, which consist of numerical parameters that are used as a measurement of the system (Law & Kelton, 2000). However, in most cases, the real world, which we want to evaluate, is too complex to allow realistic models to be evaluated analytically. In such cases, computer-based simulation is an appropriate method to numerically imitate the behavior of the system. *"Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and /or evaluating various strategies for the operation of the system."*(Shannon, 1998).

According to Harrison (2007), there are two ways for a scientist to analyze the real world: (1) deduction, which is a theoretical analysis that relies on a set of formulated assumptions, and (2) induction, which is an empirical analysis that relies on data that are analyzed to discover the relationships that exist among the variables. Simulation is considered to be a third scientific method (Axelrod, 1997). Simulation is an imitation of the operations of a real system. *"Simulations resemble deductions in that the outcomes follow directly from the assumptions made"* (without the constraints of analytic tractability). Simulations resemble induction in that the relationships among variables may be inferred by analyzing the output data (however, the data are generated by simulation programs rather than obtained from "real-world" observations)." (Axelrod, 1997)

The four most popular simulation methods include discrete event simulations, system dynamic simulations, agent-based simulations and hybrid simulations, which are a combination of these methods (Jahangirian et al., 2010). Details of each type will be discussed in the following sections.

2.2.2.2 Types of Simulation

Discrete Event Simulation:

Based on the review titled "*simulation in social science*" (Harrison et al., 2007), computer simulation began to be applied in management and operational research in the early 1960s, when it mainly consisted of discrete event simulations.

A *discrete event simulation model* is defined to be dynamic, stochastic, and discrete. This definition means that the system evolves over time (dynamic), and the state variables change instantaneously at separate points in time (stochastic) when the event occurs in some stochastic (discrete) time frame (Law & Kelton, 2000). The *model* described here has an additional definition: "*a representation of a system in terms of its entities and their attributes, sets, events, activities, and delays*", where the *event* is an "*instantaneous occurrence that may change the state of the system*" (Kheir, 1996). The most famous formalized discrete event specification is DEVS, which was described by Zeigler (2000). DEVS specifies a mathematical object as a system, with a time base, inputs, states, outputs, and functions for determining the next states and outputs from current states and inputs (Zeigler & Hammonds, 2007). There are two important concepts of a "system": decomposition and composition. *Decomposition* describes how a system can be broken down into component systems. *Composition* describes how the component systems can be coupled to form a larger system [Zeigler, 1984].

Discrete event simulation has been used to support the design and simulation of computer architectures, communications networks, manufacturing systems, workflows, management and other operational research areas (Gilbert & Troitzsch, 2005; Wainer & Mosterman, 2011). The most popular platforms include DEVS-JAVA, which was proposed by Zeigler for DEVS (Zeigler et al., 2000), Arena ("Arena Simulation," n.d.), which is a diagram notation platform for discrete event simulation, and AnyLogic (Technologies, 2009), which is a hybrid simulation method.

System Dynamic Simulation:

System dynamics focus on modeling the behavior of an entire system by applying different equations to plot the temporal trajectories of variables and their influence on each other (Gilbert & Troitzsch, 2005). System dynamics was created during the mid-1950s by Professor Jay Forrester of MIT. This method represents a causal relationship that describes interactions between the components of a system. By capturing feedback loops, a causal loop diagram reveals the structure of the system to ascertain the system's behavior.

Agent-based Simulation:

Although simulation has become one of the most widely used tools in the natural sciences, it was not used as much in the social sciences in the 1980s. Social scientists appear more concerned with understanding and explaining an existing situation than making predictions. Agent-based modeling and simulation filled this gap when it was introduced in the early 1990s.

Agent-based simulation is a new approach in simulation research that can be used to simulate interactions between autonomous objects. This approach is defined as follows: "*the process of designing an agent-based model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/ or evaluating various strategies for the operation of the system*" (Shannon, 1975). In an agent-based model, the system is a set of agents that follow some specific behavior rules, and the properties of the system are generated from agent interactions (Bonabeau, 2002). The advantages of using agent-based simulations are that *"agent-based models can explicitly model the complexity arising from individual actions and interactions that arise in the real world"* (Siebers, Macal, Garnett, Buxton, & Pidd, 2010).

Although there is an increasing amount of literature about designing and using agents in various social science studies, no generally agreed upon definition of an agent exists. Bradshaw (1997) defined agents as *"objects with attitudes"*. According to the other researchers (Wooldridge & Jennings, 1995), agents are autonomous and self-controlled in terms of their actions and their internal state, interact with other agents as a social system, and have the ability to perceive their environment (the system or the other agents) and to react on it. As a subjective object in the system, agents are goal-directed that they could perform initiatively. Most agent models are production systems with three components (Gilbert & Troitzsch, 2005): (1) a set of rules in which each rule includes a condition part and an action part, (2) a working memory that stores the facts that are produced by the agent, and (3) a rule interpreter that checks each rule in turn to see whether the conditions of the rule are met. If the conditions are met, the corresponding action is carried out.

Based on the literature (Chan, Son, & Macal, 2010; Gilbert & Troitzsch, 2005; Harrison et al., 2007; Jahangirian et al., 2010; Melão & Pidd, 2000; Siebers et al., 2010), the characteristics and application areas of three mainstream simulation methods are summarized and listed in Table 2-2.

2.2.2.3 Business Process Simulation

The power of simulation enables the generation of new ideas for change, the exploration of the effects of alternative changes, the implementation of those changes without disrupting the business system, and the comparison of the performance of both the present and reengineered systems (Greasley & Barlow, 1998). Bell et al. (Bell, Raiffa, & Tversky, 1988) discussed the generally high level of support and interest in visual interactive models among decision-makers. MacArthur, Crosslin, and Warren (Warren, Crosslin, & MacArthur, 1995) first began to investigate the suitability of applying a discrete event simulation model for a business process reengineering project for which they recognized that *"a key component of practical implementation of BPR is measurement"*. The model was constructed using the ARENA modeling system (Pegden, et al., 1995), which is based on the SIMAN/CINEMA system cited in (Greasley & Barlow, 1998).

Table 2-2 Simulation methods

Kye words	Discrete Event Simulation Model	System Dynamic Model	Agent Based Simulation Model
Model type	• Dynamic , Stochastic; • Discrete	• Dynamic , Stochastic; • Continuous	• Dynamic, Stochastic,; • Discrete: if state valuables change only at discrete time instant; • Hybrid of discrete and continuous: if there is any state variable change continuous;
Focus	• Process centered: Top-down modelling approach • Focusing on modeling sequence of a system (sequence of events or activities)	• Policy and strategy development	• Entity centered: Bottom-up modelling approach; • Focusing on modeling entities` behavior and their interactions
Strength in modelling different system	• Queuing system and queuing network • Resource Optimization • It is not powerful in simulating frequently interaction of processes, though it can.	• information feedback structure of business process • causal relation of a complex system	• Social system composed of different type of agent • System with highly interacted agents or processes
Key Concepts	• Queuing system ; • Events and activities in sequence; • Macro behavior is modeled; • Intelligence is modelled as part of system, process decision • Passive Entity, something is done on the entities when they pass through the system;	• Causal loop define structure of a system, how the constituent components interconacted • To ascertain system's behivour over time by understanding the structure of a system	• No concept of queue; • No concept of sequence; • Micro behavior is modeled • Intelligence is represented within each agent, agent decision • Active Entity, Entities themselves are goal oriented agents that can initiating activities;
Some Application Area	• Operational research • Healthcare	• Strategic decision making level • High level perspective analysis • Qualitative analysis	• Economics • Management
Limitations	• No human and organization view • Time consuming and skill requirment • No feedback loop	• Little evidence • Difficult to derive correct equations	• Unable to give qualitative assigment in BPR • Difficult to be linked with process

Simulation is currently considered to be a key technique for business process improvement and reengineering. The technique of using simulation in the context of a business process is referred to as *business process simulation* (BPS) (Du, Gu, & Zhu, 2012). BPS is not a methodology in itself; instead, it is a technique that can be used to support a chosen methodology (Robinson, Nance, Paul, Pidd, & Taylor, 2004). BPS can assist BPR because they have several common characteristics. For example, they both can be used to analyze the interrelationships between activities within a process (Greasley, 2003).

As shown in Figure 2-3 which includes both process design and improvement, accounts for the second most popular application of simulation models (Jahangirian et al., 2010); only scheduling is ranked higher.

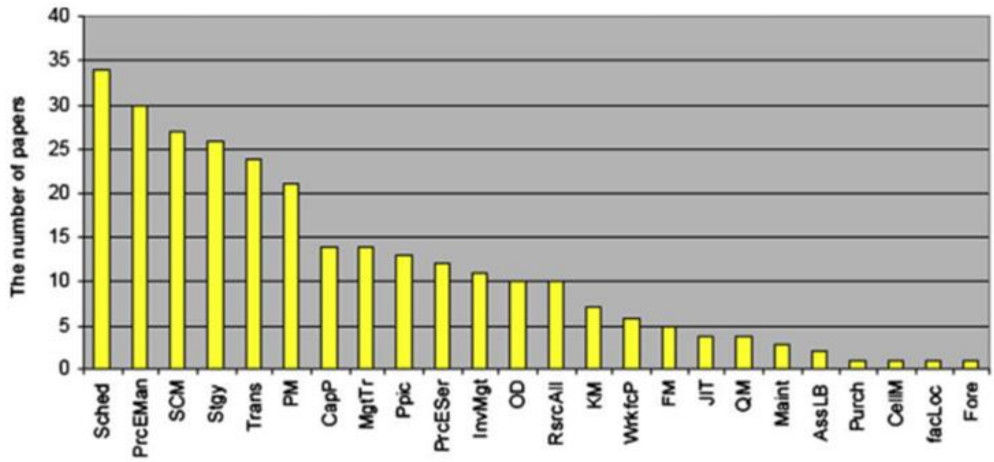


Figure 2-3 Simulation areas (Jahangirian et al., 2010)

2.2.2.4 Limitations of BPS Research

BPS is a powerful tool that can assist in change analysis and effectiveness evaluation due to its ability to measure performance, to test alternatives and to engage in processes (Greasley & Barlow, 1998). However, there are several barriers that prevent BPS from being widely used in business process change analyses.

- **Limitation 1: current simulation methods are weak at describing large, complex systems.** As some researchers have noted (Jahangirian *et al.*, 2010; Barber *et al.*, 2003), most business process improvement projects consider only a single process without taking a holistic perspective of the enterprise; the complexity of simulation increases when individual small process models are joined into a large hierarchical construct. Thus, it is inefficient to utilize these methods in process change analyses that concern an entire enterprise.
- **Limitation 2: simulation models are difficult to introduce and use in management.** Business process simulations have always required a high level of skill to make simulation models (Paul, Giaglis, & Hlupic, 1999). A major barrier that prevents many organizations from using BPS is the preparation needed to successfully introduce the technique to the organization (Greasley, 2003). As a result, detailed plans have been provided for introducing and using simulations in an organization (Harrington & Tumay, 2000). In addition, Harrison argued that a *"focus on processes, which do not affect the firm's strategic future, misdirects scarce resources into doing the wrong things right, or into reengineering*

processes in a way that is insensitive to their competitive contribution" (Harrison et al., 2007). A more important objective is to link strategy change with business process redesign. However, the most commonly used decision-making tool in management is "Microsoft Excel". Agent-based simulations, discrete-event simulations and other simulation methods are not well integrated with management in terms of supporting decision-making or strategy changes.

- **Limitation 3: changing models are very complex when simulating new designs in BPR** (Greasley, 2003). Simulation is a useful tool in comparing 'as-is' and 'to-be' models to validate the effects of change and to ensure the completeness of a model; however, simulation has a limited ability to design a 'to-be' model. Most of the business process simulation literature restricts itself to comparing the before and after conditions, providing little information about the redesign process (Reijers & Liman Mansar, 2005). Furthermore, due to the embryonic state of business process modeling research, little theoretical and methodological support has been provided to explain business process change and the consequent requirements for simulation model changes (Bosilj-Vuksic, Ceric, & Hlupic, 2007), i.e., methodologies for BPS modeling remain weak in their ability to support business process changes.
- **Limitation 4: unstructured modeling complicates confirmation of consistency.** As Valiris *et al.* (2004) noted, most reengineering methodologies *"lack a formal underpinning to ensure the logical consistency of the generation of the improved business process models"*. This issue leads to a lack of a systematic approach that can guide a process designer through a series of repeatable steps to achieve process redesign. Vergidis *et al.* (2008) argued that *"a structural and repeatable methodology that could be generally applied to business process modeling and improvement was never established"*.

These limitations are all due to the weakness of conceptual modeling. Conceptual modeling is regarded as the most important and difficult step; it is also the least investigated step in simulation, especially in BPS. As indicated by Bank et al. (2013), there are surprisingly few books and academic papers on the subject of building conceptual models for enterprise-related simulations. Robinson (2006) defined conceptual modeling as the *"representation of the abstracted world, expressed by means of diagrams and written text"*. Turnitsa et al. (2010) provided an extended definition: *"a formal specification of a conceptualization"* and *"an ontological representation of the simulation that implements it"*. One example of an ontology-based conceptual model for simulation is the system entity structure (SES) that was proposed by Zeigler (2000), which was utilized in several of Zeigler's discrete event simulation studies. However, SES emphasizes only the system's data structure. Therefore, it is poor at describing enterprises as social systems and is less applicable in the context of business processes.

In contrast, most conceptual modeling methods used in process-related simulations do not consider the ontology level. Instead, they consider the implementation level of enterprises, such as flowcharts, BPMN, UML or onto-UML (Guizzardi & Wagner, 2012). Wang and Brooks (2007) showed that the most widely used representation technique for BPS is the flowchart (used by 63% of simulation modelers). Other modeling methods, such as BPMN, UML, and IDEF, are also widely used. All of these modeling tools are based on a workflow perspective, which aims to represent the sequences used in real-world work. Thus, workflow-based discrete event simulations are more widely used in BPS than agent-based simulations. Nevertheless, the latter is considered a promising method for analyzing enterprises as social systems (Siebers et al., 2010). It is difficult to analyze tens of A4-paper-based flowcharts or UML models to understand why certain actions are performed, what relationships exist and, correspondingly, how to make changes that can confirm the consistency of a system before and after changes. By default, a workflow-based simulation has the same problem, i.e., it occasionally delves too deeply into the process details to answer the "how to" question and is weak in analyzing the "why" and "what" questions. Moreover, models that stem from a workflow perspective are typically non-modularized; instead, they are described as a sequence of activities. The workflow perspective neither reduces the complexity of modeling nor facilitates change. Workflow-based BPS is not an adequate solution for supporting enterprise reengineering. They are widely used because the simulations are performed entirely at implementation levels with which they best match. However, as noted in multiple studies (Chen, 1976; Salimifard & Wright, 2001), these models do not describe enterprise functionality. Thus, they cannot be used by management for decision-making support. Moreover, there are no clear definitions of highly abstracted enterprise ontology. These weaknesses make it difficult or impossible to apply one conceptual model with different implementations to support the same ontological structure, although this is required in business process redesign and reengineering. Furthermore, an unstructured and non-ontological conceptual model leads to low modularity, low reusability and uncontrollable changes in simulation modeling, especially when the simulation is employed for BPR.

CHAPTER 3 Enterprise Ontology – DEMO

DEMO is based on *ψ theory*, which considers an organization to be an interaction of individual social subjects. To obtain a full representation of a system, an ontological model is divided into four sub-models that describe different aspects of a complete model (Figure 3-1). A *construction model* (CM), located at the top of the triangle, is the most concise model and describes how transactions and actor roles are composed to construct a system. A *process model* (PM) describes the detailed causal relationships and constructions that exist in processes. A *fact model* (FM) describes the objects and facts that are related to a process. Lastly, an *action model* (AM) describes the action rules for the actor roles. Using these models, DEMO proposes a consistent, coherent, concise, comprehensive and essential (C4E) representation of an organization.

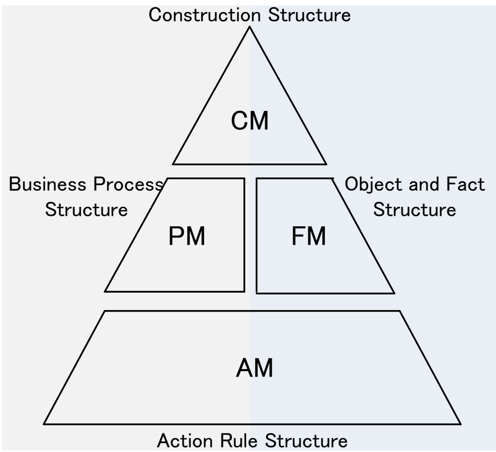


Figure 3-1 DEMO aspect models

Details of the DEMO methodology are explained through a case analysis that is presented in this chapter.

3.1 Case Description (Case C)

Buono, a pizza shop, is located in a small city in Japan and employs one manager, Mr. Inoue, and four young adults on a part-time basis.

They use an order-taking system to help manage the orders. Thirty percent of the orders are taken by phone[1], 60% are from the Internet, and only 10% are directly handled at the store counter. For all three methods, ordering[2] takes a minimum of 1 min, 3 min on average, and occasionally up to 5 min.

The order is entered in the IT system, an empty box is pulled, and a label with the customer's name[3], address, order and phone number printed on it is applied to the box (this duration is included in the ordering time). Normally, an available worker reads the waiting orders in the IT system and prepares the pizza dough. This step takes an average of 2 min. The staff then adds the requested toppings; this step takes an average of 3 min. This step can only be performed if a table is available; the table can accommodate 4 pizzas. Three ovens are available. Baking takes 6 min, and the time to place the pizza in the oven is negligible.

When the pizza exits the oven, it rests on the packing table. An available worker places the pizza in the box with the label printed on it. This step takes an average of 1 min. Then, an available deliverer will deliver the prepared pizza to the customer's location and receive payment. Buono's delivery area is within a driving distance of 10 min (5 to 10 min one way). The 4 staff members in the store can perform any task, but at least 1 staff person always remains in the store.

As a courtesy, if the wait time from placing the order to delivery is more than 30 min, Buono will provide a free pizza with the next order. However, Mr. Inoue noted that too many customers were not receiving their orders within 30 min (more than 20%). Therefore, solutions to reduce the free pizza requirement were needed. He hoped that simulation could provide some advice.

---

[1] In the description, the data-level activities are denoted in blue;

[2] In the description, the business-level activities are denoted in red;

[3] In the description, the information-level activities are denoted in green.

3.2 DEMO Aspect Models

3.2.1 DEMO construction model (CM)

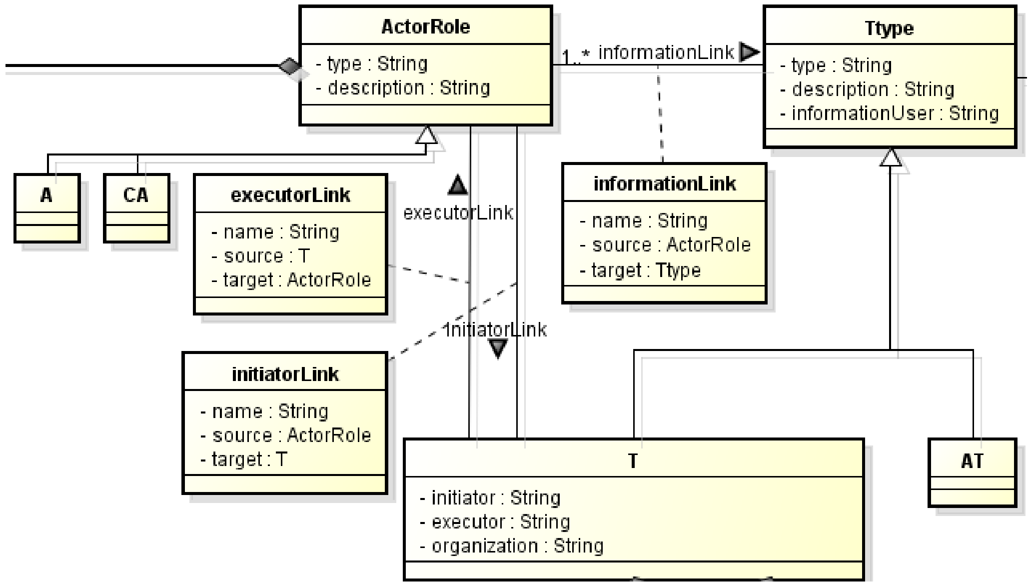


Figure 3-2 Meta-model of the DEMO construction model

The meta-model of the DEMO construction model is presented in Figure 3-2. In DEMO, every transaction is of some type, called a *transaction type* (Ttype for short). Each Ttype involves two *actor roles*, who are authorized to commit or produce the facts generated by the transactions. An actor role acts as either the initiator or executor of a transaction. Actor roles are either elementary or composite. *Elementary actor roles* (A for short) are actor roles within the system of focus, and they can be the executors of only one Ttype. *Composite actor roles* (CA for short) represent actor roles that are not focused. A CA can execute more than one Ttype. Ttypes and related actor roles (A and CA) are described in the *Organization Construction Diagram* (OCD) of the DEMO CM. As shown in Buono's OCD (Figure 3-3), the author abstracted four Ttypes: T1, T2, T3 and T4. A1 (the order completer) is an elementary actor role as the executor of T1 and the initiator of T2, T3 and T4. CA1 (the customer) is a composite actor role because his/her behavior is out of the scope of focus. Ttypes and actor roles are connected by two types of links. An *initiator link* is a link from an initiator (source of the link) to its Ttype (the target of the link), represented as a line. An example is the link from CA1 to T1 shown in Figure 3-3. An *executor link* is a link from a Ttype (the source of the link) to its executor (the target of the link), represented as a line with a black diamond at the end. An example is the link from T1 to A1 shown in Figure 3-3.

Aggregate transaction type (AT for short) represents a Ttype that belongs to the system environment. For Buono, AT1 contains map information that is obtained externally for delivery. All generated facts (cf. section 3.1) are stored as either Ttypes or ATs. These Ttypes and ATs are information banks. The actor roles can access the information banks to establish information. Links between actor roles and information banks are called *information links*, for example, the link between A3 and AT1. Initiator and executor links can both serve as information links.

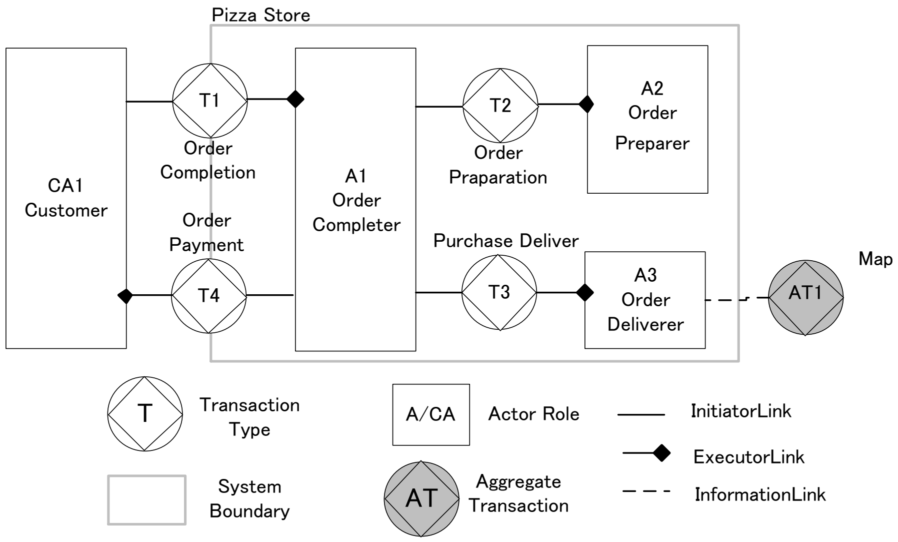


Figure 3-3 OCD for Buono pizza

Transactions of the same type concern products of the same type, called *product type* (P for short). The Ttypes and the corresponding Ps are described in the *Transaction Product Table* (TPT) of the CM. As shown in Buono's TPT (Table 3-1) there are four product types, P1, P2, P3 and P4, defined according to the four Ttypes.

Table 3-1 Buono Pizza's transaction product table

Transaction Type	Product Type
T1 Order Completion	**P1 Order** has been completed
T2 Order Preparing	**P2 Order** has been prepared
T3 Order Delivery	**P3 Order** has been delivered
T4 Order payment	**P4 Order** has been paid

3.2.2 DEMO process model (PM)

CM briefly describes how an organization is constructed. The *process model* (PM) located below the CM in Figure 3-1 describes transaction details and their inter-relationships.

In the PM, transactions are expanded into transaction patterns, in which the routines and effects of acts (described as facts) are defined. There are three types of transaction patterns: basic, standard and complete.

A *basic transaction pattern* describes the simplest "happy path" to accomplishing a transaction, including the following intentions: rq, pm, st, and ac.

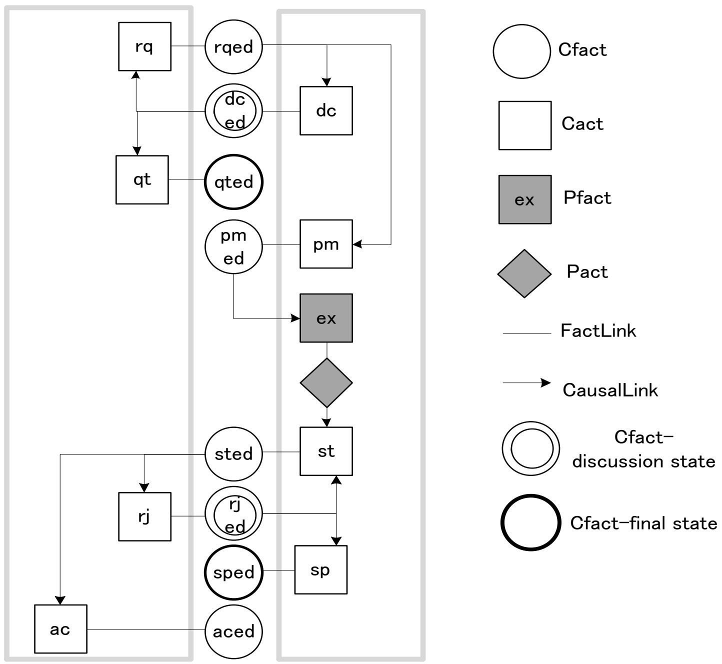


Figure 3-4 Standard transaction pattern

A *standard transaction pattern* considers both the "happy path" and exceptions in which the transaction may be stopped or redone. In addition to the basic transaction pattern, there are four new intention types: {decline (dc), quit (qt), reject (rj), stop (sp)}. As described in Figure 3-4, a transaction begins when its initiator requests a product. The executor responds to the request with a decision on whether to promise or decline the request following some action rule. If a request is declined, the process will enter a negotiation stage, and as a result of negotiation, the initiator can choose to re-request or cease cooperating. The situation is the same when a statement is rejected by the initiator: the executor can choose whether to restate or stop the process (Dietz, 2006). In transaction patterns, a link from an act (source of the link) to a fact (target of the link) is called a *fact link*, indicating that one fact is the effect of the act, represented as a line. A link from a fact (source of the link) to an act (target of the link) is called a *response link*, indicating that an act is the reaction to the fact, represented as an arrow line.

A *complete transaction pattern* concerns not only the "happy path" and exceptions but also cancellations of request, promise, state and accept. To simplify the problem, we use the standard transaction pattern to define the possible states of a transaction in this research. The meta-model of the DEMO PM is presented in Figure 3-5.

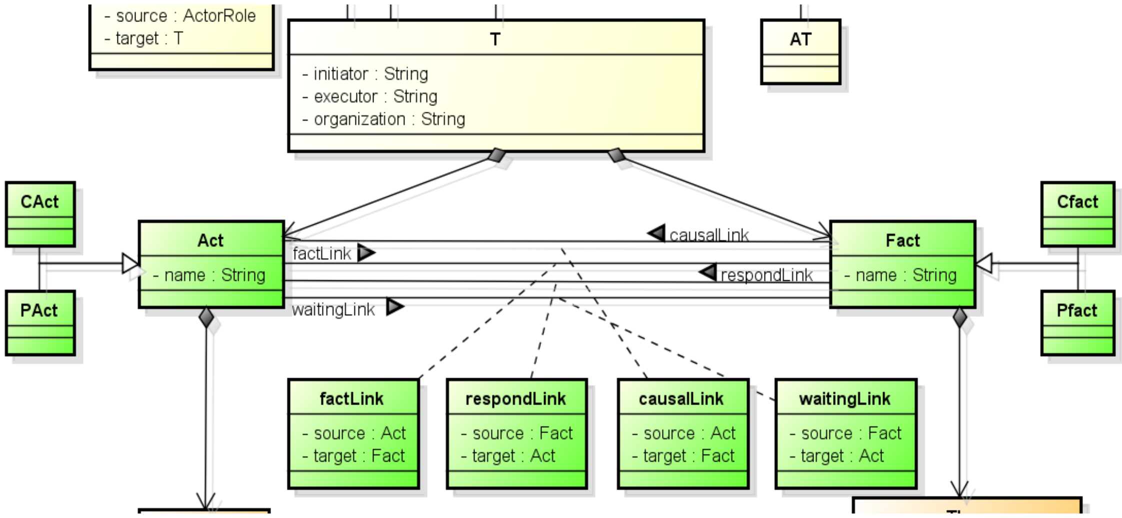


Figure 3-5 Meta-model of the DEMO PM

In the PM, the waiting and causal relationships between transactions are described in the *Process Structure Diagram* (PSD) shown in Figure 3-6. The *waiting relationship* indicates the conditions of an act. As represented in Figure 3-6, acts (expressed as small rectangles) of a Ttype may need to wait for some facts (expressed as small circles) to be created. For example, the act "execution of order delivery" [T3/ex] must wait for the fact "pizza preparation accepted" (T2/ac). The waiting relationship is represented by a dashed arrow from the fact (source of the link) to the act (target of the link); this arrow is the *waiting link*. A *causal relationship* indicates that a fact causes another Ttype to be initiated. As represented in Figure 3-6, when the order for completion is promised (T1/pm), new Ttypes T2 (order preparation), T3 (order delivery) and T4 (order payment) are initiated. Causal relationships are represented by an arrow from the fact (source of the link) to the act (target of the link); this arrow is the *causal link*. In the PSD, acts and facts are described only when there are causal or waiting relationships between them. The other acts and facts are hidden within the Ttype as part of the transaction pattern.

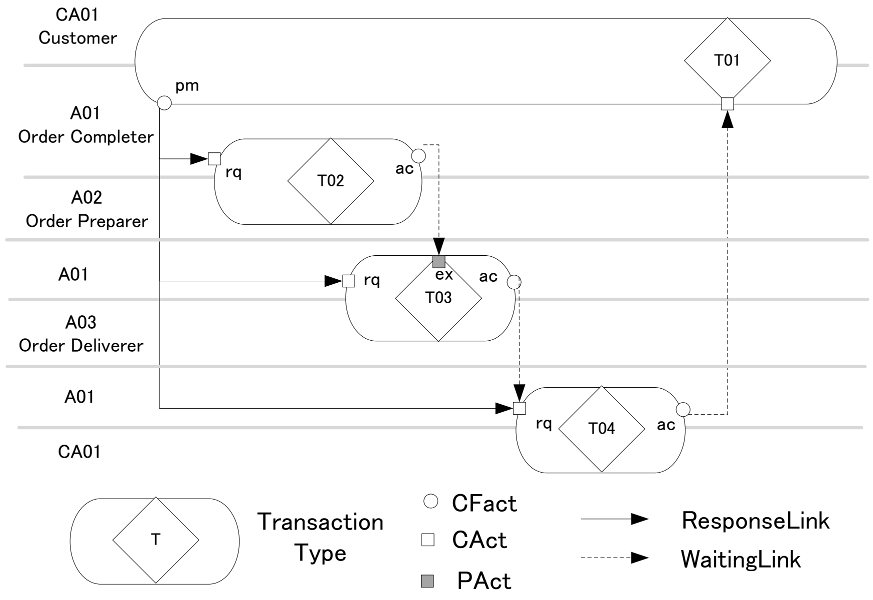


Figure 3-6 PSD of Buono pizza

3.2.3 DEMO fact model (FM)

The PM takes the process and state view when analyzing an organization. The *fact model* (FM), which is located at the same level as the PM, describes a different aspect: the object and fact structures represented in the *Object Fact Diagram* (OFD). The meta-model of the DEMO FM is as defined in Figure 3-7.

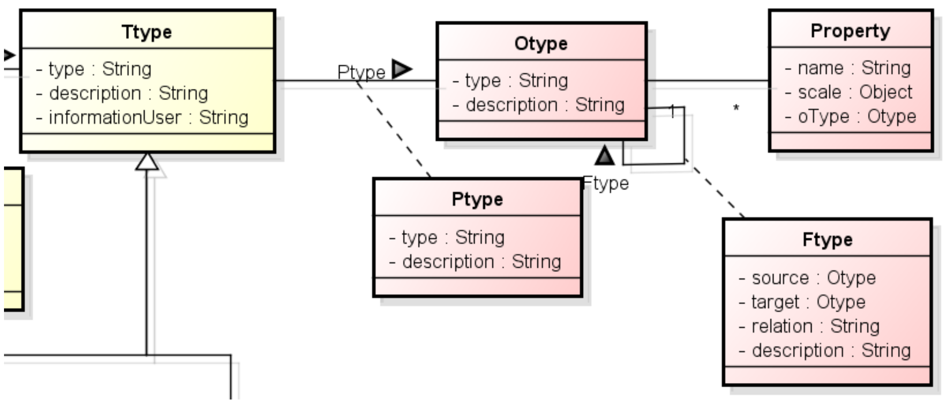


Figure 3-7 Meta-model of the DEMO FM

As shown in Figure 3-8, an *object* is an identifiable individual component. Objects are always of some type, called the *object type* (O for short). For example, in the case of Buono, “Order”, “Person” and “Pizza” are the object types. A relationship between object types is called a *fact type*. In the example, a fact type between “Order” and “Pizza” is “order O contains Pizzas P”. An instance of fact type expresses an elementary state of the world. Fact type is represented as an *OO link* in the OFD.

When an object type is related to a transaction’s production process, it is connected with the responding product type, representing possible stages of the object type. For example, “Order” is involved in all four production processes, T1, T2, T3 and T4. Thus, the possible stages of “Order” are P1, P2, P3 and P4, meaning that an order must be prepared (P2), delivered (P3), paid for (P4) and then completed (P1). This relationship is represented as the *OP link* in the OFD.

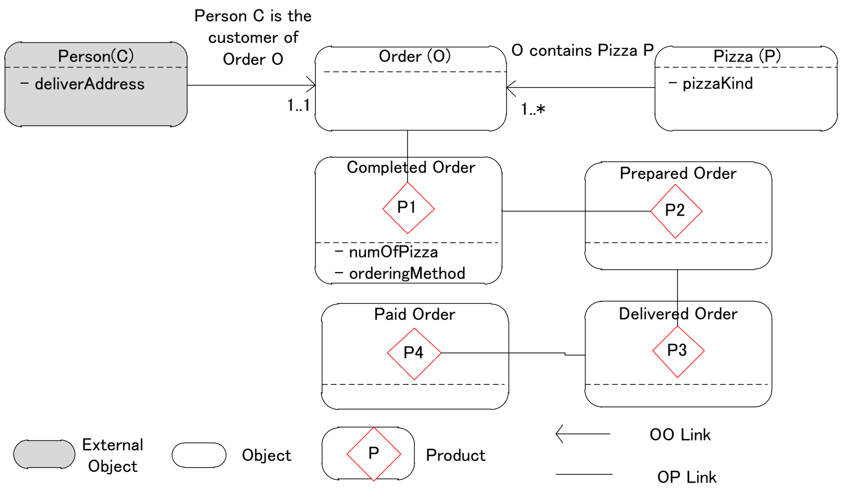


Figure 3-8 OFD for Buono pizza

3.2.4 DEMO action model (AM)

An *action model* (AM), which is located at the bottom of the triangle shown in Figure 3-1, describes action rules. An action rule is expressed as a *crispie*. Crispies are based on finite automaton theory (Hopcroft, Motwani, & Ullman, 2006), which entails a finite set of states and a finite set of state transitions.

A *crispie* is formally defined as a tuple $< C, R, I, S, P >$, *where*

- C : a set of C-fact types, which is called the coordination base

- R : a set of action rules, which is called the rule base
- I : a set of intentions, which is called the intention base
- S : a set of C-fact types and P-fact types, which is called the state base
- P : a set of product kinds, which is called the product base

$$R: \wp C * \wp S \rightarrow \wp (I * P * T * D)$$

*($\underline{C}$ is the union of the extensions of a set of concept types, and $\wp X$ is the power set of a set X)

With crispies, the world is in some state at every point in time. The *state* (S) is defined as a set of facts including both Cfacts and Pfacts. At any moment, the crispie releases an *agenda* (tasks to perform), each item on which is a pair, for instance, <c, t>, where 'c' is an instance of *Cfact*(C), e.g., (T3(o)/rq). 't' is the settlement time (T), at which point the creator will expect the event to be responded to by a crispie (e.g., t_{T3rq}). The crispie responds to the event by evaluating a particular function, which is called a *rule base* (R). The evaluation result is a set of *intentions of reactions* (I) with *productions* (P) in a *duration* (D) (Dietz, 2006).

A specification of a crispie for the actor role A3 is as follows:

```
C3= {Order Delivery for Order (o) is requested((T3(o)/rq) for short),
    Order Delivery for Order (o) has been promised  ((T3(o)/pm) for short),
    Order Delivery for Order (o) has been executed  ((T3(o)/ex) for short)}
I3 = {[T3/pm], [T3/ex], [T3/st]}
S3={Map for delivery Order (o) has been prepared,
   Delivery of Order (o) can be promised}
P3={Pizza Delivered }
R3:
   WHEN (T3(o)/rq),
     IF Delivery of Order (o) can be promised,
     THEN [T3(o)/pm]
   WHEN (T3(o)/pm),
     IF Map for delivery Order (o) has been prepared,
     THEN [T3(o)/ex]
     And [T3(o)/st]
```

Based on crispie, to describe the rules, the author mapped C, S, I and P into rule expressions and defined a Meta-model of the DEMO AM as given in Figure 3-9.

- C corresponds to the ‘facts’ that are defined in ‘When’. Each ‘Act’ is reaction to the facts;
- S corresponds to ‘WhenWith’, in which the conditions of this reaction, if any, are defined;
- I corresponds to ‘Act’; and
- P corresponds to ‘Then’ and ‘ThenWith’. *Then* defines whether there is some change that needs to be processed. If a chance needs to be processed, the conditions and results of the state change will be expressed in ‘ThenWith’.

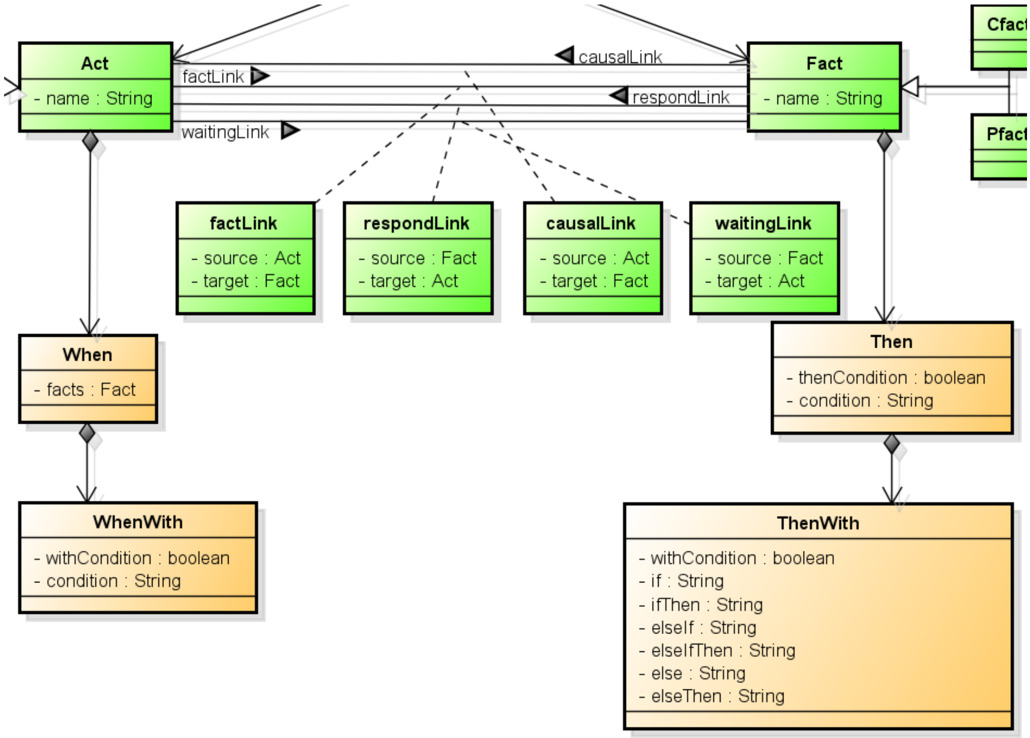


Figure 3-9 Meta-model of DEMO FM

Based on our revised definition in the Action Rule model, an action rule table is designed, as shown in Table 3-2. In the table, “When” corresponds to the “When” part of the AM; “Then” corresponds to the “Then” part of the AM; and “WhenWith” and “ThenWith” are combined in the WITH block, which denotes the “With” parts in “When” and “Then”. Thus, the specification of a crispie is translated into an action rule table.

Table 3-2 Action rule table

A1	WHEN					THEN						WITH				
	FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
	When	rqedT1	For	Order	is done.	If	none	Then	pmT1	For	Order	With	If	none	Then	none
	When	pmedT1	For	Order	is done.	If	none	Then	none	For	none	With	If	none	Then	new T2(case); new T3(case); new T4(case);
								Then	rqT2, rqT3	For	Order	With	If	none	Then	none
	When	pmedT1,acedT4	For	Order	is done.	If	none	Then	exT1	For	Order	With	If	none	Then	none
								Then	stT1	For	Order	With	If	none	Then	none
	When	stedT2	For	Order	is done.	If	none	Then	acT2	For	Order	With	If	none	Then	none
	When	stedT3	For	Order	is done.	If	none	Then	acT3	For	Order	With	If	none	Then	none
	When	acedT3	For	Order	is done.	If	none	Then	rqT4	For	Order	With	If	none	Then	none
A2	WHEN					THEN						WITH				
	FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
	When	rqedT2	For	Order	is done.	If	none	Then	pmT2	For	Order	With	If	none	Then	none
	When	pmedT2	For	Order	is done.	If	none	Then	exT2	For	Order	With	If	none	Then	none
								Then	stT2	For	Order	With	If	none	Then	none
A3	WHEN					THEN						WITH				
	FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
	When	rqedT3	For	Order	is done.	If	none	Then	pmT3	For	Order	With	If	none	Then	none
	When	pmedT3,acedT2	For	Order	is done.	If	none	Then	exT3	For	Order	With	If	none	Then	none
								Then	stT3	For	Order	With	If	none	Then	none
CA1	WHEN					THEN						WITH				
	FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
	When	rqedT4	For	Order	is done.	If	none	Then	pmT4	For	Order	With	If	none	Then	none
	When	pmedT4	For	Order	is done.	If	none	Then	exT4	For	Order	With	If	none	Then	none
								Then	stT4	For	Order	With	If	none	Then	none
	When	stedT1	For	Order	is done.	If	none	Then	acT1	For	Order	With	If	none	Then	none

CHAPTER 4 Business Process Change Analysis in the Context of Enterprise Engineering

4.1 Introduction

In the context of enterprise engineering, as described in Figure 4-1, the world is constructed by two systems: a using system (US) and an object system (OS) (Dietz, 2006). In the *using system*, the functional requirements need to be designed to explain "what". It is a black box model, which means that you cannot evaluate the objects inside. Functional requirements are supported by a specific construction in the object system. The construction is a white box model, which explains "how". For example, how could the functional requirements have been supported? In enterprise engineering, the construction is explained with a high abstracted model, enterprise ontology, which describes only the essence of the system without including any implementation details. The construction design process, which converts function into construction, is related to the enterprise's architecture.

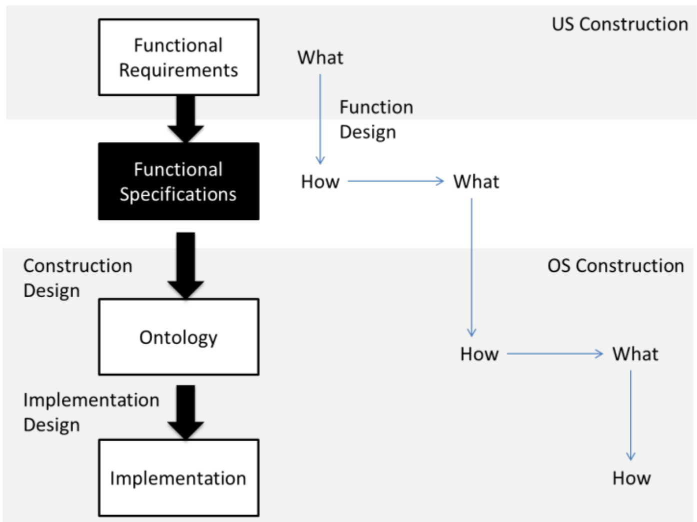


Figure 4-1 Function and construction (ref. Dietz, 2006)

4.2 Types of Construction Changes

As portrayed in Figure 4-1, ontology is the first level of construction. Implementation that supports the ontology is the second level of construction.

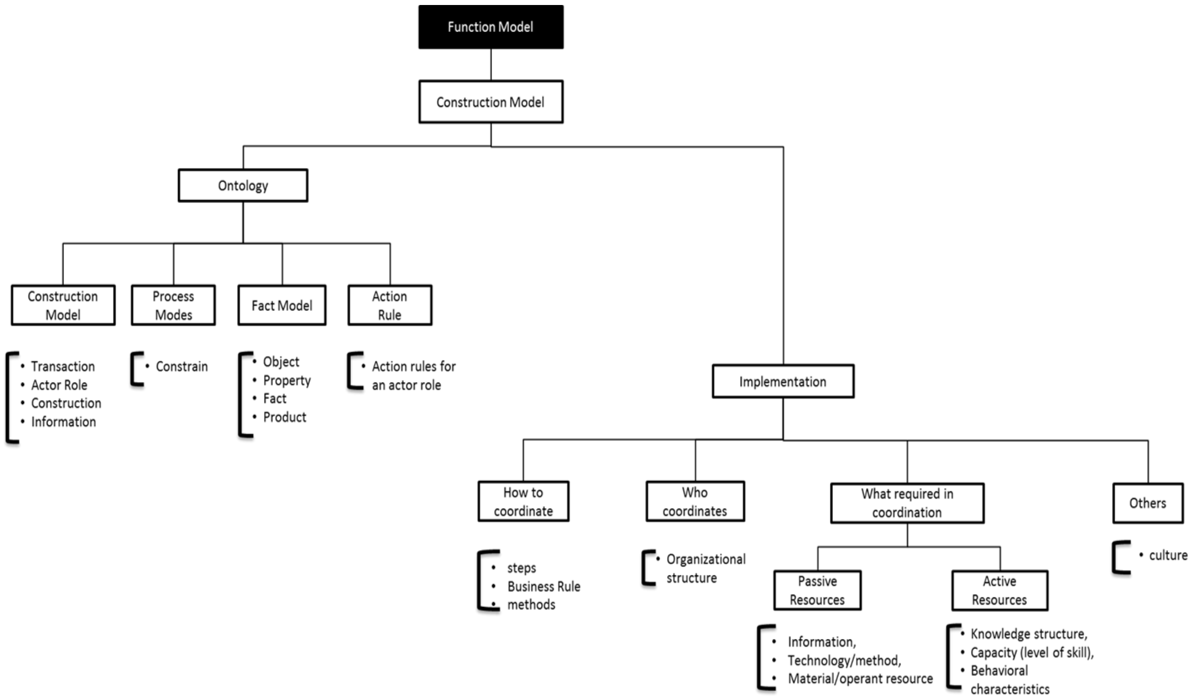


Figure 4-2 Key concepts in construction level

- Ontology: A high level of construction that describes abstractly how an enterprise is constructed to realize a strategy. In particular, the goal of enterprise ontology is to "*offer a new understanding of enterprises, such that one is able to look through the distracting and confusing actual appearance of an enterprise right into its deep essence*" (Dietz et al., 2013). As shown in left side of Figure 4-2, this research employs DEMO as ontological model, describing an enterprise from four aspects: (1) transaction, actor role, construction and information bank in the construction model; (2) constraint of process in the process model; (3) objects, properties of objects, relations between objects and possible status of objects in the fact model; (4) action rule of each actor role in the action model.
- Implementation: A detailed level of construction that describes how an enterprise is constructed to realize an ontology. In enterprise ontology, Dietz proposed a three layer enterprise, top-down including Business-level organization, Information-level organization and Data-level organization. The lower layer supports the realization of upper layer. Enterprise ontology focuses on the top layer, the business-level of enterprise, describing "why coordinate" question. However, there are the other supportive activities that related with information-level activities, e.g., calculation, re-construction etc.; or data-level activities, e.g.,

gathering information, exchange information etc. Those are realizations of the coordination. By concluding the concepts in enterprise architecture and enterprise governance researches of enterprise engineering (Hoogervorst, 2009; Op`t Land, Proper, Waage, Cloo, & Steghuis, 2009; Op`t Land, 2006), this research considered implementation from the following aspects:

- How to coordinate? This question is related to the methods and steps at the information level and the data level that support the business-level realization.
- Who coordinates? This question is related to the organizational structure, e.g., actors who play the actor role at the ontological level;
- What is required in the coordination? This question is related to the resources required for coordination.
  - Passive resources: information, technology/methods, and materials/resources.
  - Active resources: knowledge structures, capacity (level of skill), and behavioral characteristics.
- Other aspects related to culture, knowledge structure, etc.

Functional changes are primarily related to strategy changes, business model changes or vision and mission changes. Any type of change will ultimately be translated into functional specifications that express the functional requirements.

Table 4-1 Types of changes

Level		Models		Description	Types of Changes
Function	Functional requirement				
Construction	Ontology	CM		Construction model	Type 1.1
		PM		Process model	Type 1.2
		FM		State model	Type 1.3
		AM		Action Rule	Type 1.4
	Implementation	Coordination	How to coordinat	Processes, Business Rule Quality and performance	Type 2.1
			Who coordinate	Organizational structure	Type 2.2
		Resources for coordination	Passive resources	Information, Technology/method, Material/operant resource	Type 2.3
			Active resources	Knowledge structure, Capacity (level of skill), Behavioral characteristics	Type 2.4
		Context	Context	Culture	Type 2.5

As described in section 4.1, all of the requirements are described as by a "black box". The functional requirements are realized by a construction model. According to the keys defined for construction shown in Figure 4-2, this research defines nine types of changes within two levels of construction, which are shown in Table 4-1 and are explained in the following sections.

4.2.1 Changes in ontology level

Ontology changes are related to the "change what to do" to fulfill a particular set of requirements. Corresponding to the DEMO definitions, Type 1 ontological changes include four subtypes: Type 1.1, which includes construction changes in DEMO CM, Type 1.2, which includes process changes in DEMO PM, Type 1.3, which includes fact and object changes in DEMO FM, and Type 1.4, which includes action rule changes in DEMO AM.

An enterprise is a system with a boundary. The world outside the boundary is called the environment of the system. In DEMO, the transactions that connect an enterprise with the actor roles outside a boundary are called *boundary transaction types*. Moreover, transactions that occur inside this boundary are called *internal transaction types*. Internal transaction types are transactions that support the boundary transactions to be performed. In most cases, people are interested in reducing exceptions to boundary transaction types because they are more closely related to financial performance and the effective and efficiency of the entire enterprise.

In fact, this research considers all limitations, bottlenecks and other issues are ultimately related to a particular boundary transaction problem. Based on this perspective, boundary transactions may not be the direct cause of an enterprise change; however, they are regarded as one of the internal driving forces. Therefore, solving these types of problems may provide opportunities to improve a business process. Hence, it might enable business process change. In the context of enterprise engineering, the author considers these problems from three aspects: unhappy paths, complaints and delays.

- **Unhappy paths:** As shown in Figure 3-4 of section 3.2.2, "unhappy paths" in a DEMO transaction pattern include the following: "when a request is not promised, but declined", "when a statement is not accepted, but rejected" and “cancellation”, “quit”, and “stop”. Unhappy paths always lead to stops, loops or even cancel and quits. For example, a customer might reject some delivered products and/or services because he/she is not satisfied with the quality of the products and/or services. Then, the customers and the enterprise may need to negotiate to solve the problem. These negotiations and re-dos may use extra time, resources, managerial efforts, and costs.

- **Complaints:** A complaint refers to all of the unsatisfied parts of a boundary transaction type. Although a complaint follows a happy path to the end of the transaction, it may become the reason for an unhappy path next time or lead to some lost business opportunities. Complaints should be treated seriously as an unhappy path.
- **Delays:** There is no problem that leads to an unhappy path or a complaint. To increase competitiveness in the market, a company may want to reduce the delay time associated with some boundary transaction. Delays may also lead to enterprise changes.

Ontology changes include four sub-changes that correspond to four different aspect models. When the current construction is not sufficient, some supportive or management/pre-decision transaction types are required to gain assurance that there will be fewer problems associated with a particular type of boundary transaction. Additional boundary transactions may be required as part of the service. The current action rule may also require adjustments for better support. When the initiator of a transaction is inside an organizational boundary, the addition of a pre-decision/management type transaction inside the boundary may reduce exceptions. When the executor of a transaction is inside an organizational boundary, the addition of a supportive type of transaction for the executor may assist in reducing exceptions. Providing additional boundary transactions as a service interface may also reduce exceptions. Improved actor rules may assist in reducing exceptions.

4.2.2 Changes in implementation level

Implementation level changes are related to “improving how work is currently performed”. In the implementation level, the problems associated with a transaction that are caused by communication issues, routines or problems are related to the actor who is playing an actor role.

- **Type 2.1: Changes in "how to coordinate".** According to DEMO theory, a business level is supported by the information level through information transformation. Information translation is supported by data-level transformations. Problems often occur when information is not effectively shared among the appropriate stakeholders or when the information that is used in a communication loop is incomplete. Incomplete decision-making information makes the commitment of communication loop invalid and causes exceptions. To ensure information completeness, it is important to clarify who will use what information and at what time in the communication loop. However, the complementation is on the implementation level without any change at the ontological or strategy levels. Effectiveness, efficiency and speed are all factors that may cause ontological problems.
- **Type 2.2: Changes to "who coordinates".** This type describes the DEMO-defined actor roles and the actions for playing an actor role. However, issues may be generated when an actor plays an actor role. An actor role represents the execution unit of a transaction type and the initiation unit of sub-transactions from an ontological level. Furthermore, in practice, actors who play an actor role at an implementation level bring different capabilities, accessible resources, and authorities into the process, suggesting that different actors who play the same actor role may achieve different results even with the same ontological construction. For example, when a salesperson plays the actor role of an order completer, he/she is more likely to decline an order when new products and/or services that are not currently offered by a company are requested; this result occurs because he/she only has the authority to sell existing goods and services. When a manager plays this role (because he/she can also play the production manager role), he/she has the authority to ask the developer to develop new products and/or services to reduce the number of declined transactions. The issue is whether an actor role is completely defined so that no additional responsibilities are requested, which may occur whether established routines for realizing an actor role are appropriate or not. Actors who play a role are always related to an organization's structure. When actors can play appropriate actor roles, the efficiency and effectiveness of information exchange is increased.
- **Type 2.3: Changes in passive resources.** This type analyzes how passive

resources are utilized and what is the effect of changing resources. Sometimes, humans are considered to be passive resources; this occurs when humans are not typically related to creative work and are instead simply acting as "a part of a machine".

- **Type 2.4: Changes in active resources.** This type analyzes why active resources are utilized (actor playing an actor role). Moreover, this type considers how resources are connected and whether the capabilities, knowledge structures, and skill levels affect how he/she plays a role. Humans are active resources, and these social aspects must be considered in coordination and cooperation.
- **Type 2.5: Changes in the other aspects.** When any aspect of a strategy or a business model requires adjustments, there will be a corresponding adjustment in the functional requirements with possible solutions in the construction level, including both ontological and implementation changes. Adjustments in the construction level ontology and implementation will affect the readiness of the organization to support a strategy change. When ontology or implementation changes ultimately lead to some culture or "bar" changes, the author considers it as a high "readiness" rate for change. It is also part of implementation.

4.3 DEMO for Analyzing Business Process Improvements

4.3.1 Case description (Case A)

The analysis of "Company A" is conducted in 2012 from February to April by interviewing CEO, headquarter-staffs and retail shop-stuffs for three times. This case study aims to discover how DEMO assists in business process improvement. What are the advantages, potentials and limitations?

"Company A" is a small retail company that was founded in 2005. The main business of "Company A" is importing baby goods, such as baby carriages, baby carts, and toys from overseas, which are locally sold in Japan. The company sells approximately 50 brands from all over the world. When the company first started, its business was entirely based online (i.e., no physical stores). The headquarters, which are located in the center of Tokyo, have since doubled, and now there is also a warehouse. At first, it was easy to manage the business with just the CEO and a few part-time staffers. When a customer order arrived, the staff checked the inventory. If the ordered

products were available, they asked the customer for the payment. Then, they delivered the ordered goods to the customer with an external delivery service. Inventory was updated weekly.

After two years of growing, their business expanded from an online business to both an online and an offline business; three physical shops were opened. The first store is located just five minutes from the headquarters. Only small items are stored in the shop; most of the goods that they sell are stored in the nearby headquarters. When a customer wants to buy something that is not kept in the store, e.g., a baby cart, the ordered goods can be prepared and delivered from the warehouse (headquarters) to the shop in only a few minutes. If the item is not in the warehouse, the staff can help the customer place an order in the store if necessary. This is not a complex process; therefore, they use a spreadsheet to manage the entire inventory.

However, when the second shop was opened in Yokoham and a third shop opened in Fukuoka in 2011, "Company A" started to see challenges in their business process. Staff in the headquarters responded by controlling inventories in all three shops. However, it was difficult for the staff to maintain a balance between sales and purchases in a timely fashion because they were have troubling sharing information among the three shops.

- *Although the shops reported daily to the headquarters about their sales and inventory balance, it was disorganized because they did not have clear assignments for who would take the responsibility of handling the inventory at headquarters.*
- *It was difficult to manage their inventory because of the extensive amount of information that was involved. The stores contained more than 1,000 types of items. As a result, the staff preferred to communicate via phones to confirm the status of the items rather than completing numerous reports. This process led to inefficiency, imprecise inventory control and redundant communications about payments.*

Because of the communication issues mentioned above, it became difficult for the staff in the headquarters to control the variety and the quality of the purchases. Moreover, *because of inefficiency in their communication method, repeatable issues were always generated in different shops. As a result, additional effort and time were wasted.*

4.3.2 Exceptions and business process improvements in "Company A"

The OCD of Case A is abstracted and presented in Figure 4-3. Five main exceptions in "Company A" were defined and analyzed with the following solution. The transactions that may cause problems are indicated by red rectangles.

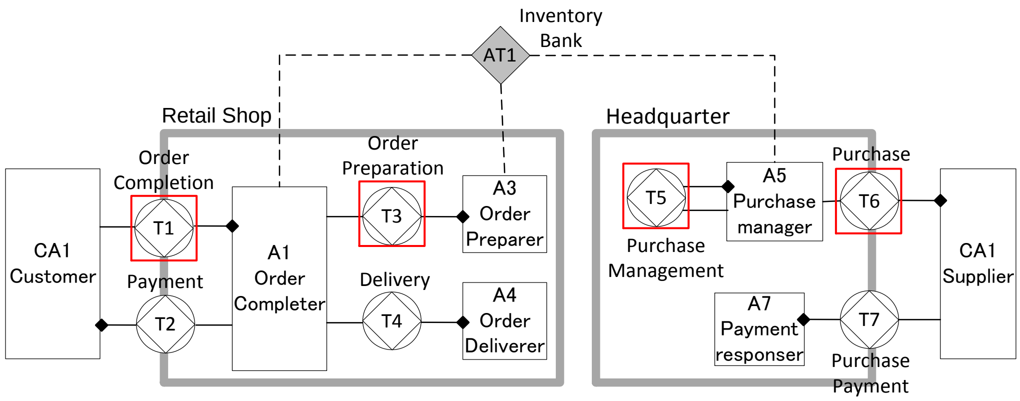


Figure 4-3 OCD as-is of "Company A"

4.3.2.1 Problems and functional requirements

- **Problem 1:** Request of customer is rejected by the order completer because there is no available inventory (request of T3 is rejected).

  Functional requirement 1 (F1): Improve inventory management to avoid unavailability of inventory.

- **Problem 2:** Ordered products are declined by the customer (T1 is rejected) because the quality is unsatisfactory.

  Functional requirement 2 (F2): Improve customer's satisfaction

- **Problem 3:** If the supplier cannot fulfill "Company A's" requirement, the supplier will reject the order (T6 is rejected).

  Functional requirement 3 (F3): Improve purchase management.

- **Problem 4**: The problem in T6 is caused by T5. Incomplete requirement information leads to inappropriate purchases and redundant inventory levels.

  Functional requirement 4 (F4): Accurate inventory management system.

4.3.2.2 Solutions

Solutions at the ontology level (Figure 4-4).

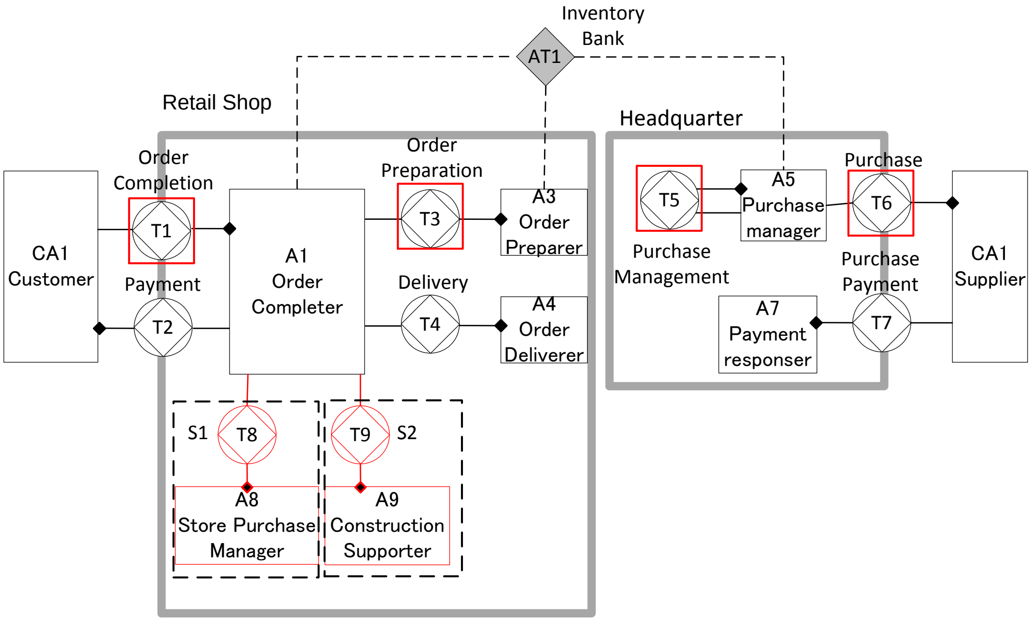


Figure 4-4 ATD to-be of "Company A"

- **Solution 1(S1 for short):** Add a new transaction type to the internal executor of a transaction. To decrease exception E1 in T1, a new management-related transaction type called "store purchase management" T8 was added to the executor of transaction T1 (which is shown in Figure 4-4). As a result, purchases can be separately managed in the retail shops instead of intensively managed at the headquarters. After the change, the actor who plays the actor role A8 or "store inventory manager" in each shop must calculate and report their requirements for their purchase plan to the purchasing manager in the headquarters twice a week. The day after obtaining the reports, responders in the headquarters order the necessary number of items according to requirements that are gathered from the stores. Therefore, the purchasing process becomes more organized and better reflects the actual demand.
- **Solution 2 (S2):** To reduce the number of declined transactions, a supportive transaction, T9 or "construction support" (Figure 4-4), was added for the initiator of the transaction. Before the carts are delivered to the customer, an experienced manager checks the contents of the cart to confirm the quality. Before the improvement, a shop requested additional purchases when a store's inventory was below what was considered the safety inventory. These requests generated mass communications and inefficiency in the purchasing process. Additionally, although a salesperson reported their inventory and sales situation every day to

the headquarters, these reports could not be used efficiently to support purchasing decisions because of too much information.

To conclude, as shown in Table 4-2, there are two types of transactions with corresponding two actor roles added into the construction model, work together with existing actor roles for accomplishing the new functional requirements (F1, F2 and F3).

Table 4-2 Actor-actor role-function mapping table (Case A)

Actors				Actor Role	Functions		
Org1 Customer	Org2 Retail shop	Org3 Headquarter	Org4 Supplier		F1	F2	F3
				A1			
				A2			
				A3			
				A4			
				A5			
				CA2			
				A7			
				A8			
				A9			

Solutions at the implementation level

- **Solution 3 (S3):** This was the main reason that the suppliers could not fulfill an urgent order. To handle this exception, "Company A" improved their information sharing process. Now, they make purchase orders twice a week instead of at random time periods.
- **Solution 4 (S4):** An actor's authority is now verified to improve how work is performed. Two staff members at the headquarters are assigned to take the responsibility of purchase management and the purchasers. One staff member is responsible for overseas brand goods; the other staff member is responsible for domestic brands.
- **Solution 5 (S5):** Ensure complete and efficient information transmission in the communication loop to improve how work is performed. Before improvement, shops sent a free-formatted email to the headquarters to inquire about products. After improvement, all of the requirements are managed by the shop purchase

manager, and he/she sends a standard form document to the headquarters' purchase manager twice a week. Thus, the communication fee is significantly reduced. Furthermore, "Company A" plans to implement a small ERP package to make their communications flow more efficiency.

Functional requirements F1, F2, F3 also need the support of S3 and S4. S5 is for the requirement of F4. In this case, the business process improvements primarily focus on adding new transactions to perform several new things. Meanwhile, their process and organizational structures are justified to better fulfill any new requirements. Moreover, some information transitions were organized and changed. The types of changes in the case are summarized in Table 4-3.

Table 4-3 Change type validation table for "Case A"

Cases	Ontology				Implementation				
	Type 1.1 CM	Type 1.2 PM	Type 1.3 FM	Type 1.4 AM	Type 2.1 How to coordinate	Type 2.2 Who coordinate	Type 2.3 Passive Resource	Type 2.4 Active Resource	Type 2.5 Others
Case A	S1, S2				S3	S4	S5		

4.4 DEMO for Analyzing Enterprise Transformations

Enterprises increasingly have to consider and pursue fundamental change, such as upgrading their current business and implementing innovations such as expending, M&A, and globalization to maintain or gain competitive advantage. Distinct from traditional routine changes, fundamental change is defined as enterprise transformation (Rouse, 2005a, 2005b). Fundamental change is enabled by *"work process change"* to approach value deficiency, which requires "*allocating of attention and resource*" to allow the enterprise to anticipate and adapt to change with resources to yield a further enterprise state (Rouse, 2005a). Enterprise transformation is a more innovative and strategy-related change that influences multiple aspects of an organization, such as routine, organization structure, human capital, and marketing strategy. It is more difficult to model and manage compared with low level changes (Krouwel & Op`t Land, 2012; Op`t Land, 2009).

4.4.1 Case description (Case B)

The analysis of "Company B" was conducted from June to September 2012 by interviewing the project manager and the primary sales person in the first division group, the director of the second division group, and technical managers. This case study aims to discover how DEMO assists higher-level changes such as enterprise transformation. What are the advantages, potentials and limitations?

"Company B" is a Japanese IT company that was founded in 1969 as a software provider. After a long period of growth, several well-known application packages were designed and developed in the early 1980s. One of them, an accounting system for a local government, was recognized as an "outstanding information system." This system is now a primary business of the "first solution division."

"Company B" subsequently expanded its business to include system integration in the late 1980s. For this business, the primary company helps customers analyze their requirements and provides solution plans. As a sub-constructor, "Company B" dispatches work to users' firms to finish assigned tasks according to the upper-level primary constructors' solutions. This is the primary business of the "second solution division".

However, with the evolution of new technology (cloud computing, etc.) and keen competition in the IT area, "Company B" had sought other business opportunities to increase its competitiveness. The transformation inside the company includes the following two business logic changes for different solution divisions:

- From a **passive type** of business **(**"waiting for", and "according to"**)** to an **active type** of business **(**"looking for" and "plan to"**)** in the second solution division;
- From a **software package provider** to an **application service provider** (software as a service type of business) in the first solution division.

Two transformation cases are described in the following sections. The description for each case is based on DEMO IAM. Each case was conducted from three changing aspects: (1) Construction change; (2) Knowledge structure change and (3) Mindset/culture change.

4.4.2 From passive to active

This was the transformation that occurred in "the second solution division" in company B. As a system integrator before transformation, "company B" mainly played the role of a sub-constructor. Typically, the primary constructor has the "know how" to provide solutions that fulfill a customer's requirements. The sub-constructor simply dispatches skilled workers, who have different unit prices according to different skill levels, to the customer's side for software development according to the primary constructor's requirements. In this type of business, a sub-constructor does not need to have much knowledge about a customer's business. In addition, it is not necessary for the sub-constructor to manage the schedule and risk for the entire project. The primary constructor takes responsibility for the schedule and risk instead. The revenue is calculated as the product of the unit price and work time.

"Company B" has proceeded with this type of business for approximately 10 years. One of its largest customers as a sub-constructor is "X", one of the largest Japanese IT infrastructure and systems integrators. "X" has accumulated not only substantial experience in "software development" but also "know how" in different business sectors, particularly billing systems. These advantages drive their transformation from a "sub-constructor" to a "primary constructor".

As a primary constructor, "Company B" provides another type of contract choice: (Request for Proposal). In the new model, "Company B" is responsible for delivering a final solution with controlled time, quality and fixed cost. The complete solution includes requirement analysis, IT system design, software development, and testing. The first successful case of this new type of business was project "Y" in 2009. "Y" is a traditional distributor in Japan that planned to employ a new service product but needed solutions for an effective billing system to support this new business. "Company B" was chosen as the solution provider. This project lasted from October 2010 to October 2011, a duration of one year. By interviewing the project manager and the primary sales person in the second solution division, the conclusion of the transformation was analyzed *from three aspects, including what changed and how it changed.*

4.4.2.1 Solution 1(S1), construction change

What changed:

The as-is construction model of group two of "Company B" is shown in the left side of Figure 4-5. In the as-is model, eight transaction types are defined: transaction type T1, T4, T5 and T10 are out of the boundary of "Company B"; the others are in the scope of "Company B's" business. The to-be model for running the new type of business is shown in the right side of Figure 4-5, where the new transactions are denoted in red and the changed transactions are denoted in yellow. The details of the change are explained in the following section.

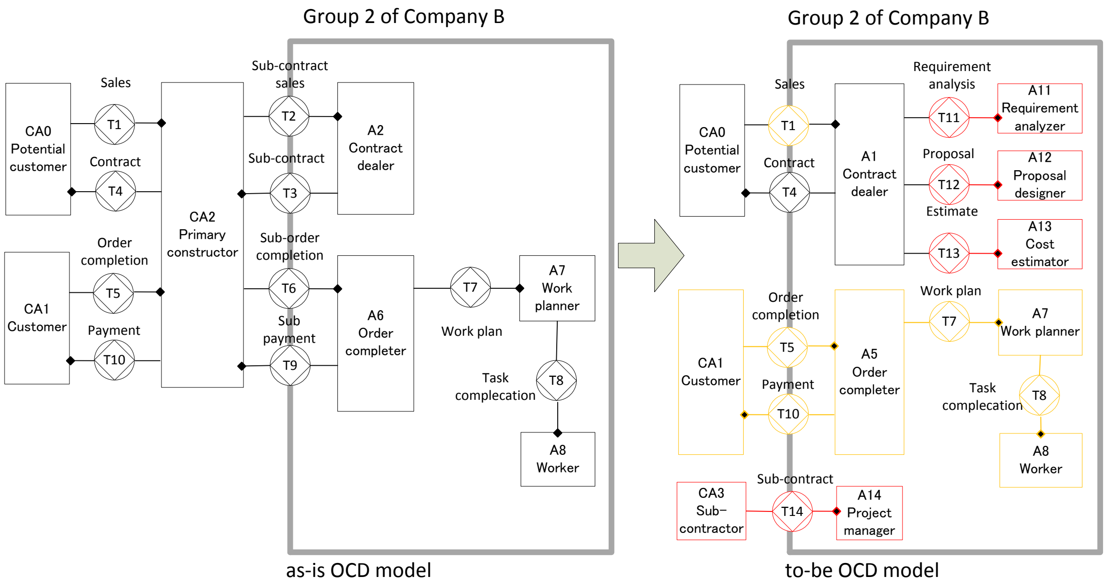


Figure 4-5 OCD of the second group: 'as-is' model and 'to-be' model

As described in Figure 4-5, in the route sales model, "Company B" did not need to propose their solutions. The sales person who played the actor role "A2" as a contract dealer passively sold his own human resources, the available skilled workers who could be dispatched to the customer's side. On basis of a good understanding of the primary constructor's requirements, the sales person provided information about staff and entered into a contract with the customer that compensated staff by the hour. Comparing the as-is model and the to-be model clearly reveals that Ttypes T11, T12, T13 and T14 were not businesses of "Company B" before the change. However, in the new business logic, the old T1, T4, T5 and T10 became part of the business of "Company B" with some changes. A proposal group and not the salesman himself was involved in the sales process. As shown in the to-be model in Figure 4-5, three new transactions (T11, T12, and T13) were added as sub-transactions of transaction T1, which indicates that to finish the new type of sales (pre-sales), “Company B” must analyze customers' requirements, propose a solution to the customers, and prepare a cost estimate. “Company B” can finish the sales process to enter into a contract only when these transactions are complete.

The corresponding actor roles (A11, A12 and A13) for each transaction define new requirements for the capability, responsibility and authority of an actor who plays the actor role. Thus the proposal group includes the following:

- A pre-sales person who can play A11 as a “requirement analyzer”, with the capability to understand customers’ requirements very well;
- A pre-sales person is also required to fulfill A12 as a “cost estimator” with the responsibility of estimating cost and negotiating the price with the customer based on the solution.
- Consulting staff who can play A13 as the “proposal designer”, with the capability of understanding technical details as well as designing and proposing solutions to customers.

How it changed:

For the new type of business, additional communication skills, proposal skills, and “know-how” are required. In particular, a pre-sales person who can play actor roles A11, A12 and A13 is required.

However, “Company B” did not have sufficient capability when they focused on dispatching this business. As a subsidiary company of a large IT company, it was not difficult for “Company B” to invite several experienced sales people with pre-sale skills from the large company to join. With the lead of such experience in pre-sales, the sales people inside the company obtained more knowledge about pre-sales.

The “proposal group” was built in “Company B”, including pre-sales, a manager, and consultants to play the actor roles of A1, A11, A12 and A13 together. The “proposal group” integrates closely with customers to actively understand their business and requirements and then proposes a suitable solution. Cost, risk, and schedule must be carefully estimated and managed. When the proposal is accepted by the customer, the analysis and design process will be in progress until final delivery is confirmed by the customer.

4.4.2.2 Solution 2(S2), project management method change

What changed:

In the dispatched business model, the primary constructor or the customer managed the project to control cost, quality and schedule.

However, in the new business model, all responsibilities are managed by "Company B", which must therefore execute cost control, project schedule and quality control.

In the to-be model of Figure 4-6, although the construction of transactions T1, T5, T7, T8 and T10 is similar to that of the traditional construction, the action rule for the actor role has changed. For project management, the content of T7, the "work plan", changed with the expanded plan requirement, such as cost control and quality control.

As a primary constructor, "Company B" also dispatches work to sub-constructors according to their work plan. The additional transaction T14, "sub-construction", can be initiated by the project manager in "Company B".

How it changed:

It was a big challenge for managers in the second solution division. Their solutions were practical experience and on-the-job training.

4.4.2.3 Solution 3(S3), knowledge structure change

What changed:

Route sales persons did not require compact knowledge about customers' business and technical details of the solution. In addition, technology staff did not need proposal skills. Neither required estimation knowledge about cost and risk control.

However, it is necessary for the "sales group" to have this knowledge to proceed with the proposal type of business as a prime contractor.

How it changed:

As mentioned above, by inviting experienced pre-sales persons to join "Company B", the company obtained the capability of conducting proposal sales. In addition, training courses related to cost estimation, project management and proposal skills were introduced to enhance the employees' knowledge of structure evolution. The most important training was "on-the-job-training" which involved staff in the project to grow their skills quickly and effectively. After the first "Project B" was finished, "Company B" requested that the project members conduct lectures sharing knowledge and experience with all staff inside the company.

4.4.2.4 Solution 4 (S4), mindset/culture transformation

What changed:

The "request for proposal" type of business forced "Company B" to actively integrate its resources for proposing and fulfilling customers' requirements rather than passively "selling" their existing resources. Correspondingly, a culture transformation from passive mode ("waiting for", and "according to") to active mode ("looking for" and "plan to") was required.

However, it was difficult to progress with the new type of business without a positive working mindset and competitive spirit. Employees typically do not want to change their current work style, particularly given the high risk of this first project, which was ultimately successful.

How it changed:

"Company B" successfully encouraged this mindset transformation in three ways:

- Top-down encouragement: because of the decrease in dispatched business, the boards of directors decided to transform from a dispatched type of business to a proposal type of business. After serious research, the CEO gave several lectures on the necessity and benefits of introducing the new type of business, beginning in 2009. This approach encouraged a change in the employee mindset by making the entire company realize that they had to proceed with this change.
- Environmental motivation: as introduced, "Company B" has 40 years of IT

experience. Middle-aged (40s) staff were not as positive about learning new concepts and undertaking this challenge. This attitude was a barrier to transformation. At the beginning stage, "Company B" chose those employees who were eager to accept the challenge of being involved in the new type of project. They also planned substantial "on-the-job training" to assist these employees in their growth. The change in these employees and the success of the project increased confidence in conducting this type of project, increasing motivation. In addition, their knowledge-sharing mechanisms assisted the mindset transformation. A spirit system was also introduced to encourage the culture change from passive mode to active mode. The interviewers confirmed that most of the company employees became more positive to the proposal type of business after this project.

- Standardization of new business logic. By successfully completing the first proposal type project, "Company B" standardized the proposal process and built a "Primary Center" to open up new business opportunities. It was a development center composed of pre-sales persons and consultants to provide entire solutions. This center not only develops their own business based on their "know-how" but also integrates with the other company's products to propose the best solution to the customer.

By the time of the interview, approximately 10% of revenue was derived from the proposal type of business in the second solution division.

4.4.3 From product to service

This is the transformation in the first solution division of "Company B" that focused on "package business" before the transformation.

Local government support is a traditional business in the first solution division of "Company B". As a package vendor, their position is defined as serving small and medium cities and towns in Japan, which distinguished "Company B" from other e-government production vendors. "Company B" will not be competitive unless it can transform from being a package vendor to a service provider.

"Company B" made its first step by successfully providing an application service to regional government "Z". The construction model of the 'as-is' model is presented in the left side of Figure 4-6, and the 'to-be' model is presented in the right side of Figure 4-6. In the to-be model, the new transactions are denoted in red, and the changed transactions are denoted in yellow. Primary challenges, transformations and solutions are discussed in the following sections.

4.4.3.1 Solution 5 (S5), construction change

The original construction model of the first group in "Company B" is given in the left side of Figure 4-6.

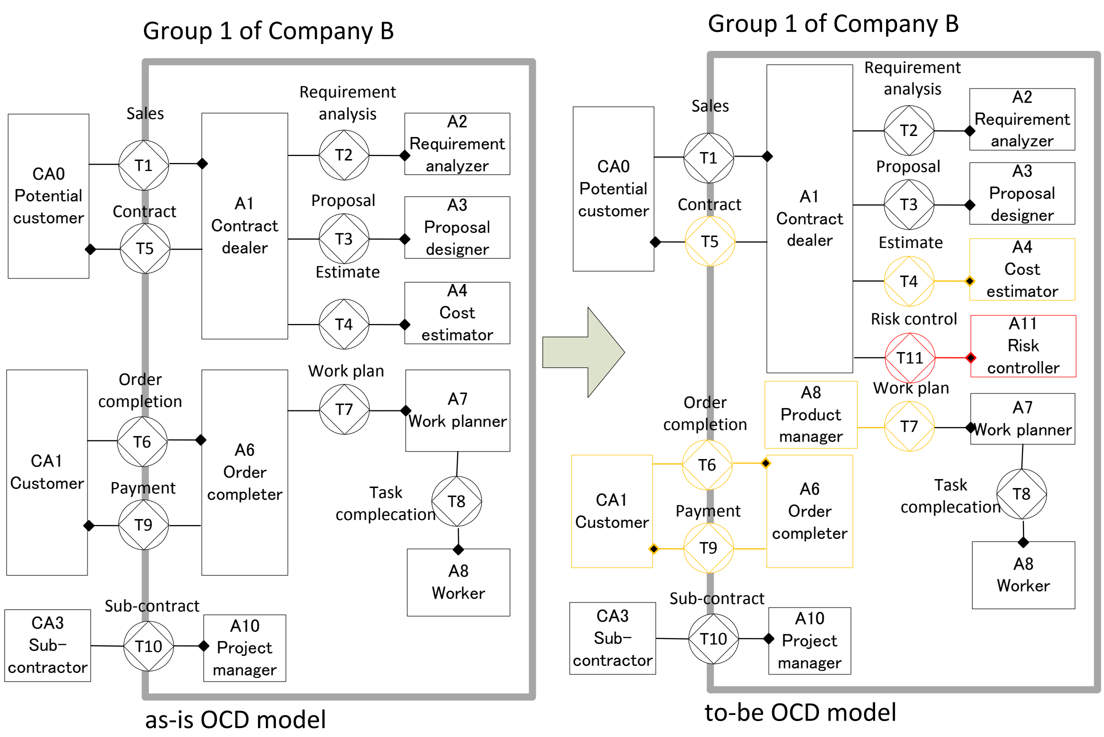


Figure 4-6 OCD of the first group: 'as-is' model and 'to-be' model

To provide a service instead of selling products, risk control must first be added. In the new type of business logic, sales people must estimate the costs and risks of providing the service before making a proposal. As a service provider, "Company B" must consider the risks of service and control those risks. A new transaction T11 is therefore added in the 'to-be' model in Figure 4-6.

For transaction T7, the product manager and not the order completer initiates T7 to prepare the "work plan" before the order is received. As a service provider, T6 and T9 remain the same, but the action rule changes. For T6, "Company B" supplies a service to a customer instead of producing to order. Accordingly, the rule of T9 "payment" changes as well.

4.4.3.2 Solution 6 (S6), knowledge structure change

What changed:

The knowledge for cost estimation, risk control and service quality control, such as the service reaction time, recovery time, and error rate, was all new but required additional knowledge to achieve the transformation from product to service.

How it changed:

The first solution division obtained the required capability through the learning process. They asked other experienced consultants to gain such knowledge. In addition, they learned by practice.

4.4.3.3 Solution 7 (S7), mindset/culture change

What changed:

The transformation from products to service required a corresponding mindset transformation from products to service (Bettis & Prahalad, 1995; Prahalad & Bettis, 1986; Vargo & Lusch, 2008; Vargo, Maglio, & Akaka, 2008).

How it changed:

The mindset change was traced by the evolution of "virtual machining computer technology" and environmental motivation.

4.4.4 Assumptions and analysis

In "Company B's" case, the author concluded that the enterprise transformation included five stages, as shown in Figure 4-7 and explained as follows:

4.4.4.1 S0 (TDL): traditional dominant logic stage

S0 is the beginning stage. In this stage, the enterprise conducts the traditional business with the traditional dominant logic.

4.4.4.2 S1 (P): preparation stage

In this stage, although the organization retains the traditional dominant logic, it is preparing for change because of "value deficiencies", environmental forces, existing defects, exceptions or other factors.

Individuals achieve corresponding knowledge as well as skills, trust and passion, tension, etc. These factors are inside individuals in the organization and are known as tacit knowledge. In S1, tacit knowledge is accumulated by acquiring experience in the new type of business.

4.4.4.3 S2 (TF1): transforming stage 1.

Transforming stage1 is the stage in which new business logic is added to the current business structure, although the new logic participates only to a very small extent without becoming the dominant logic.

In this stage, the organization has had experience with the new type of business. Some documents, routines and methodologies have been accumulated as organizational knowledge.

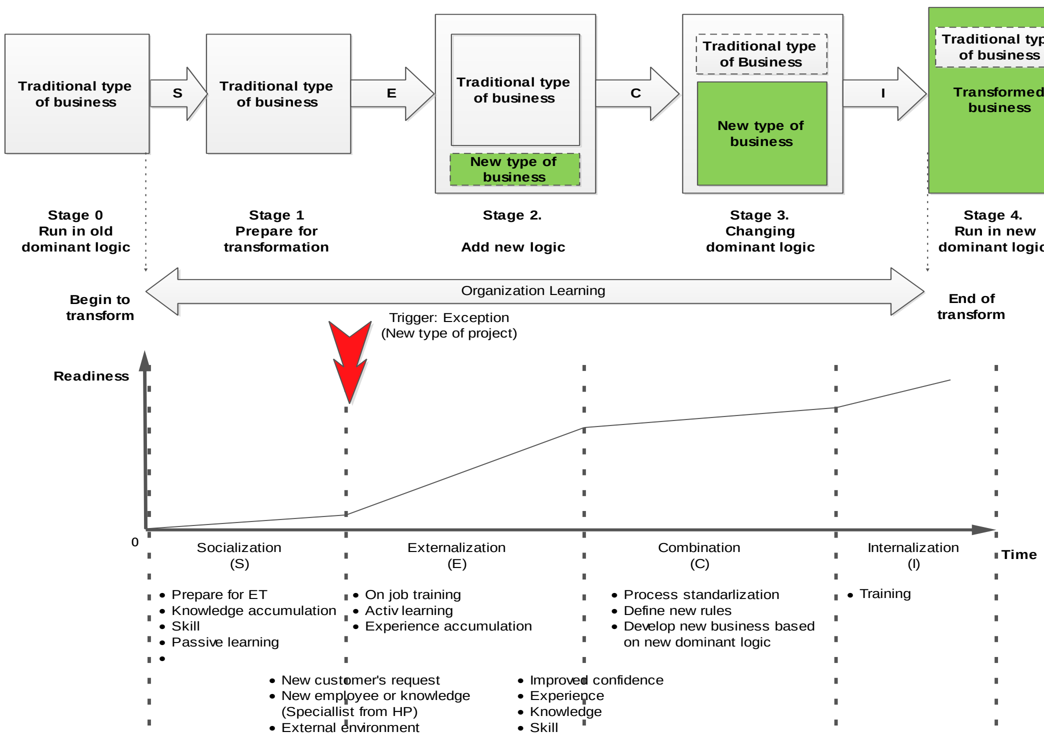


Figure 4-7 Five stages of enterprise transformation

4.4.4.4 S3 (TF2): transforming stage 2.

In transforming stage 2, the new business is growing. It represents an increasing percentage of the whole business structure so that the organization begins to change its dominant logic to include the new type of business.

In this stage, the organization has acquired more experience in the new type of business. New knowledge has been accumulated and combined with existing knowledge. The related documents, routines and methodologies are standardized as the guide for the new business logic.

4.4.4.5 S4 (NDL): new dominant logic stage

In the new dominant logic, the changed dominant logic targets the best performance. The traditional business and the new business are balanced, and the transformation is complete.

4.4.5 Transitions in the transformation process

As Figure 4-8 shows, the transformation from Stage 0 to Stage 4 is an organization in the learning and knowledge creation process. This transformation can be discussed and mapped using the SECI model (Nonaka & Takeuchi, 1995) to specify how organization learning supports transformation. Transformation includes four transitions: Socialization, Externalization, Combination, and Internalization.

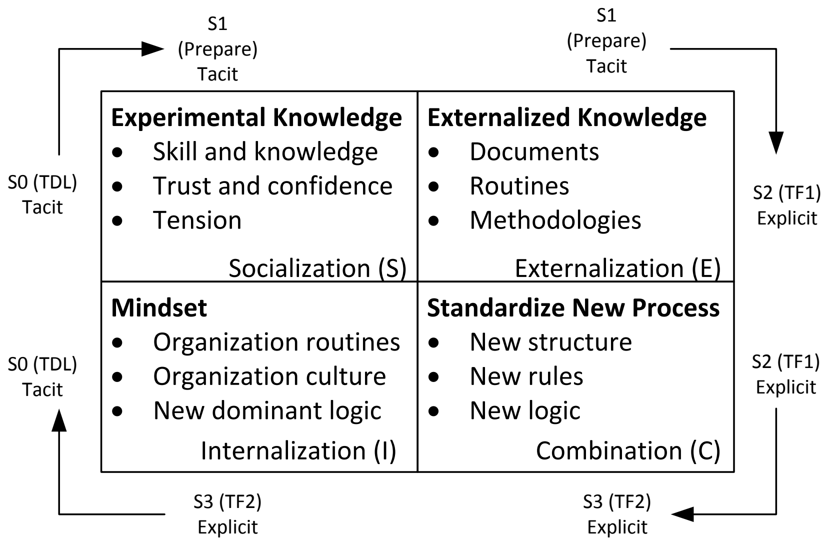


Figure 4-8 SECI for enterprise transformation

4.4.5.1 Transition S: From S0 to S1 (Socialization)

This is a transition from the traditional dominant logic stage to the stage of preparation for change. Tacit knowledge, such as knowledge, skills, confidences, trusts and so on, is accumulated. Individuals obtain such tacit knowledge via the learning process, which may include training or experiences.

In “Company B”, the staff accumulated tacit knowledge of the billing system through considerable working experience with “Company B”. This is the tacit knowledge “Company B” had before considering the transformation.

Meanwhile, when “Company B” began to consider the transformation, it hired experienced pre-sales persons who had tacit knowledge of the new type of business. Thus, tacit knowledge was obtained.

4.4.5.2 Transition E: From S1 to S2 (Externalization)

This is a process of transition from preparing for change to the stage of conducting a new type of business. In this stage, tacit knowledge is externalized as explicit knowledge.

The externalization is realized by conceptualizing tacit knowledge through multiple methods. Symbols, figures, and other methods are utilized to express the tacit knowledge of individuals. This explicit knowledge for enterprise transformation could be new process documents, routines, and problem-solving methodologies.

As a knowledge representation methodology, engineering methodology assists the externalization of the tacit knowledge of the enterprise structure. The DEMO construction model can be used to compare the traditional business structure and the new business structure. As shown in Figure 4-8, T11, T12, T13 and T14 are newly defined transactions with corresponding actor roles. The actor roles in DEMO define “authority,” “responsibility,” and “capability”. Thus, the accession of new actor roles requires operant resources with the corresponding capabilities and knowledge.

Handling exceptions can also trigger the externalization process. In “Company B”, the externalization of tacit knowledge occurred through its exception handling process. “Company X” and “Local government Z” were both “exception” cases that differed from “Company B’s” traditional business. The exception handling processes in “Company B” were not single-loop learning (Argyris & Schon, 1987), which focuses on routine change, but double-loop learning (Argyris & Schon, 1987), with the development of more complex rules and associations regarding the new business.

This learning process demands correspondingly upgraded organizational knowledge structure. For “Company B’s” knowledge structure in an ‘as-is’ model, “Company B” did not have such capability to support the ‘to-be’ business. One of “Company B’s” solutions was to hire experienced pre-sales persons that had related tacit knowledge. In the learning process, this tacit knowledge was externalized as procedures, project documents, and templates and so on.

For the staff involved in such a project, the experience of acting as a primary constructor is a socialization process that also increases their tacit knowledge.

4.4.5.3 Transition C: From S2 to S3 (Combination)

This is the process of expanding the new type of business.

In this process, newly achieved explicit knowledge is combined with existing or other newly achieved explicit knowledge to improve the current system.

In Company B’s case, after the success of delivering their first “active type” of business, they performed a series of systemizing actions. The “primary center” was created to continue this type of business, with newly developed solutions for other industries besides the billing system. The “project’s” new business and externalized knowledge was taken as a new standard, similar to the “software as a service” type of business.

4.4.5.4 Transition I: From S3 to S4 (Internalization)

As Figure 4-8 shows, this transition is a process of internalizing explicit knowledge into individuals. By defining routines and rules and sharing experiences, the new explicit knowledge is accepted by other individuals in the organization and internalized as part of their tacit knowledge.

In “Company B”, in response to the successful experience of two “exception handling” projects and the standardization of the new business, employees increasingly accepted the change and intended to learn about and become involved in similar projects. Explicit knowledge of accomplishing these projects is internalized.

The SECI model is not a sequential type of learning but a spiral type of improvement. The increasing intention, confidence and benefit change the staff’s mindset and the culture of “Company B” to a more active atmosphere for expanding the new business. With more experience and externalization, combined with the internalization loop, “Company B” will be able to complete its transformation.

At the time of the interview, “Company B” was in the beginning of Transforming Stage2. The revenue of the new business represented only 10% of the whole. With the growth of the new type of business, “Company B” planned to transform the dominant business logic of its second division to the “active type of business” in the near future, the same as in the first solution division. The SECI model and learning process were continuing in “Company B”.

4.4.6 Conclusion

As a knowledge-intensive company, the transformation of “Company B” was strongly related to its organizational learning and knowledge creation process. When the company became aware that the change requires a double-loop learning process, they decided to change not only at the routine level but also at the rule, dominant business and knowledge structure levels.

By coping with the organizational learning process, externalized relative knowledge, as an output of the E (Externalization) process or C (Combination) process, is accumulated in the organizational memory (Levitt & March, 1988). The content of organizational memory can be internalized as tacit knowledge, thereby affecting the organizational culture and mind set.

In “Company B”, the transformation has not yet had a significant effect on KPIs (Key Performance Indicators) or other performance indicators. However, by analyzing the SECI process, an improved and dynamic change in “readiness” is apparent, as shown in Figure 4-8. This indicator describes how ready a company is for change from the knowledge, resources and procedures perspectives. When the readiness for transformation is reached through this continuous accumulation process, the enterprise can easily be transformed into the new business.

The readiness for transformation increases during the learning process. It is also affected by organizational structure change and culture change.

This research benefits from the engineering viewpoint in these two aspects:

- By analyzing the DEMO CM model, structure characteristics under business change can be simplified and analyzed.
- The actor roles defined in DEMO represent capability, responsibility, and authority. When the structure of responsibilities and required capabilities can be pre-defined, innovation methods or transformation can be easily analyzed and produced.

The types of changes in "Company B" are presented in Table 4-4.

Table 4-4 Change type validation table for "Case B"

Cases	Ontology				Implementation				
	Type 1.1 CM	Type 1.2 PM	Type 1.3 FM	Type 1.4 AM	Type 2.1 How to coordinate	Type 2.2 Who coordinate	Type 2.3 Passive Resource	Type 2.4 Active Resource	Type 2.5 Others
Case B	S1,S5	S5		S1, S2, S5		S2		S3,S6	S4, S7

4.5 Discussion

Business process improvements are more concerned about business process changes and organizational structure changes, while enterprise transformations are more related to changing the knowledge structures, culture and organizational structures. Both of these changes are required to consider not only a high level of construction, such as ontology, but also the details of construction, such as implementation.

DEMO provides a new perspective for analyzing enterprise changes that consider the "why" and "what is required" questions and part of the question "how" question. However, the weakness of this approach is that DEMO is not well integrated with implementation, which imitates the usage. To solve this problem, a framework following the concepts of enterprise engineering is defined, in which the enterprise is analyzed by function and construction. The author focused on construction changes; these changes were classified into six types and two levels: the ontological level and the implementation level. This framework was validated using two real-world case studies. The first case (Case A) is for business process improvements, while the second case (Case B) is for enterprise transformations.

In this work, the first research question, regarding analyzing business process changes, is addressed, i.e., "how do we analyze business process changes in the context of enterprise engineering?" By illustrating these two cases, we show how to analyze enterprise changes as functional requirement changes and how the functional requirement changes can be mapped into constructional changes, including both ontological-level changes and implementation-level changes.

However, both methods of applying DEMO to a real-world case analysis are qualitative research studies. Qualitative analyses are a valid method; however, they are not sufficiently robust to support business process changes. Such methods are good at explaining and comparing options; however, they provide limited evidence to answer the "why" question. Both case studies do not consider resources because the qualitative method does not provide evaluation assistance, which was discussed in chapter 2. We require a more powerful framework that is measurable and predicable to analyze enterprise changes, which are simulation based. The method is introduced in chapter 5.

CHAPTER 5 Business Process Simulation in the Context of Enterprise Engineering

5.1 Introduction

Business process models are always used as multi-purpose tools for understanding the operations of existing organizations and to assist in business process redesign and reengineering. However, redesign and reengineering always involve changes in people, processes and technology over time. Sometimes, the interactions of people with processes and the possible outcomes of these changes must be evaluated and compared. However, modeling alone may not provide sufficient information to achieve the objective. This is where simulation can provide values. Several studies (Gladwin & Tumay, 1994; Paul & Seranno, 2003; Seila, 2005) have been conducted in the business process simulation field. Other researchers (Barjis & Dietz, 1998; Barjis, 2007b, 2008) have also suggested that business process modeling and business process simulation should be combined. On one hand, a business process model should be complemented by simulation to achieve significant benefits and accurate results. On the other hand, simulation may provide little assistance without substantial prior conceptual modeling.

However, as discussed in section 2.2.2.4 (i.e., "limitations of BPS research"), there are several barriers that prevent BPS from being widely used in support of business process change analyses.

- Limitation 1: Current simulation methods are weak at describing large, complex systems.
- Limitation 2: Introducing and using the method in management is challenging.
- Limitation 3: There is a high level of complexity associated with changing models to simulate new designs in BPR.
- Limitation 4: There is weakness in confirming consistency.

To overcome these barriers, the author took a new perspective that analyzes business process changes in the context of enterprise engineering. In this context, an enterprise is a white box that consists of construction models that built to realize functional requirements. There are two levels: a high-level abstracted enterprise ontology and a detailed-level implementation of the ontological model. Then, the research must address the second question, i.e., "how can a simulation be conducted in the context of enterprise engineering to assist business process change analyses?"

5.2 Research Questions and Research Design

Zeigler et al. (2000) suggested a framework for discrete event modeling and simulation. A *conceptual model* is built through a collection of assumptions about the system components and the interactions among them, which involves some degree of abstraction about the system operations. An *operational model* is an executable model that implements a conceptual model. Our research is designed to follow Zeigler's conceptual and operational model structures. There are three levels, which are described on the left-hand side of Figure 5-1: an ontology model, a conceptual model and an operation model.

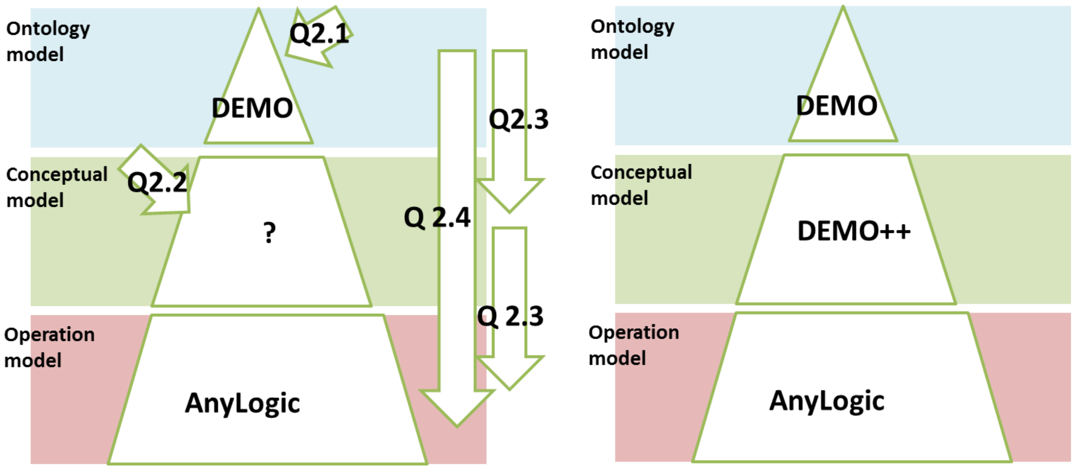


Figure 5-1 Framework for BPS

To conduct a simulation in the context of enterprise engineering to assist in business process change analyses, the following sub-research questions must be solved.

- **Q2.1**: Is DEMO adequate for the specific simulation?
- **Q2.2**: If not, what type of conceptual model should be defined for BPS?
- **Q2.3**: How can DEMO be translated into a new model and ultimately into an executable simulation model?
- **Q2.4:** Is it possible to carry out this process automatically or semi-automatically?

How can this process be accomplished?

Thus, this study was designed as shown on the right-hand side of Figure 5-1.

- At the ontological level, DEMO is used to describe the real world at a high level of abstraction. DEMO has been proven to support discrete event systems in several studies (Barjis, 2008; Barjis, 2007a; Barjis & Dietz, 2000; Barjis et al., 2001; Barjis et al., 2009). However, DEMO is not sufficient for specifying a simulation. Different implementations for the same ontology that achieve the same functional requirement are possible. In some cases, business process changes with related construction changes can be considered to implement the same enterprise ontology in a different manner. In other cases, both must be changed. However, ontology is not sufficient for simulation. This research uses simulation to understand implementation problems, such as optimization or an as-is/to-be analysis, which are closely related to how to arrange the ontology to occur because ontology can have different implementations in the same strategic instances.
- To take advantage of enterprise ontology, DEMO++ is proposed, which is an expanded version of DEMO with an implementation model that represents a conceptual modeling method for business process simulation. There are two parts of the framework: ontology and implementation. The former describes how the enterprise is constructed, while the latter describes how the notable portion of an ontological model is implied. DEMO++ is a method for analyzing, executing and evaluating business processes that reduces the complexity associated with modeling and simulating large systems and that focuses on implementing the interesting part. DEMO++ is fully modularized such that the generated simulation is entirely based on individual components and includes various features, such as controllable changes and reusability. Moreover, DEMO++ supports business process simulation, especially when the simulation is used in top-down enterprise reengineering.
- At the operation level, AnyLogic (XJ Technologies, 2009) is used as the execution environment to test DEMO++. There are different types of simulation environments. AnyLogic is utilized for its excellent user interface and hybrid simulation support capability. For the purpose of enterprise reengineering and

enterprise transformation, we must consider not only business activities from a workflow perspective but also actor interactions from an agent perspective to obtain a full understanding of how a system works and the potential effects of any changes. The effectiveness and efficiency of an entire system from a system dynamics perspective may also be necessary. AnyLogic is the only simulation software that has powerful support for hybrid simulation and can combine discrete events, an agent base, system dynamics and other methods.

Based on previous research, a meta-model mapping-based model transformation was conducted to translate DEMO into an executable simulation model. The outcomes include three parts: (1) the meta-models, i.e., DEMO CM, DEMO PM, DEMO AM, DEMO FM and DEMO++, (2) the AnyLogic DEMO++ library and (3) the transformation rules for the meta-models (T1, T2, T3 and T4).

- T1: From DEMO CM to DEMO PM.
- T2: From DEMO PM to DEMO AM.
- T3: From the DEMO aspect models to the DEMO++ model.
- T4: From the DEMO++ model to the AnyLogic executable model.

The entire structure is introduced in this chapter; it is validated using the Buono Pizza case study (Case C), which was mentioned in section 3.1.

5.3 DEMO

The related meta-models for each aspect model were introduced in chapter 3. The entire meta-model is summarized in Figure 5-2. The yellow part describes the CM, the pink part describes the FM, the green part describes the PM, and the orange part describes the AM.

5.4 DEMO++

5.4.1 Overview of DEMO++

DEMO++ is an ontology-based conceptual model for simulation. Following Zeigler's framework (2000), DEMO++ is designed as a hierarchical structure of components. The meta-model of DEMO++ is shown in Figure 5-3. There are three parts in the DEMO++ framework: ontology, implementation and main.

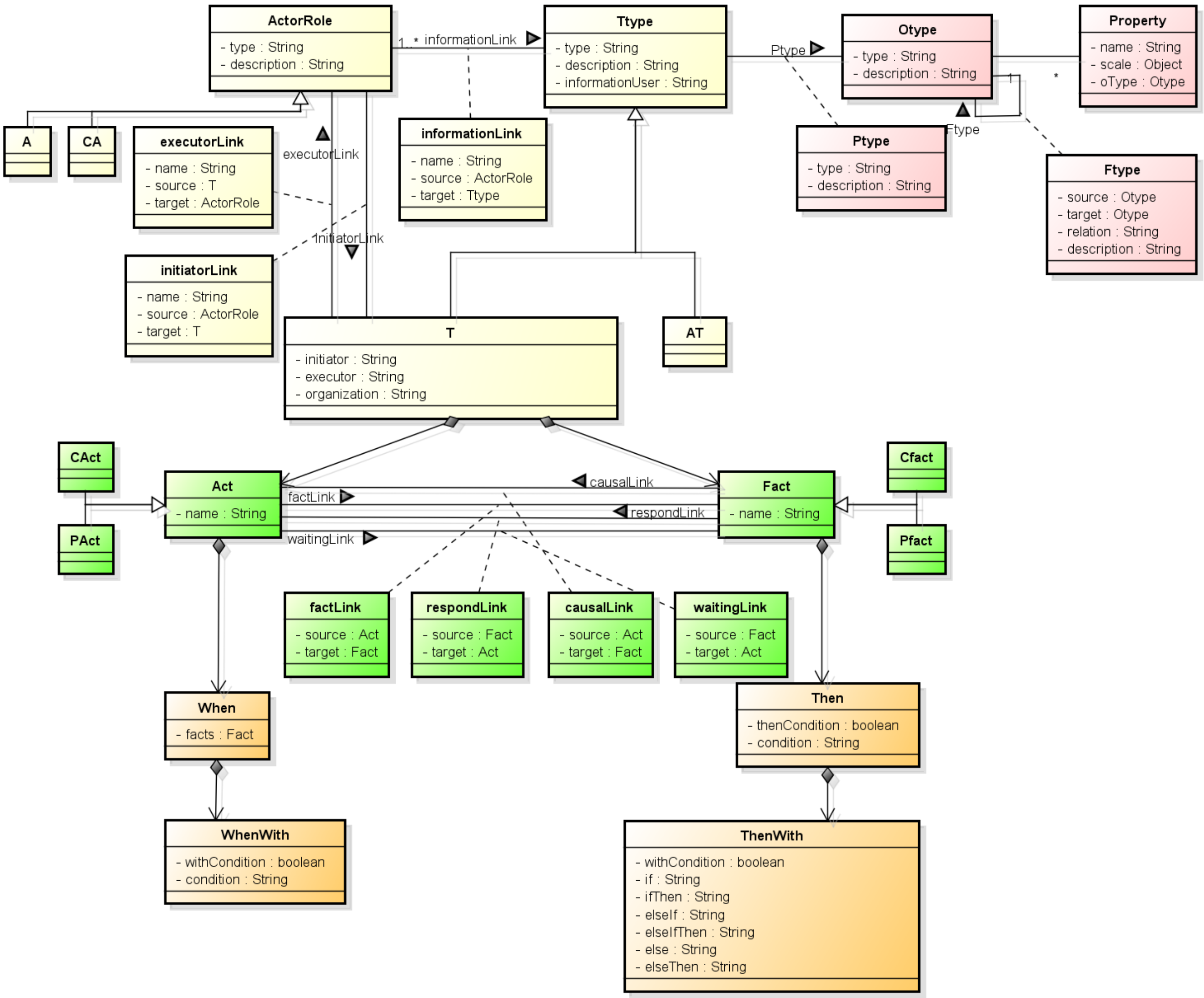


Figure 5-2 Meta-model of DEMO

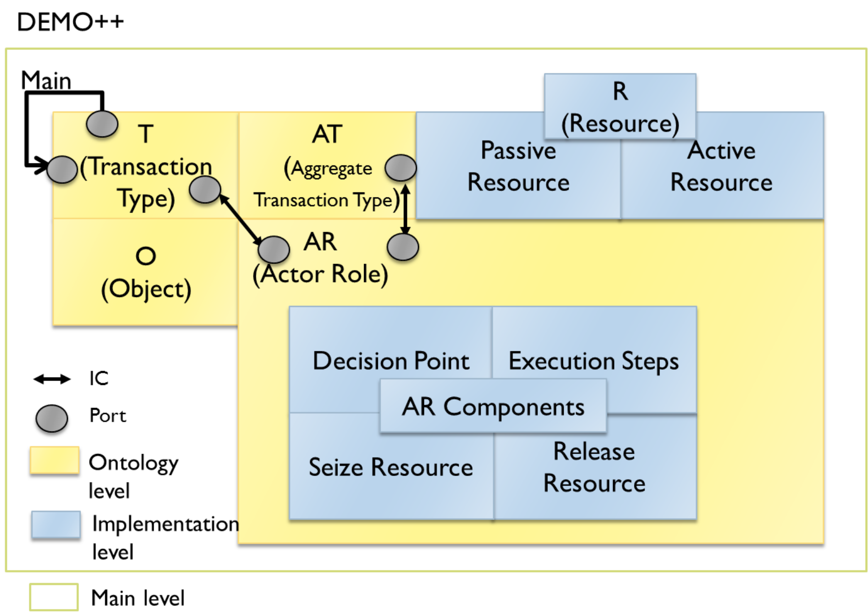


Figure 5-3 DEMO++

- *Ontology* (the yellow portion of Figure 5-3) describes the ontology of an enterprise. It describes the components of the transaction types (T) and aggregate transaction types (AT), such as information banks, object types (O), and actor roles (AR). These components are connected through ports. The internal connections (IC) between them include the following: (1) causal or waiting relationships between transaction types (T), (2) relationships between AR (initiators or executors) and transaction types (T), and (3) information access between actor roles (AR) and information banks (AT).
- *Implementation* (the blue portion of Figure 5-3) defines the implementation of the ontology by considering how the execution steps of the ontological acts are defined, how they are related to resources, and what the delays due to a lack of cooperation are. The implementation components of the resource type (R) and actor role component (AR component) are defined.
- Both ontology and implementation are contained and connected in the *Main* component (the boundary rectangle shown in Figure 5-3). Each component contains ports, including an *input port* through which the component can obtain events or information and an *output port* through which the component can send events or information to others. The *Inter-Connection* (IC) is a connection between the output and input ports through which the components are connected so that they can communicate. The *External Input Connection* (EIC) is a connection between the input ports of hierarchical components through which the obtained event or information can be transferred from a component to the subcomponents. The *External Output Connection* (EOC) is a connection between the output ports of hierarchical components through which generated events or information can be sent out.

5.4.2 Ontological model

The meta-model of the ontological model of DEMO++ is defined as shown in Figure 5-4, including O, T, AT and AR. Each component will be explained in detail in the following sections.

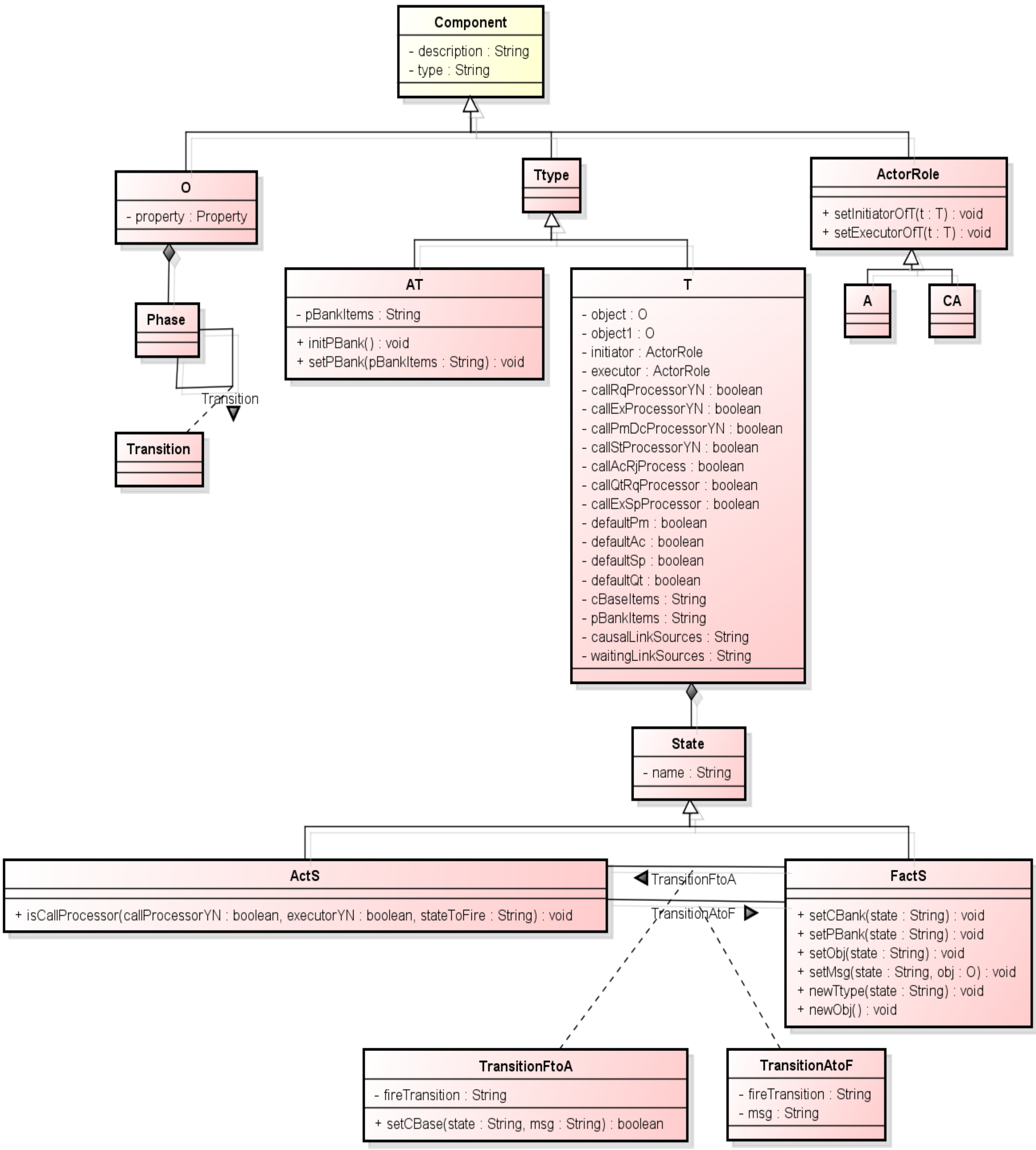


Figure 5-4 Meta-model of the DEMO++ ontological model

5.4.2.1 O (Object type)

O is a component of object type whose state will be changed by transactions. It follows the concept of Object type in DEMO with the definitions shown in Figure 5-4 (O). For Buono, three objects are defined (cf. Figure 3-8). The object “Order” is described in Figure 5-5 for illustration.

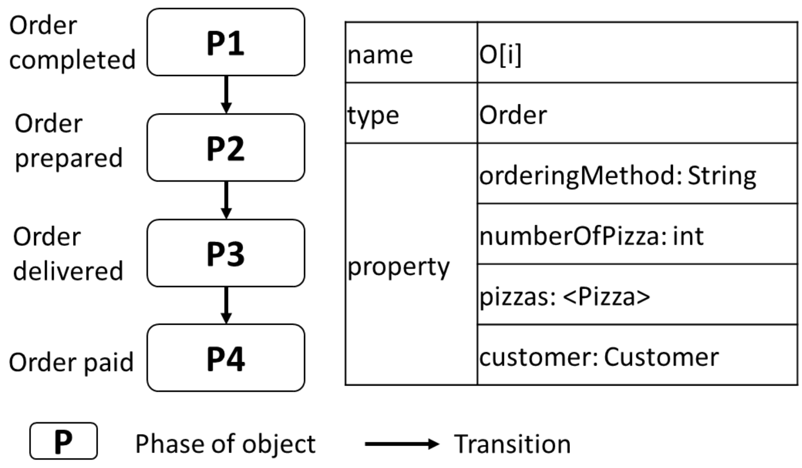


Figure 5-5 Object type component "Order"

- *name* is the full name of the object type (e.g., "Order");
- *type* indicates the short form of the type name (e.g., "O");
- *properties* are obtained from the properties or related fact types defined in the OFD.
- *phases* and *transitions* are defined on the left side of Figure 5-4, a state transition diagram. *Phases* are an object's possible stages. Derived from the OPLink in OFD, there are four phases for the object type "Order": P1, P2, P3 and P4. These phases indicate that an order must be prepared, delivered, and paid for before it can be completed. Transition is the phase change.

5.4.2.2 T (Transaction type)

T is a component of transaction type that is derived from the transaction type (cf. section 3.2.1, Figure 3-3) in DEMO. According to the meta-model (T in Figure 5-4), the following must be defined for each transaction type, as shown in Figure 5-6 (table Transaction):

- *type* is the transaction type (e.g., T1);
- *name* is the description of the transaction type (e.g., "Order completion" for T1);
- *initiator* and *executor* follow the DEMO definitions;
- *object* is the related objects

A transaction type is not only a set of acts but also a collection of generated facts in the information bank. Information banks must be defined for each transaction, including cBank and pBank:

- *cBank* contains all of the Cfacts generated by the transaction type. Each bank item is a type of fact, defined as a tuple <type, status, time> (also called a factum), indicating whether an instance of this Cfact type has been settled (*status* is true) or not and the settlement time *t*. For example, <rqedT1, true, t_1> means that Cfact instance "rqedT1" was settled at time t_1.
- *pBank* contains all of the Pfacts generated by the transaction type. pBank items are formatted in the same manner as cBank items.

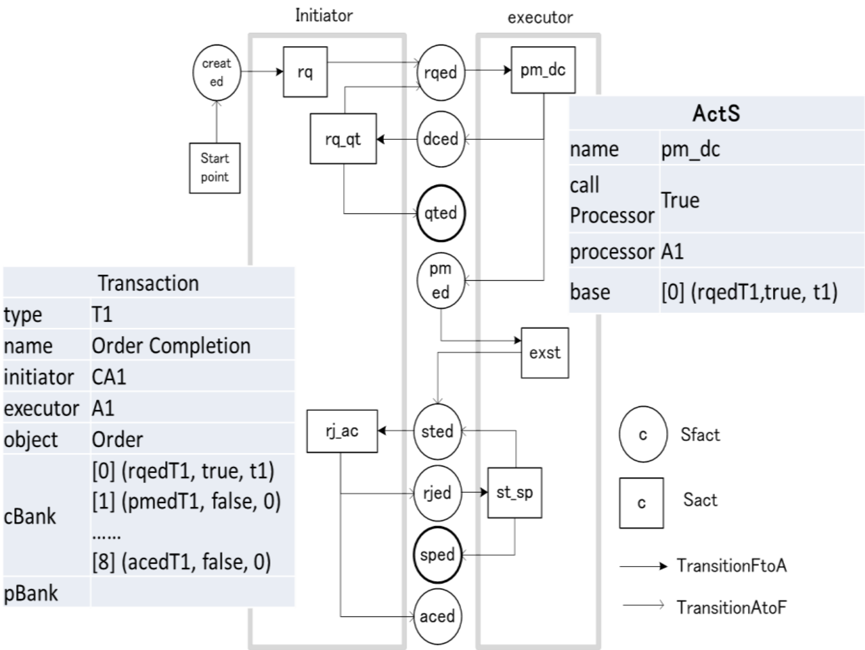


Figure 5-6 Transaction type component "T1"

Following DEMO's standard transaction pattern, a state transition pattern is defined for each T, as shown in Figure 5-6. There are two types of states, and two types of transitions between states are defined:

- *ActS*: indicating the state of acting, represented as a rectangle. In DEMO, acts are abstracted ontological concepts. To combine ontology with implementation, each ActS in DEMO++ is related to a processor. As presented in Figure 5-6, the *processor* is the corresponding actor role who takes responsibility for making decisions and implying the ontological act. Ontology can either be expanded into implementation or not, depending on whether resource utilization, communication flows or other operation-related details must be examined. The shift is controlled by the Boolean property *callProcessor*: "true" indicates that the ontological act will be processed with a feedback of generated results after some delay. According to the DEMO AM, an automaton can proceed only when all of the state conditions are satisfied. *Base* contains the statuses of all Cfacts and Pfacts necessary to perform an act. An act can be performed only when all items in its "base" are satisfied. Base items are formatted as facta. For example, the required state condition for act pmT1 is "T1 has been requested". This condition is represented as base item <rqedT1, true, t1> of ActS "pm_dc". By using the base, the multiple state conditions required to perform an act can be verified.

- *FactS* indicates the state after an act is performed, called the fact state and represented as a circle. For FactS, we must update the effects of an act with a number of functions, including updating banks of the transaction type, updating the states of related objects, and adding new transaction instances if necessary.
- *TransitionFtoA* is a transition from FactS to ActS. It is triggered when all of the items in the act's base are true.
- *TransitionAtoF* is a transition from ActS to FactS. It is triggered by a particular message.

All transaction types are modeled following a state transition pattern, with different names, types, initiators, executors, objects, banks, and bases for ActS and reactions for FactS.

Several types of ports are defined for transaction types corresponding to different types of links in DEMO:

- Port type i: input and output port, connected to the initiator;
- Port type e: input and output port, connected to the executor;
- Port type w: input port, connected to other transaction types (derived from the waiting link)
- Port type o: output port, connected to other transaction types (derived from the waiting link and causal link).

- Port type i: input port, connected to transaction types (derived from the causal link).
- Port type info: input and output port, connected to the actor roles that can access this bank.

AT (Aggregate Transaction type)

AT is derived from DEMO aggregate transaction types. It is used as an information bank when an actor requests external information. As represented in Figure 5-4 (AT), only bank is defined. Here, bank shares the same meaning as pBank in T: it contains the facts. AT is connected with the actor role through the port type *info*:

- Port info: input and output port, connected to the actor roles who can access this bank.

5.4.2.3 Actor role (A and CA)

Actor role is a bridge that connects ontology with implementation, as shown in Figure 5-4 (ActorRole). At the ontological level, *actor role* describes responsibilities and authorities, whereas at the implementation level, *actor role* defines the details of executing ontological acts. In other words, as the initiator or executor of a transaction, the actor role is defined as the processor of the ontological acts for implementation.

Through ports, actor roles are linked with transactions as the processors of corresponding acts or with information banks as information users. Three types of ports are defined for the actor role:

- Port type i: input and output port, linked with the transactions initiated by it;
- Port type e: input and output port, linked with the transactions executed by it;
- Port type info: input and output port, linked with the information bank.

5.4.3 Implementation model

In implementation, DEMO is expanded with implementation details: ActorRoleComponent and Resource. The meta-model of implementation is presented in Figure 5-7.

5.4.3.1 Actor Role Component

ActorRoleComponent indicates the subcomponents of the actor role who defines the implementation details, including the following:

- *ExecutionStep*: the blocks for describing business process details. For example, for actor role A1 (Figure 5-8), implementation process pmT1 is defined, which describes how to promise, what the required resources (Staff) are for keeping the promise, and what steps (Order Taking) must be taken to complete this coordination.
- *SeizeResource:* a block for seizing resources if necessary.
- *ReleaseResource:* a block for releasing resources if necessary.

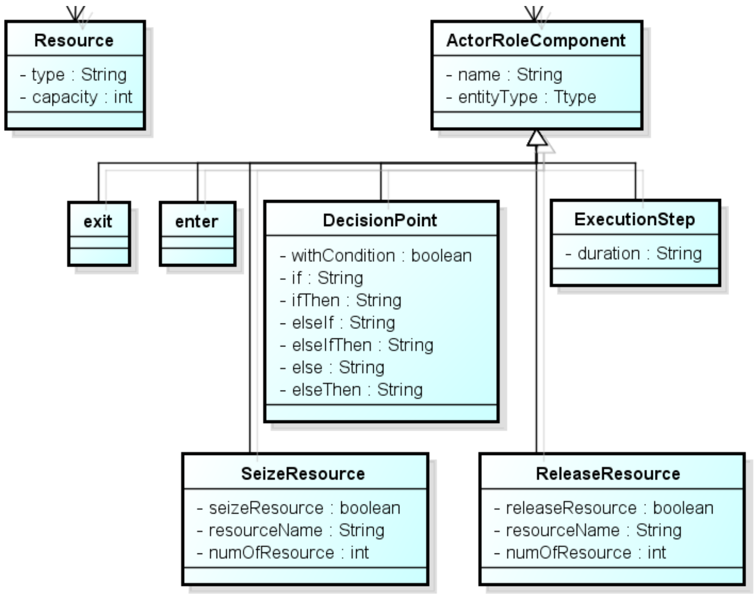


Figure 5-7 Meta-model of the DEMO++ implementation model

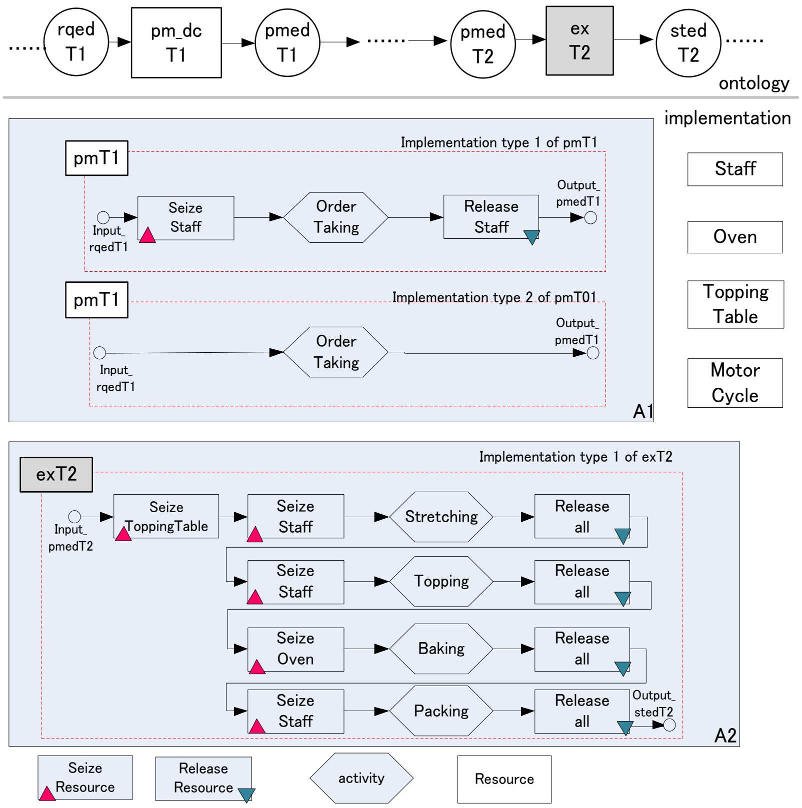


Figure 5-8 Examples of implementation models

5.4.3.2 R (Resource type)

R is utilized in transactions. There are two types of resources: *operator* (an actor who takes responsibility) and *operand* (utilized by the operator). Both can be seized or released by the actor role. The author defines the *actor plays actor role* through this seize-release resource procedure. Seizing an operator resource indicates that the actor role is played by this type of resource.

5.4.3.3 Discussion of Implementing Ontology

Figure 5-8 describes the implementation model of A1 (the order completer) and A2 (the baker) at Buono. For A1, two different implementations are available for promising the “order complete”: at the window or by phone, which requires staff resources, or through the online ordering system, which has no resource requirement. For A2, the detailed execution process of preparing the pizza is defined: stretching, topping, baking and packing, with resource requirements.

Ontology could be implied differently. The key factors of different implementations are concluded from the following two aspects:

- Who plays the actor role?
- Actors are an organization's resources. The actor playing the actor role is represented as an actor role in seizing one type of resource (operator) if necessary. As described in the actor-actor role mapping in Table 5-1, one type of actor (resource) can play different ontological actor roles: at Buono, any staff member can play all three elementary actor roles, A1, A2 and A3. Meanwhile, one actor role can be played by different actors (resources): for example, if there are two types of staff members in the store, staff type one can perform only the duties inside the store, such as taking orders and preparing pizza; staff type two responds by delivering the pizza. Then, it is possible to define another implementation plan: staff type one can play actor roles A1 and A2, and staff type 2 can play actor role A3, as shown in Table 5-2. *Actor-actor role mapping* explains how humans interact by playing actor roles. Although the ontology remains the same, different human interactions during implementation will affect the effectiveness of communication in cooperation and resource utilization and, consequently, the entire process. In our method, it is simple to change the actor-actor role mapping plan to illustrate the effects of different implementation plans in the simulation model.

Table 5-1 Actor-actor role mapping table plan one (Case C)

Actor Role / Actor	A1 Order Completer	A2 Order Preparer	A3 Order Deliverer
Stuff	A	A	A

Table 5-2 Actor-actor role mapping table plan two (Case C)

Actor Role / Resource	A1 Order Completer	A2 Order Preparer	A3 Order Deliverer
Stuff type1	A	A	
Stuff type2			A

- How to play the actor role

  Implementation relates to how ontology is applied in an enterprise; different implementation plans are possible for the same ontological design. As illustrated in Figure 5-8, two implementations are defined for A1, corresponding to the same ontological act pmT1. In reality, all possible implementation plans could be defined independently in the corresponding actor roles, so it is possible to choose different approaches to compare the effects. The changes are controlled: implementation changes will not affect the ontological level, and the ontological changes will not affect implementation if it is not necessary. For example, all of the execution phases within T2 could be outsourced to other companies. In this situation, we only need to change the implementations within A2, with no change to the ontological model. Similarly, if we need to enable customer payment through the Internet using a credit card, it is not necessary to wait for delivery to be accepted (acedT3) to request the payment (rqT4). We can change the ontological model to make "acedT3" a condition of "exT1" instead of "rqT4". However, this ontological change does not change the implementation of any acts.

5.4.4 Main model

Main contains both ontological and implementation components, with inter-connections (ICs) between the components defined. The meta-model of Main is defined as shown in Figure 5-9.

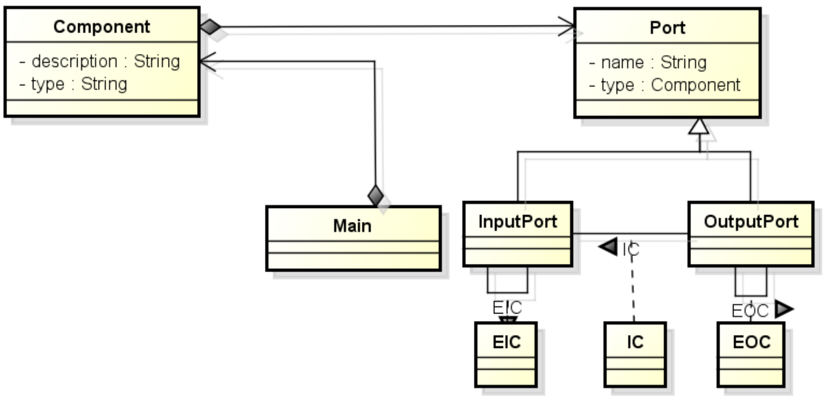


Figure 5-9 Meta-model of DEMO++ Main

An example of component "Main" for Buono is given in Figure 5-10. Component ports are connected by ICs. ICs between transaction types are derived from the PSD's causal and waiting links. ICs between transaction types and actor roles are derived from the OCD initiator, executor and information links. ICs between the actor role and aggregate transactions are derived from the OCD information link.

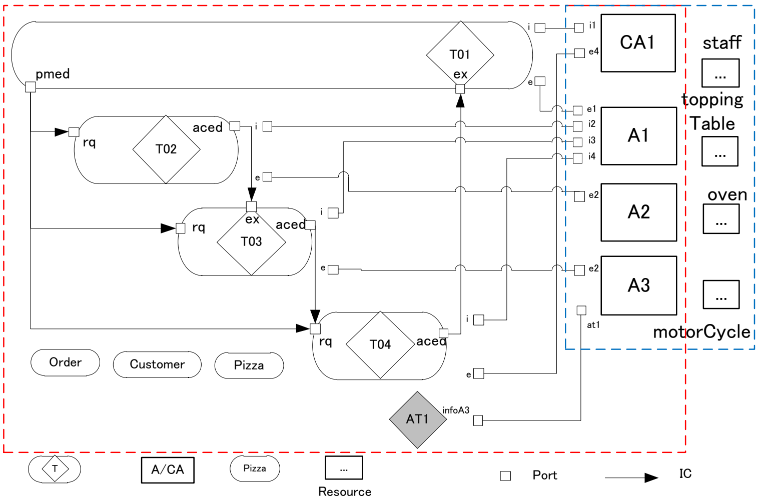


Figure 5-10 Example of the Main component

5.4.5 Meta-model of DEMO++

The entire meta-model of DEMO++ is defined as shown in Figure 5-11. The pink part denotes the DEMO++ ontological level model (as explained in section 5.4.2); the blue part denotes the DEMO++ implementation level model (as explained in section 5.4.3) and the yellow part denotes the DEMO++ main model (as explained in section 5.4.4).

5.5 DEMO++ Library for AnyLogic

To translate DEMO into DEMO++ and, finally, an executable model in the AnyLogic platform, the author developed an AnyLogic DEMO++ library. The DEMO++ AnyLogic library was developed completely following the definition of DEMO++. As shown in Figure 5-12, eight types of components were defined:

- T: following Transaction type defined in DEMO++, defined as an agent in AnyLogic. The transaction type is developed as a T library element; the detailed definition is given in Appendix B.
- AT: corresponding to the AT model defined in DEMO++, defined as an agent in AnyLogic;
- O: corresponding to the O model defined in DEMO++, defined as an agent in AnyLogic;
- AR: corresponding to the AR model defined in DEMO++, defined as an agent in AnyLogic, including AR components inside;
- AR Components: corresponding to the AR component model defined in

DEMO++, defined by the “business process library” provided by AnyLogic;

- R: corresponding to the R model defined in DEMO++, defined by the “Resource” of “business process library” in AnyLogic in the case of simple resources or by an agent in AnyLogic in the case of complex resources;
- Factum: defined as a tuple <name, status, time>, describing a factum. “name” is a string type and denotes the name of the factum; “status” is a Boolean type and denotes whether the factum has been realized; “time” is a double type and denotes the time that the factum becomes true.
- Main: corresponding to the Main model defined in DEMO++;

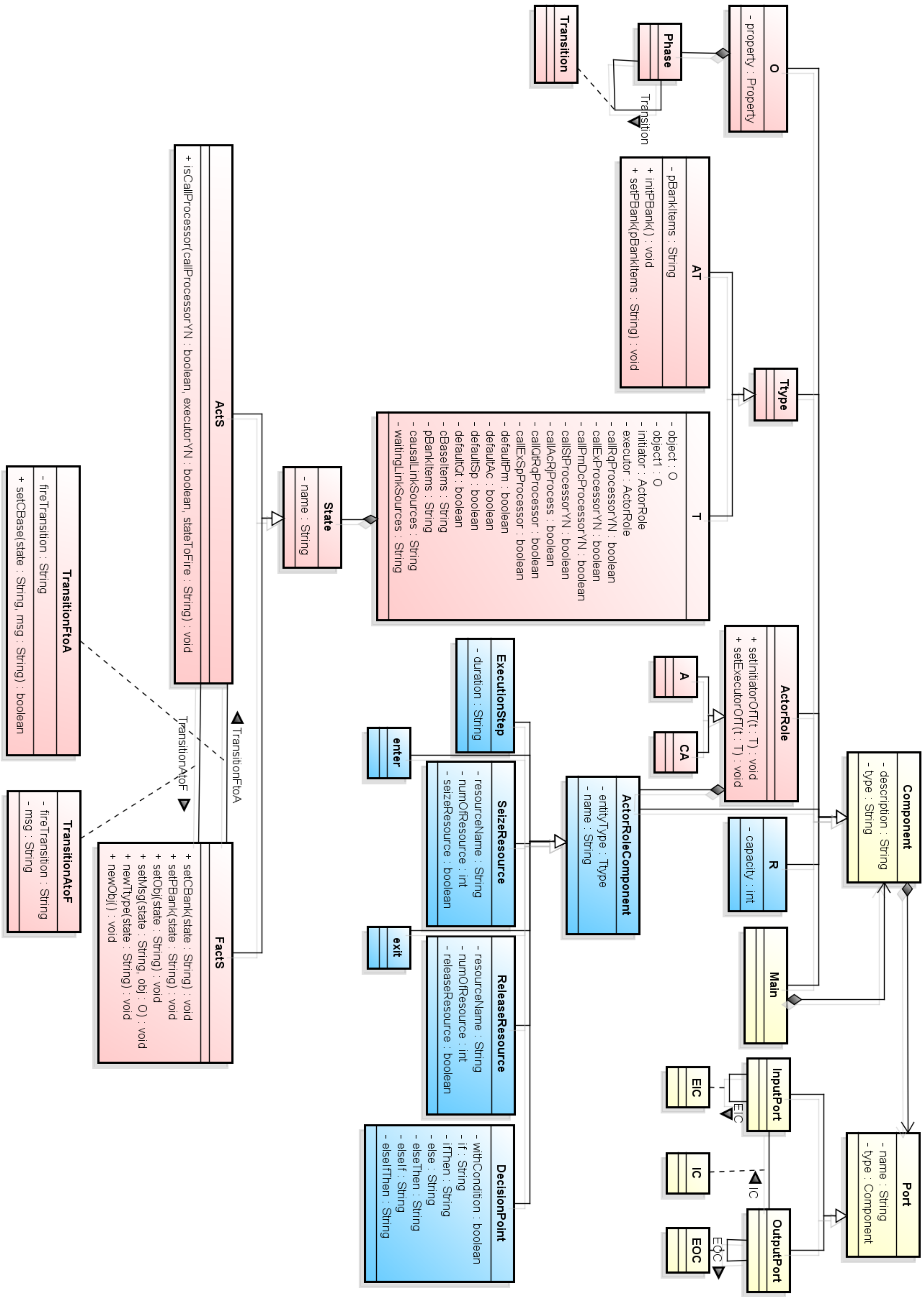


Figure 5-11 Meta-Model of DEMO++

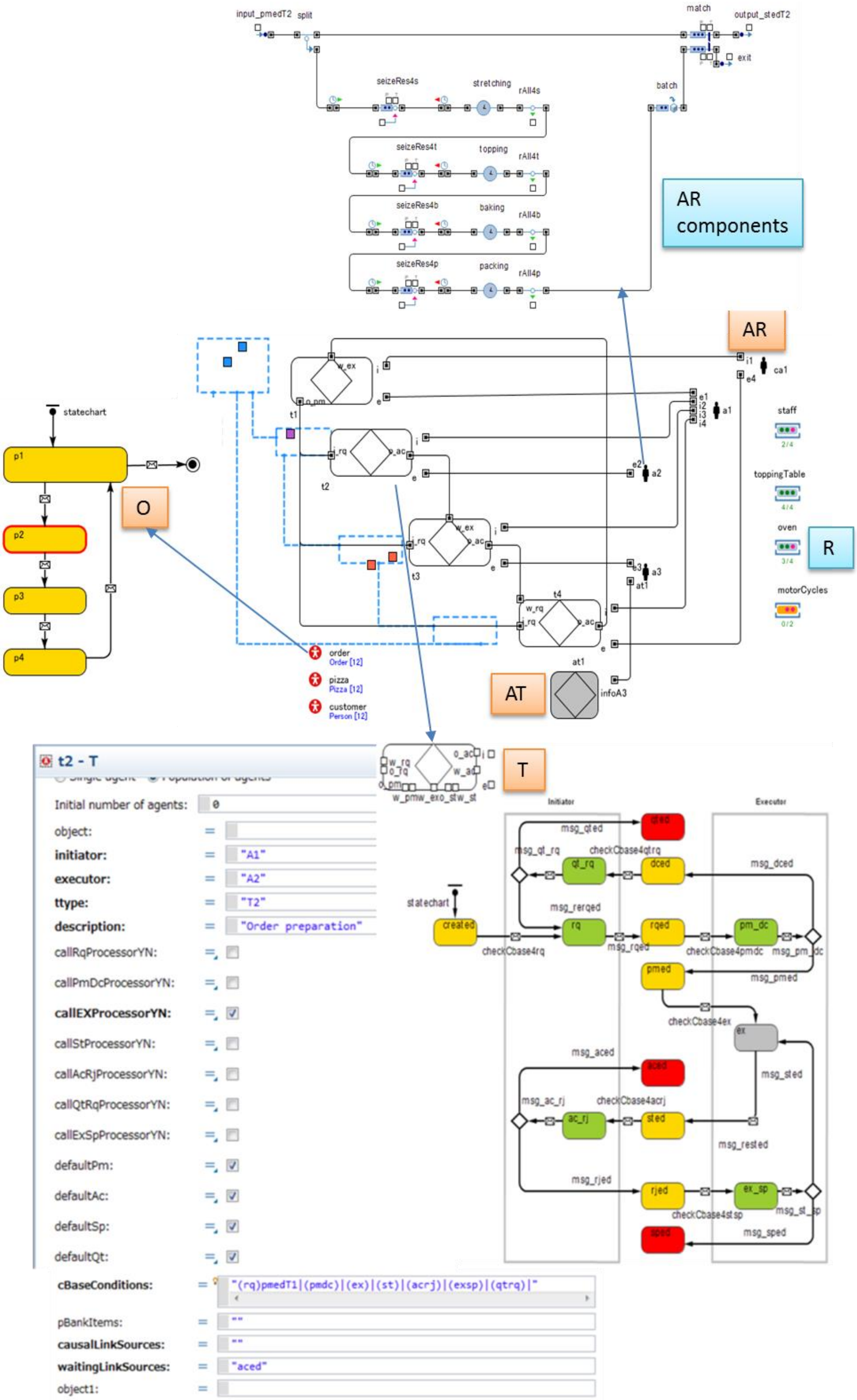


Figure 5-12 DEMO++ AnyLogic library

5.6 Model Transformation

The model transformation was conducted based on pre-defined meta-models of each model and the mappings among them, which are called transformation rules. The meta-model of DEMO was defined in Section 5.3 (Figure 5-2). The meta-model of DEMO++ was defined in section 5.4, Figure 5-11. This structure of the library follows the definitions that were provided in DEMO++ so that they can share the same meta-model.

As described in Figure 5-13, there are four transformation rules that are defined.

- T1: From DEMO CM to DEMO PM.
- T2: From DEMO PM to DEMO AM.
- T3: From the DEMO aspect models to the DEMO++ model.
- T4: From the DEMO++ model to the AnyLogic executable model.

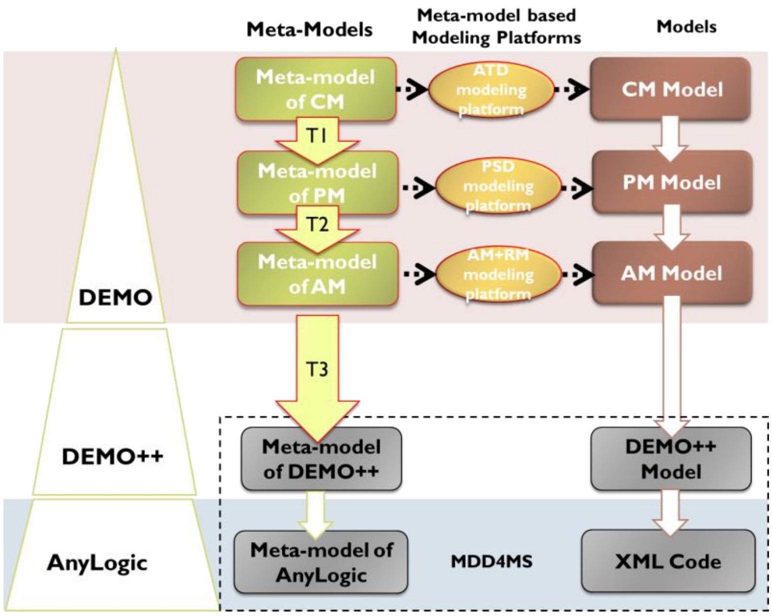


Figure 5-13 Model transformation framework

5.6.1 Model transformation inside the DEMO aspect models (T1 and T2)

There are four aspect models defined in DEMO; each aspect uses different perspectives and details of ontology. To reduce the modeling complexity and defining duplicated reworks in separately creating these four models, the author defined meta-model-based transformation rules to assist in the DEMO-related modeling. There are two included transformations: T1, which is from CM to PM, and T2, which is from PM to AM (see Figure 5-14).

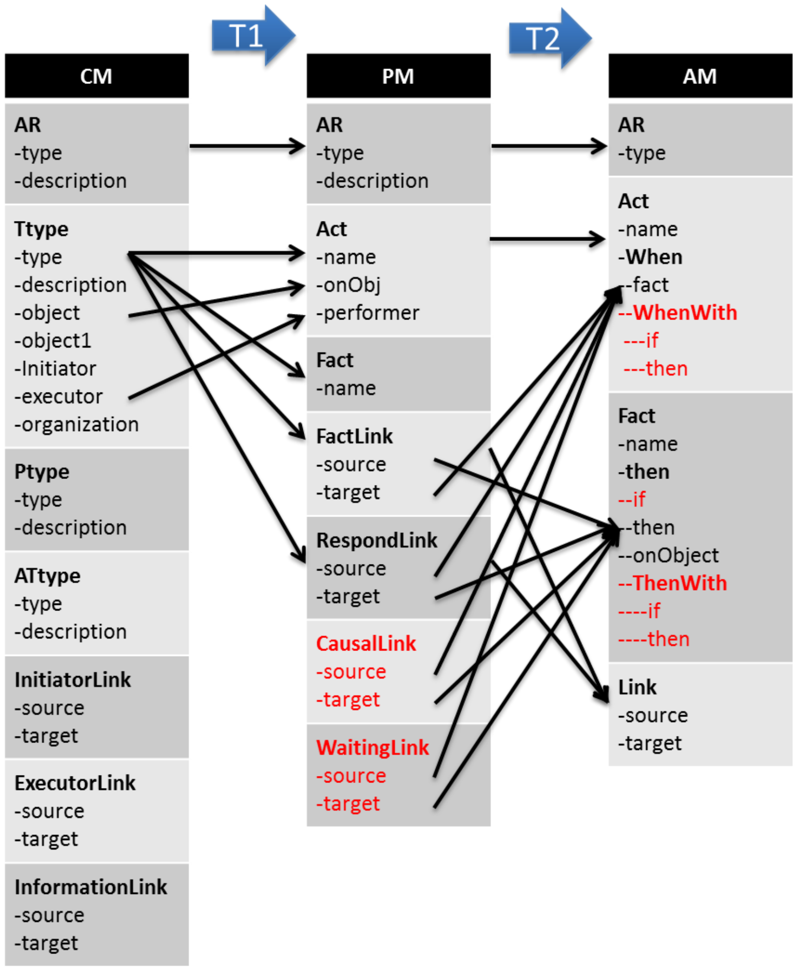


Figure 5-14 Meta-model mapping among the DEMO aspect models

- T1: In this first transformation step, some CM elements, e.g., *AR* and *Ttype* are transformed into PSD elements. AR elements are transformed into PM elements. *Ttype* elements are translated, following a standard transaction pattern, into *Act, Fact, FactLink and RespondLink e*lements. The *FactLink* elements are translated from Act to Fact, while the RespondLink elements are converted from Fact to Act, which occur inside a transaction. The red elements must be manually added because they are related by relationships that exist between transaction types, which cannot be obtained from a CM model.
- T2: In the second transformation step, which is shown in T2 in Figure 5-14, the PM elements are transformed into AM elements. AM.Act.When.fact includes all conditional facts of the act, which can be derived from the FactLink target and

the RespondLink, CausalLink and WaitingLink sources. AM.Fact.Then.then is the reaction after the fact is achieved, which can be derived from the FactLink source and the RespondLink, CausalLink and WaitingLink targets. The red items must be manually completed because this information is not included in the PM. In the WhenWith part, the conditional assignment is defined if necessary. In AM. Fact.Then, the if condition of the reaction must be provided, while in the AM. Fact.Then.ThenWith part, the assignment is defined if necessary.

Beginning with the simplest construction model and by performing these two transformations, the most consistent and complicated action model can be determined.

5.6.2 Model transformation from DEMO to DEMO++

The transformation from DEMO to DEMO++ is conducted to include all aspect models in DEMO.

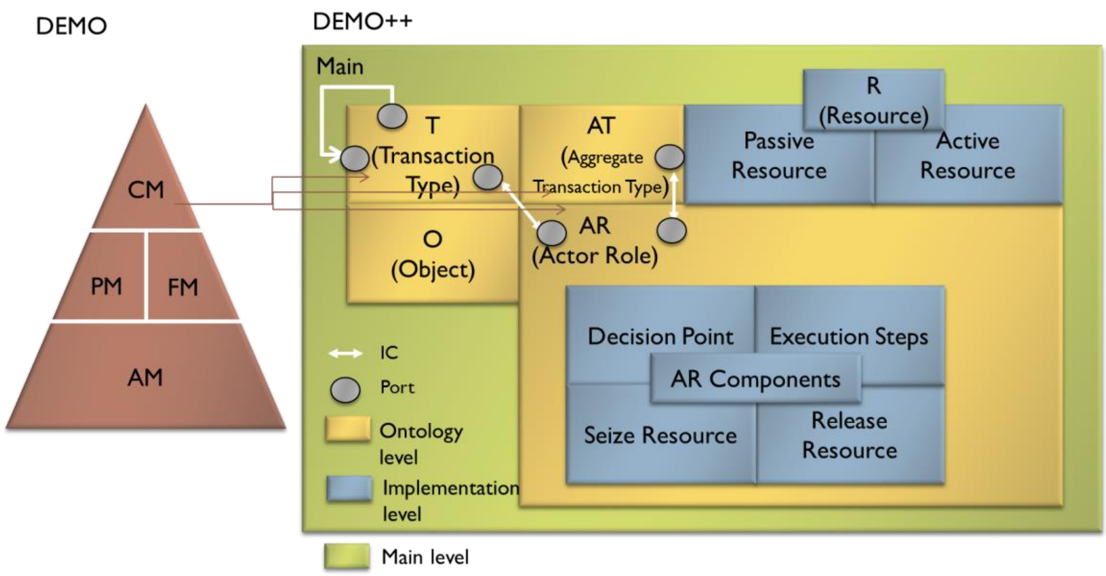


Figure 5-15 Conversion from DEMO to DEMO++

As briefly described in Figure 5-15 and in detail in Figure 5-16, the ontological level model in DEMO++ can be converted from the DEMO aspect models. DEMO++.AR is derived from DEMO CM.AR; DEMO++.T is derived from DEMO CM.Ttype, PM.ACT, PM.FACT, PM.FactLink and PM.RespondLink. The red attributes in Figure 5-16 cannot be obtained in DEMO because they must be manually defined. The DEMO++.Main section concerns the connections between elements, which can be derived from links in DEMO.CM. The implementation level, which is part of the implementation details in the ActorRoleComponents, can be derived from DEMO.AM, which is similar to the decision point. This step is also part of the execution steps because this information is described in the action rule for an actor role. In the other situation, this information must be manually defined.

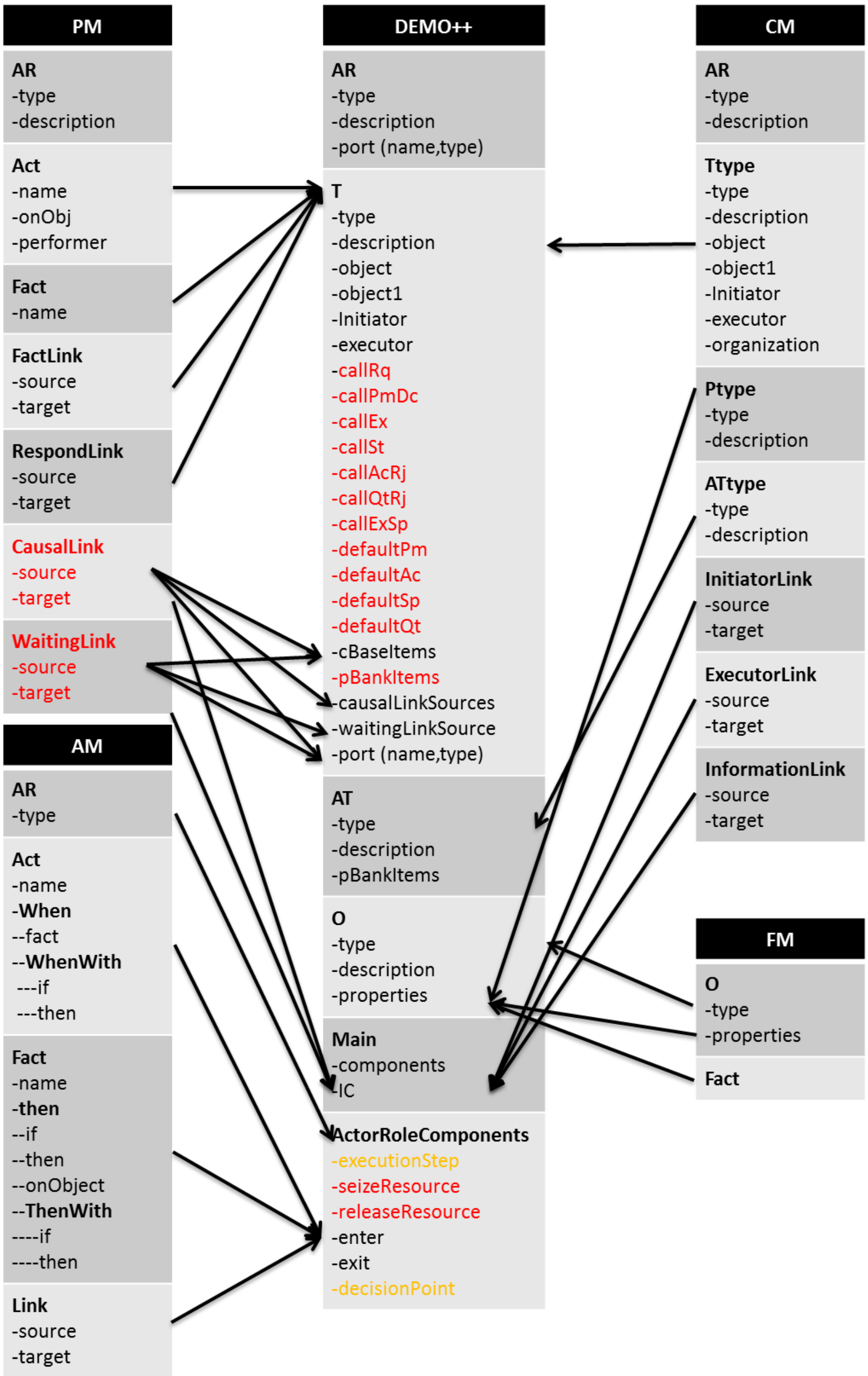
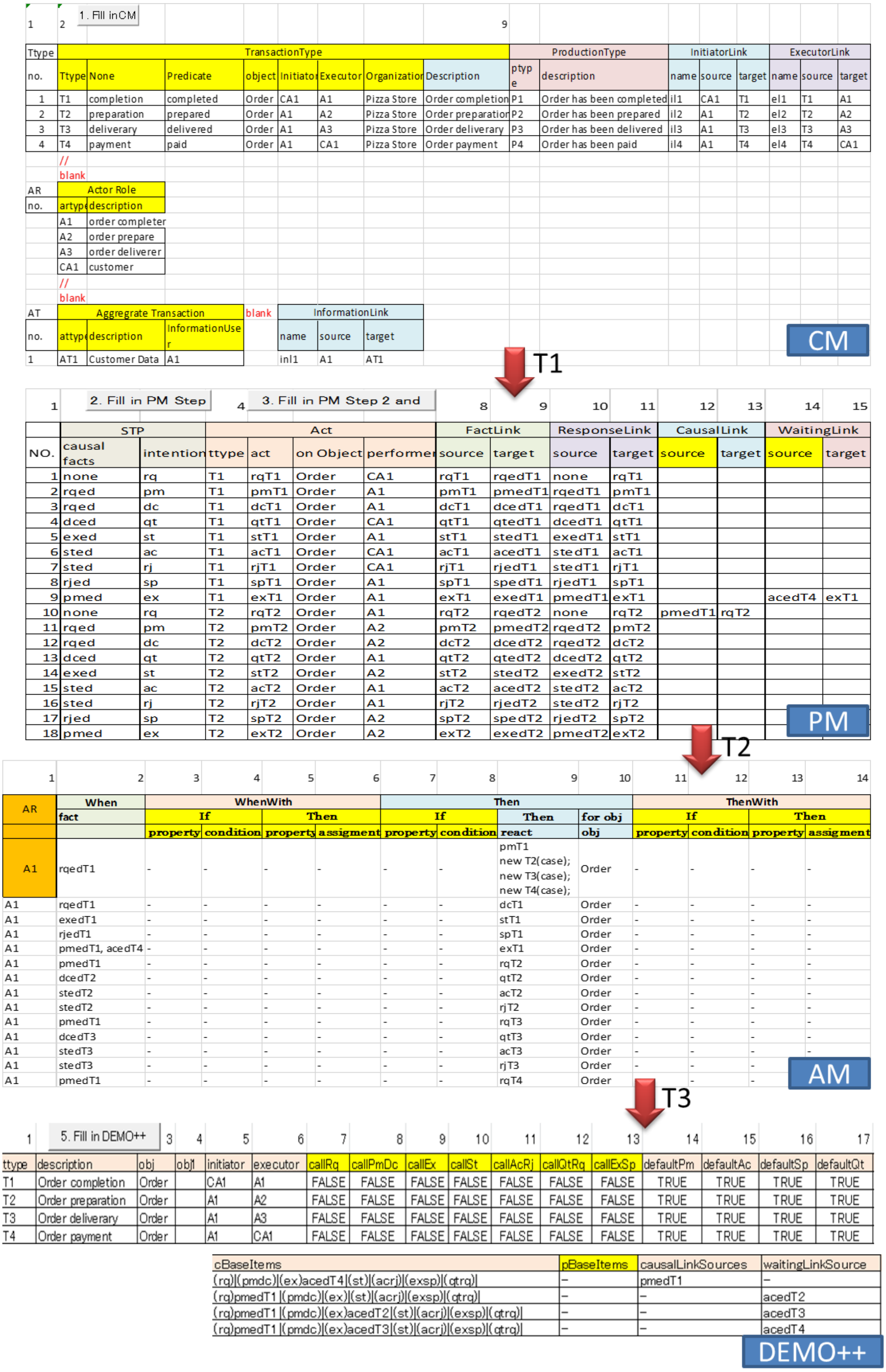


Figure 5-16 Meta-model mapping between DEMO aspect models and DEMO++

The DEMO model specification, which follows a meta-model definition, is ultimately organized as a series of excel files. By using excel, the models can be easily modified and transformed into other types. All possible transactions are programmed to encourage semi-automatic transfers. Only those items that are not covered in the transformation rules must be manually added. The definitions of DEMO CM, DEMO PM, DEMO FM, DEMO AM and DEMO++ follow the meta-models given in chapter 3 and section 5.4. The transition from DEMO CM to DEMO PM to DEMO AM and ultimately to DEMO++ is shown in Figure 5-17.

- T1 is the transition from DEMO CM to DEMO PM. In the CM model, the items that are colored in yellow must be provided manually, such as transaction type items, including type, noun, predicate, object, initiator, executor; actor role items, including type of actor role and description; aggregate transaction items, including type of AT, description and the users of the aggregate transaction items. The other items can be automatically derived.
- T2 is the transition from DEMO PM to DEMO AM. In the PM model, “STP” is the given standard transaction pattern. According to the STP and CM model, the items “Act”, “FactLink” and “RespondLink” can be automatically derived. The items colored in yellow must be manually completed, including the items “source of CausalLink” and “Source of WaitingLink”. The target items can also be derived. From PM, the AM table can be partially derived. In the obtained AM model, the items colored in yellow must be manually completed.
- T3 is the transition from DEMO AM to DEMO++. All AM items are translated into DEMO++ items, for example, cBaseItems. However, whether an act needs to call a processor, the pBaseItems must be manually added.

The obtained DEMO++ parameters are used to define the DEMO++ AnyLogic library elements.



Ttype	TransactionType								ProductionType		InitiatorLink			ExecutorLink		
no.	Ttype	None	Predicate	object	Initiator	Executor	Organizatio	Description	ptype	description	name	source	target	name	source	target
1	T1	completion	completed	Order	CA1	A1	Pizza Store	Order completion	P1	Order has been completed	il1	CA1	T1	el1	T1	A1
2	T2	preparation	prepared	Order	A1	A2	Pizza Store	Order preparatio	P2	Order has been prepared	il2	A1	T2	el2	T2	A2
3	T3	deliverary	delivered	Order	A1	A3	Pizza Store	Order deliverary	P3	Order has been delivered	il3	A1	T3	el3	T3	A3
4	T4	payment	paid	Order	A1	CA1	Pizza Store	Order payment	P4	Order has been paid	il4	A1	T4	el4	T4	CA1
	//															
	blank															

AR	Actor Role	
no.	artype	description
	A1	order completer
	A2	order prepare
	A3	order deliverer
	CA1	customer
	//	
	blank	

AT	Aggregrate Transaction			blank	InformationLink		
no.	attype	description	InformationUser		name	source	target
1	AT1	Customer Data	A1		inl1	A1	AT1

	STP		Act				FactLink		ResponseLink		CausalLink		WaitingLink	
NO.	causal facts	intention	ttype	act	on Object	performer	source	target	source	target	source	target	source	target
1	none	rq	T1	rqT1	Order	CA1	rqT1	rqedT1	none	rqT1				
2	rqed	pm	T1	pmT1	Order	A1	pmT1	pmedT1	rqedT1	pmT1				
3	rqed	dc	T1	dcT1	Order	A1	dcT1	dcedT1	rqedT1	dcT1				
4	dced	qt	T1	qtT1	Order	CA1	qtT1	qtedT1	dcedT1	qtT1				
5	exed	st	T1	stT1	Order	A1	stT1	stedT1	exedT1	stT1				
6	sted	ac	T1	acT1	Order	CA1	acT1	acedT1	stedT1	acT1				
7	sted	rj	T1	rjT1	Order	CA1	rjT1	rjedT1	stedT1	rjT1				
8	rjed	sp	T1	spT1	Order	A1	spT1	spedT1	rjedT1	spT1				
9	pmed	ex	T1	exT1	Order	A1	exT1	exedT1	pmedT1	exT1			acedT4	exT1
10	none	rq	T2	rqT2	Order	A1	rqT2	rqedT2	none	rqT2	pmedT1	rqT2		
11	rqed	pm	T2	pmT2	Order	A2	pmT2	pmedT2	rqedT2	pmT2				
12	rqed	dc	T2	dcT2	Order	A2	dcT2	dcedT2	rqedT2	dcT2				
13	dced	qt	T2	qtT2	Order	A1	qtT2	qtedT2	dcedT2	qtT2				
14	exed	st	T2	stT2	Order	A2	stT2	stedT2	exedT2	stT2				
15	sted	ac	T2	acT2	Order	A1	acT2	acedT2	stedT2	acT2				
16	sted	rj	T2	rjT2	Order	A1	rjT2	rjedT2	stedT2	rjT2				
17	rjed	sp	T2	spT2	Order	A2	spT2	spedT2	rjedT2	spT2				
18	pmed	ex	T2	exT2	Order	A2	exT2	exedT2	pmedT2	exT2				

AR	When	WhenWith				Then				ThenWith			
	fact	If		Then		If		Then	for obj	If		Then	
		property	condition	property	assigment	property	condition	react	obj	property	condition	property	assigment
A1	rqedT1	-	-	-	-	-	-	pmT1 new T2(case); new T3(case); new T4(case);	Order	-	-	-	-
A1	rqedT1	-	-	-	-	-	-	dcT1	Order	-	-	-	-
A1	exedT1	-	-	-	-	-	-	stT1	Order	-	-	-	-
A1	rjedT1	-	-	-	-	-	-	spT1	Order	-	-	-	-
A1	pmedT1, acedT4	-	-	-	-	-	-	exT1	Order	-	-	-	-
A1	pmedT1	-	-	-	-	-	-	rqT2	Order	-	-	-	-
A1	dcedT2	-	-	-	-	-	-	qtT2	Order	-	-	-	-
A1	stedT2	-	-	-	-	-	-	acT2	Order	-	-	-	-
A1	stedT2	-	-	-	-	-	-	rjT2	Order	-	-	-	-
A1	pmedT1	-	-	-	-	-	-	rqT3	Order	-	-	-	-
A1	dcedT3	-	-	-	-	-	-	qtT3	Order	-	-	-	-
A1	stedT3	-	-	-	-	-	-	acT3	Order	-	-	-	-
A1	stedT3	-	-	-	-	-	-	rjT3	Order	-	-	-	
A1	pmedT1	-	-	-	-	-	-	rqT4	Order	-	-	-	

ttype	description	obj	obj1	initiator	executor	callRq	callPmDc	callEx	callSt	callAcRj	callQtRq	callExSp	defaultPm	defaultAc	defaultSp	defaultQt
T1	Order completion	Order		CA1	A1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
T2	Order preparation	Order		A1	A2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
T3	Order deliverary	Order		A1	A3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
T4	Order payment	Order		A1	CA1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE

cBaseItems	pBaseItems	causalLinkSources	waitingLinkSource
(rq)|(pmdc)|(ex)acedT4|(st)|(acrj)|(exsp)|(qtrq)|	–	pmedT1	–
(rq)pmedT1 |(pmdc)|(ex)|(st)|(acrj)|(exsp)|(qtrq)|	–	–	acedT2
(rq)pmedT1 |(pmdc)|(ex)acedT2|(st)|(acrj)|(exsp)|(qtrq)|	–	–	acedT3
(rq)pmedT1 |(pmdc)|(ex)acedT3|(st)|(acrj)|(exsp)|(qtrq)|	–	–	acedT4

Figure 5-17 Model transformation

Following DEMO++, a DEMO++ library is performed in AnyLogic, including the AR, AT, T, O, Main and Factum elements. Library element T is illustrated in Figure 5-18. The parameters for T are obtained from the derived DEMO++ model.

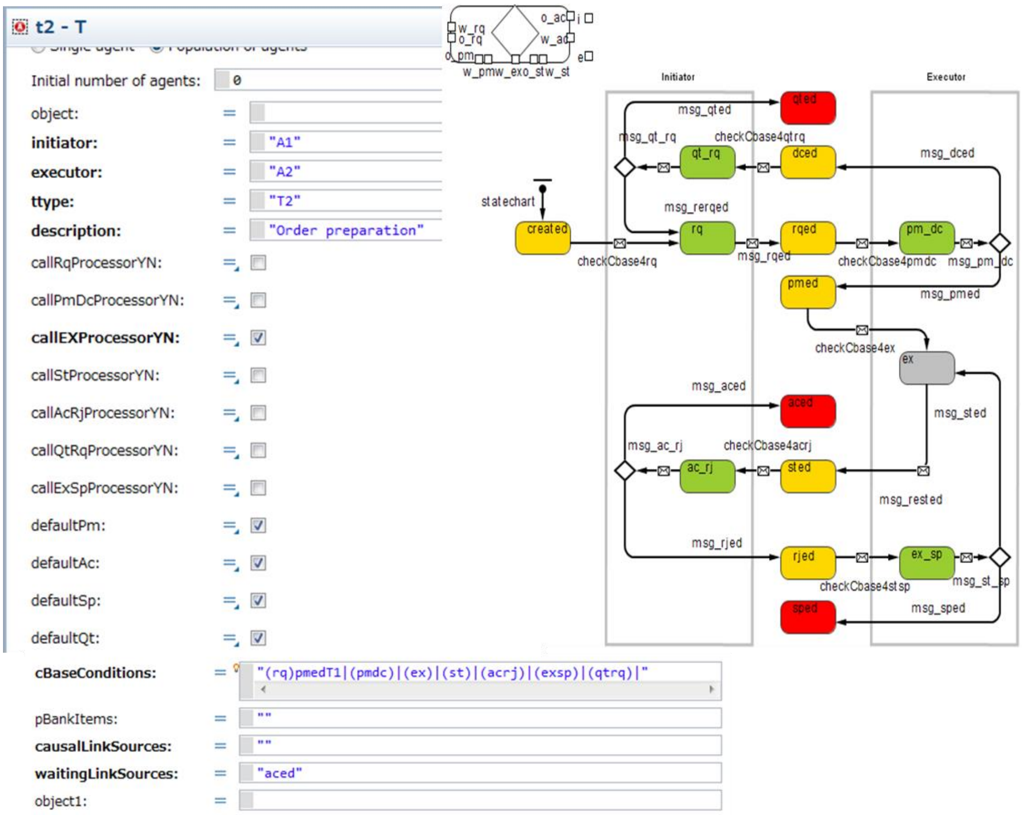


Figure 5-18 T in the DEMO++ library

The mappings among DEMO, DEMO++ and AnyLogic library elements are concluded and listed in Table 5-3.

Table 5-3 Meta-model mapping table

DEMO		DEMO++		AnyLogic	
Objects	Links	Objects	Links	Objects	Links
Model		Model		Model	
				ActorRole	
A		A		A	
CA		CA		CA	
T		T		T	
AT		AT		AT	
	InformationLink		InformationLink		IC
	InitiatorLink		InitiatorLink		
	ExecutorLink		ExecutorLink		
Otype		Otype		O	
Property		Property		O.property	
	Ptype		Ptype	O.Phase	
	Ftype		Ftype	O.property	
Act		Act		ActS	
Fact		Fact		FactS	
	factLink		factLink		TransitionFtoA
	respondLink		respondLink		TransitionAtoF
	causalLink		causalLink		IC
	waitingLink		waitingLink		IC
Act.When		Act.When		actS.cbase	
Act.When.WhenWith		Act.When.WhenWith		actS.pbase	
Fact.Then		Fact.Then		factS.	
Fact.Then.ThenWith		Fact.Then.ThenWith			
		ExecutionStep		Delay	
		DecisionStep		SelectOutput	
		SeizeResource		SeizeResource	
		ReleaseResource		ReleaseResource	
		Resource		R	

5.7 Case C: DEMO++ Based Business Process Simulation

The simulation based on DEMO++ was conducted in AnyLogic. The main model is represented in Figure 5-19, following the Main component in DEMO++.

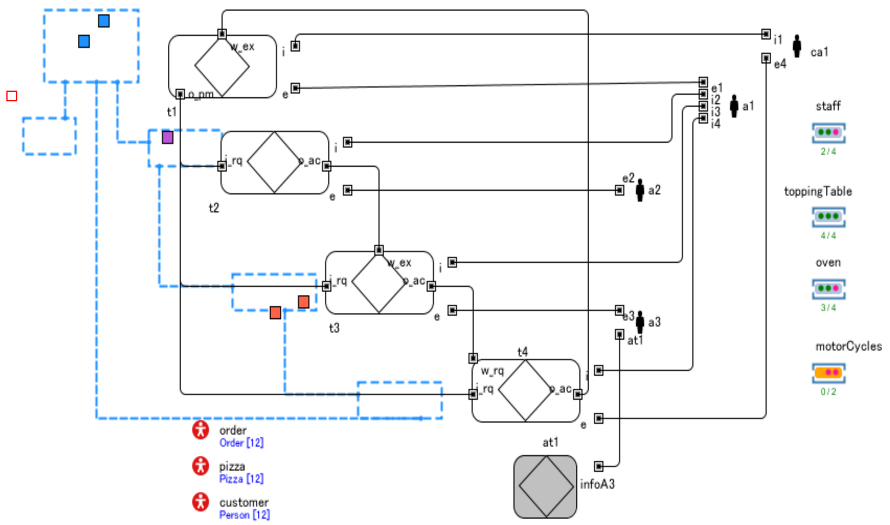


Figure 5-19 Main for Buono Pizza

The object type “Order” shows the possible states of this object type, which are shared by all instances. For example, the instance “o5” is represented by red lines in Figure 5-20, and its state transition chart is as presented in Figure 5-20. At the moment, this object is in state “P2”, which is related to transaction T2.

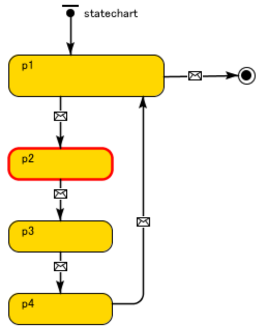


Figure 5-20 State transition diagram of order instance “o5”

Figure 5-21 shows the state transition pattern of transaction type T1. At the moment, the instance T1 (o5) is in “exT2”, meaning that order (o5) is in preparation. The implementation model of exT2 is defined in actor role A2, including stretching, topping, baking and packing (Figure 5-22).

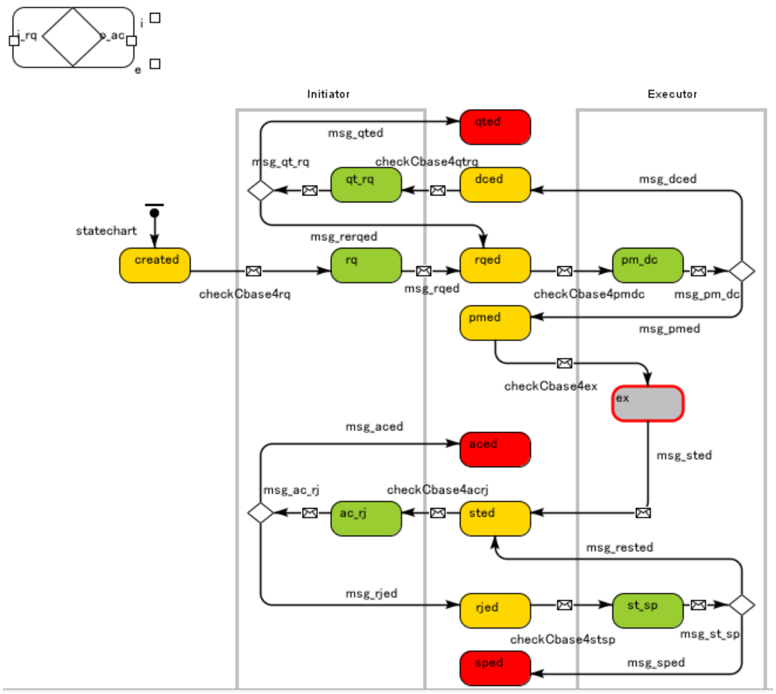


Figure 5-21 Transaction pattern of transaction T1

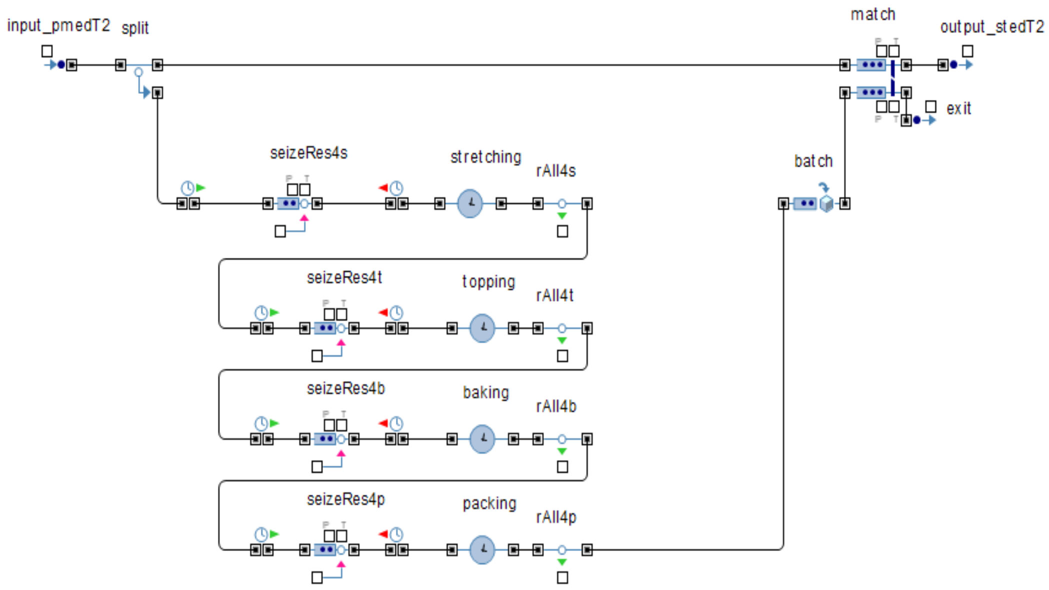


Figure 5-22 Implementation phase of [T2/ex]

Based on the description of the Buono Pizza case, the author established experimental parameters, including the number of resources and time for each action, as shown in Figure 5-23.

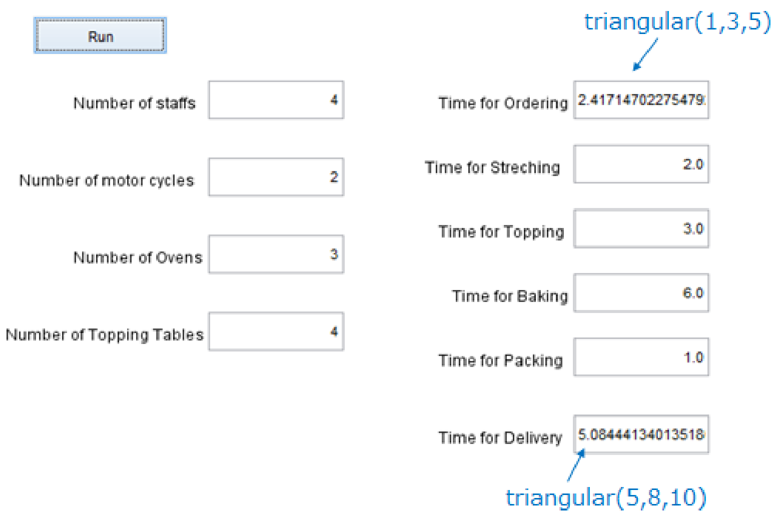


Figure 5-23 Experimental parameter setting

The simulation results are presented in Figure 5-24. The upper-left chart shows that the customer waiting time was concentrated between 20 and 30 minutes. With the current number of resources and processes, approximately 25% of customers wait for more than 30 min to receive their pizzas (upper-right pie-chart in Figure 5-24). The staffs and motors have higher utilization than the other resources (the lower-left chart in Figure 5-24), and thus these two resources are considered bottleneck resources. At the current modeling time, there is no line waiting for ordering, topping, baking or delivering (the lower-right chart in Figure 5-24).

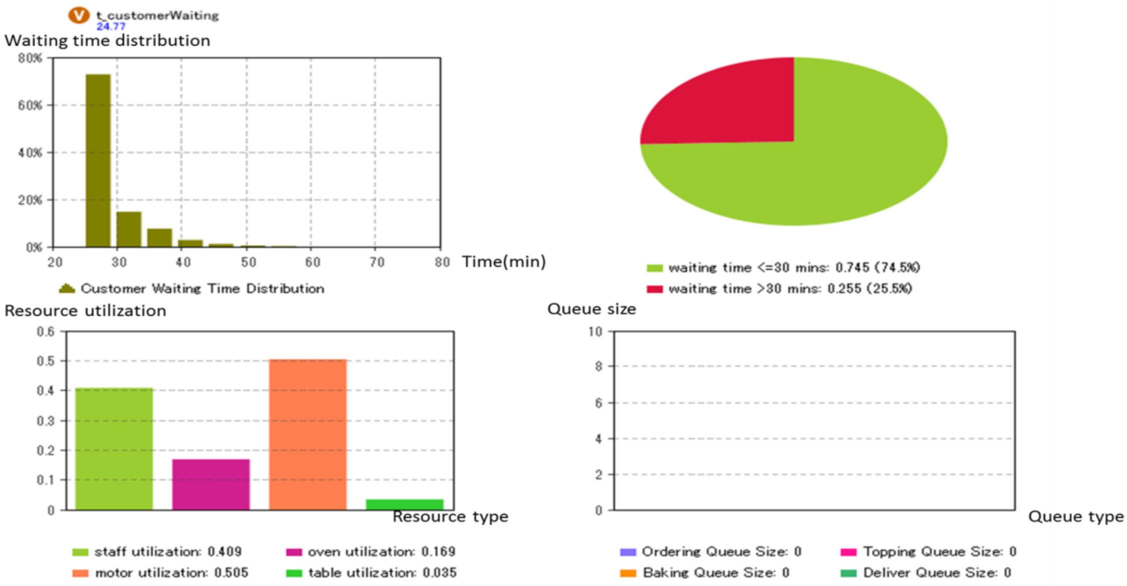


Figure 5-24 Simulation result (4 staff, 4 preparation tables, 3 ovens, 2 motors)

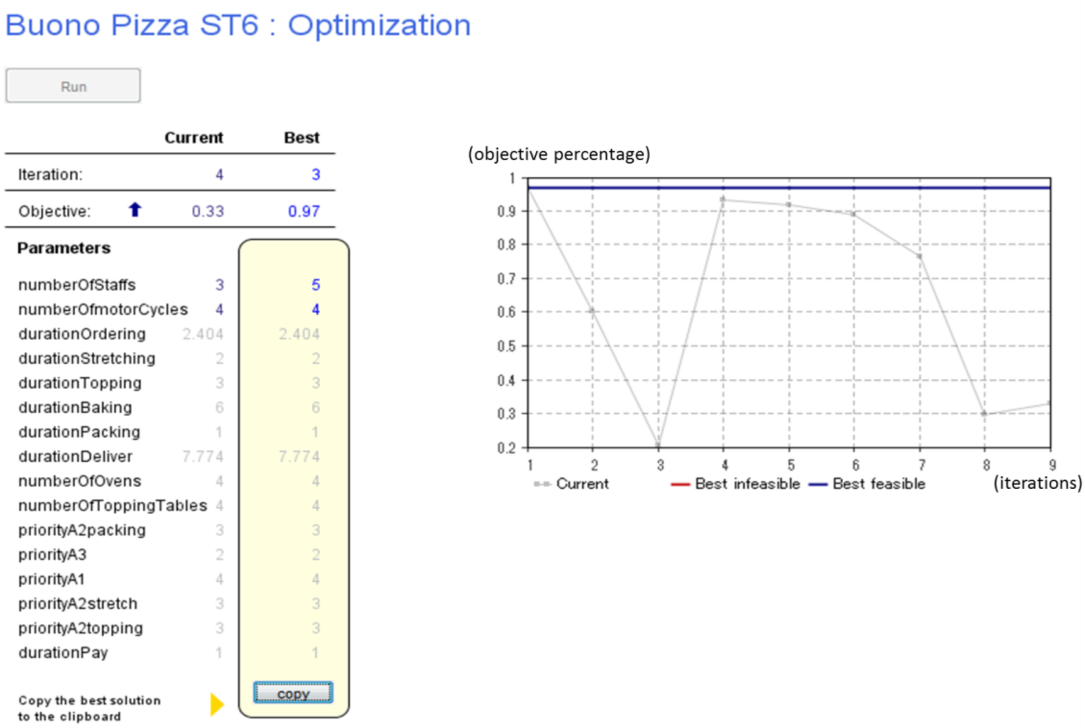


Figure 5-25 Optimization result

This observation suggests that Buono’s problem is mainly a function of resource allocation. Because they were already using an online ordering system, no significant changes were required in their business processes. By optimizing (Figure 5-25) the bottleneck resources, the staff and motors, the author concluded that the percentage of >30-min delivery time could be greatly reduced if there were 6 staff members and 4 vehicles available. This best parameter set gives the best solution, with a >30-min delivery rate of 3%.

5.8 Discussion

The only unchanging concept is change itself. However, traditional business process simulations are inadequate for describing complex systems; it is difficult to change models to simulate reengineering, and the reengineering process cannot be repeated because of a lack of systematic approaches. These limitations are caused by the use of the workflow viewpoint and by the low abstraction levels found in conceptual modeling as well as by indistinguishable opinions on design and implementation. These concerns are limitations not only in BPS but also in BPR.

Enterprise engineering is a promising research area because of its ability to reduce complexities and provide a design-focused perspective and human-centered interaction. Enterprise engineering enables reengineering to be modularized and controllable. The methodology takes an enterprise as a whole and investigates its core structures with separate designs and implementation. In addition, it uses a human-centered viewpoint and considers the social aspect of enterprise as well as its processes. However, the advantages of enterprise engineering have not been fully investigated in enterprise-change-related business process simulations. A DEMO-based Petri net model (Dietz & Barjis, 2000) explained how we can use an ontology as an executable model. However, the ontology alone is not sufficient for enterprise reengineering simulations, because many changes are related to implementation. The main problem for those who use an enterprise ontology in their work is how to connect the ontology with the implementation.

The proposed DEMO++ is a conceptual model that expands and combines an ontology with implementation in the context of enterprise engineering to assist in business process simulation. DEMO++ concerns not only the abstracted essence of an enterprise but also its implementation. In DEMO++, the business process is controlled in its ontology; decision making and implementation are called for only when necessary during implementation. The separately defined core and implementations reduce the complexity of modeling complex systems and increase the flexibility and agility of simulation models. In addition, DEMO++-based conceptual models make changes controllable. In some situations, the ontological model must be changed. These are always fundamental changes in BPR; in most cases, the changes only involve providing an alternative implementation plan for the same ontological objective. Regardless of the change types, the changes are within certain components.

Compared with traditional workflow-based business process modeling and simulation methods, DEMO++ is a simulation as well as analysis method. This method can be properly connected with management for analyzing problems, seeking solutions and evaluating alternative plans for enterprise reengineering.

Another contribution of this research is considered from a practical viewpoint. By clarifying the differences and dependencies between the ontological and implementation models, our methodology proposed a generic framework for generating a modularized, component-based simulation model with increased reusability. The proposed conceptual model DEMO++ was developed as an AnyLogic DEMO++ library and can be reused in other DEMO++-based simulations.

The methodology is utilized for change analysis. The method whereby the simulation is built allows one to support several types of construction changes, including ontology changes; execution process changes; organizational structure changes, such as different actor-actor role mappings; cooperation method changes; and resource-related changes. To explain the entire concept of enterprise-engineering-based business process simulations without making the description too complex to be understood, only a very simple change in resource allocation is analyzed in this chapter, as shown in Table 5-4. In the pizza case C, “staff” is considered as passive resources without a knowledge structure, skill level definition, or any other social factors. A more complex case will be analyzed in the next chapter to explain how this method can assist in terms of real enterprise changes.

Table 5-4 Change type validation table for “Case C”

Cases	Ontology				Implementation				
	Type 1.1 CM	Type 1.2 PM	Type 1.3 FM	Type 1.4 AM	Type 2.1 How to coordinate	Type 2.2 Who coordinate	Type 2.3 Passive Resource	Type 2.4 Active Resource	Type 2.5 Others
Case C							S		

CHAPTER 6 A Case Study of Business Process Simulation

6.1 Introduction

The business process analysis and business process simulation in the context of enterprise engineering was introduced in previous chapters. This methodology is utilized in a real-world case study (Case D): "Company D". "Company D" is a large Japanese information system integrator. This case study is a project supported by "Company D" to analyze its "proposal and estimation process" using our methodology (from September to December 2014). The objectives are as follows:

- To evaluate the advantages, potentials and limitations of the methodology and to further investigate how the proposed business process simulation can assist in business process change;
- To standardize and demonstrate the process of developing a DEMO++-based simulation for future application (e.g., the "simulation of a Japanese pension system" project); and
- To observe problems in the "proposal and estimation" process and to provide suggestions for improvement.

6.2 Case Description (Case D)

The proposal and estimation process of "Company D" is described in the following section. In the description, the business-level activities are denoted in red, information-level activities are denoted in green, and data-level activities are denoted in blue. Flowcharts are listed in Appendix A.

The customers' requirements are first sent to the salesperson in charge in "Company D". The sales department will evaluate the customer to decide if the case can be accepted for proposal and estimation. The customers` relevant information will first be checked as follows:

(1) Are they a new customer?
(2) What are their credits?
(3) What is the financial situation of the company? What information do we have

about their profits and robustness?

The organizational risk evaluation criteria for the customer are applied. If the customer can meet the criteria, they will be accepted. If the customer cannot meet any of the criteria, they will be rejected. In situations in which the case is complex, the sales manager decides whether to initiate a case receipt symposium (CRS). In requesting a CRS, the salesperson in charge must prepare all related documents and schedule the symposium. The case is discussed in a CRS. Moreover, if the risk is acceptable, the case reception is accepted; otherwise, it is rejected. After discussion, the CRS record is organized and confirmed by the sales manager.

When a case reception is accepted, "Company D" must evaluate the risk of the case. In the risk evaluation process, the salesperson in charge prepares all of the documents. The evaluation is performed within the sales department. There are four levels of risks: 'S' indicates super-high-risk cases; 'A' indicates high-risk cases; 'B' indicates normal-risk cases; and 'C' indicates low-risk cases. According to the risk level, the sales manager decides whether to request a prior review board (prior RB or PRB). If a PRB is necessary, the salesperson in charge prepares all required documents for a PRB and sends them to the RB office. When the RB office receives the request, they prepare and make all arrangements, including coordinating between all participants. A PRB discusses the possibility of making a proposal to the case. If the PRB determines that the risk level of the case is too high to be accepted, the case reception is canceled. If the PRB does not provide a conclusion, the risk must be re-evaluated. If the PRB reaches an agreement, then the case moves into the proposal and estimation phase. After discussion, the PRB record is organized by the salesperson and confirmed by the sales manager.

The salesperson prepares a proposal estimation and a proposal that refers to the PRB record, the customer information, and the risk review sheet. After the documents are prepared, the salesperson must submit the proposal for review. The reviewers include other staff members from the sales department, the sales manager and staff from the development department. If the proposal is not acceptable, it must be redone. In contrast, if the proposal is acceptable, the salesperson requests an estimation symposium to evaluate the estimate. The salesperson also responds by preparing the required documents for the estimation symposium; he/she also arranges the symposium. In the symposium, the estimate is evaluated following risk evaluation rules. The evaluation is produced from four main dimensions. Each dimension includes several key factors:

(1) The profit of the project;
(2) The customer (as mentioned above);
(3) The system, including the novelty, difficulty, particularity, and security; and
(4) The project, including the required time, application area and other conditions.

An unacceptable estimate must be redone. For any accepted proposals and estimates, the sales manager decides if a regular review board (regular RB or RRB) is necessary based on the risk level of the proposal. If a regular RB is necessary, the salesperson prepares the documents and submits a request to the RB officer. Upon receiving a regular RB request, the RB officer prepares and arranges an RB. In a regular RB, high-risk proposals and estimates are discussed and re-evaluated. In the case of an "NG", the accepted case reception is immediately canceled, although the percentage of these cases is very small. In most cases, if there is any problem in the proposal or the estimate, the proposal is modified. If the proposal is acceptable, the regular RB record is prepared and submitted to the QA department for a commitment. The QA department decides whether to ask for an executive symposium according to the RB result and the risk level of the case. If an executive symposium is required, the sales manager must prepare the documents for an executive symposium and submit them to the executive office. Members who attend the executive symposium typically include the sales manager and the executive officers. If the proposal is not acceptable, it must be re-proposed. If the proposal is committed, it is proposed to the customer as a final solution.

6.3 DEMO Aspect Models

As described in section 6.2, there are four main functional requirements in the proposal and estimation processes: (F1) case reception; (F2) case risk evaluation, where 'S'- and 'A'-level cases may request another evaluation process; (F3) proposal and estimation; and (F4) proposal and estimation evaluation according to the risk level. The process was analyzed and abstracted into the DEMO aspect models.

6.3.1 CM

The construction model is shown in Figure 6-1. Nine transaction types are defined to indicate different objectives.

Transaction types T1 (Case proposal), T2 (Case receipt) and T3 (High-risk case receipt) are defined for the case reception; Transaction types T4 (Case evaluation) and T5 (High-risk case evaluation) are defined for the case risk evaluation; Transaction type T6 (Proposal completion) is defined for the proposal and estimation; and Transaction types T7 (Proposal evaluation), T8 (High-risk proposal evaluation) and T9 (Super-high-risk proposal evaluation) are defined for the proposal and proposal evaluation.

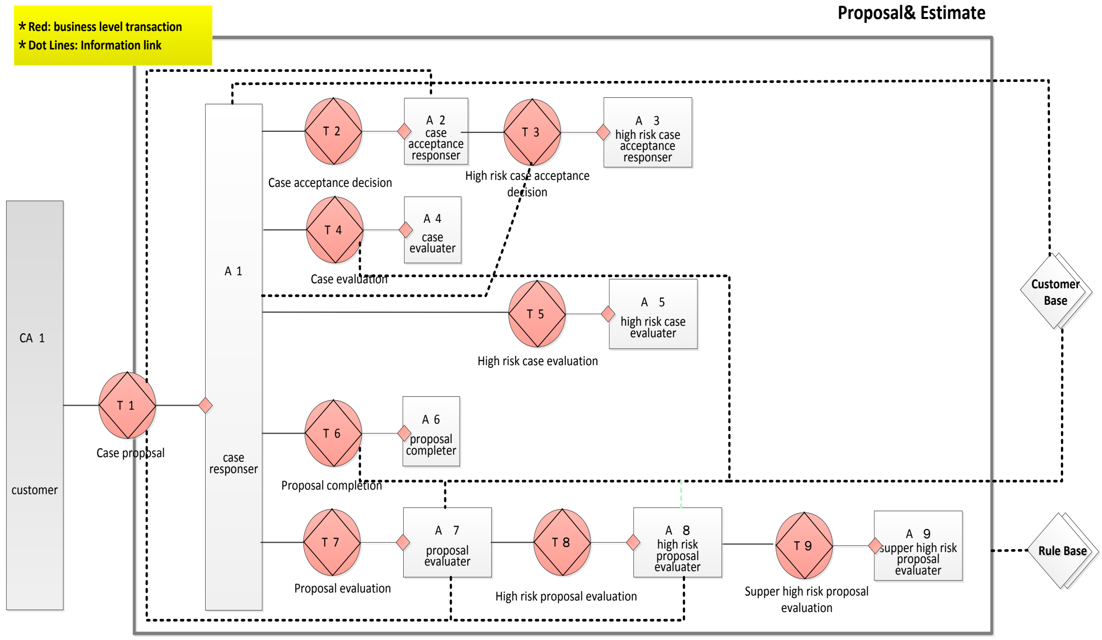


Figure 6-1 OCD of the "Company D" proposal and estimation process

Table 6-1 TPT of CAC proposal and estimation process

Transaction Type	Production Type
T1 Case Proposal	P1 Case S has been proposed
T2 Case receipt	P2 Case S has been receipted
T3 High-risk case receipt	P3 High-risk case S has been receipted
T4 Case evaluation	P4 Case S has been evaluated
T5 High-risk case evaluation	P5 High-risk case S has been evaluated
T6 Proposal completion	P6 Proposal P has been completed
T7 Proposal evaluation	P7 Proposal P has been evaluated
T8 High-risk proposal evaluation	P8 High-risk proposal P has been evaluated
T9 Super-high-risk proposal evaluation	P9 Super-high-risk proposal P has been evaluated

There are two information banks:

- AT1: The customer base includes all customer-related information (e.g., company name, sales turnover, and business).
- AT2: The rule base includes all rules related to the proposal and estimates inside the company (e.g., risk-level division rules).

The product of each transaction type is described in Table 6-1. For illustrative purposes, the production type of transaction type T1 is P1, "Case S has been proposed".

6.3.2 PM

A PSD model is subsequently defined to further describe the details of the processes. From the construction model, part of the PSD model (transactions) can be derived so that only the waiting links between these transactions must be manually added to define the constraints. The PSD model for company D is shown in Figure 6-2 and described in the following sections.

- The promise of T1 "case proposal" [T1/pm][4] must wait until the acceptance of T2 "case receipt decision" is complete (T2/ac)[5].
- In T2 ("case receipt decision"), some of the "high-risk cases" must also be evaluated by T3 ("high-risk case receipt decision"); therefore, those acts [T2/ex] must wait for the "acceptance of high-risk case receipt decision" (T3/ac) fact.

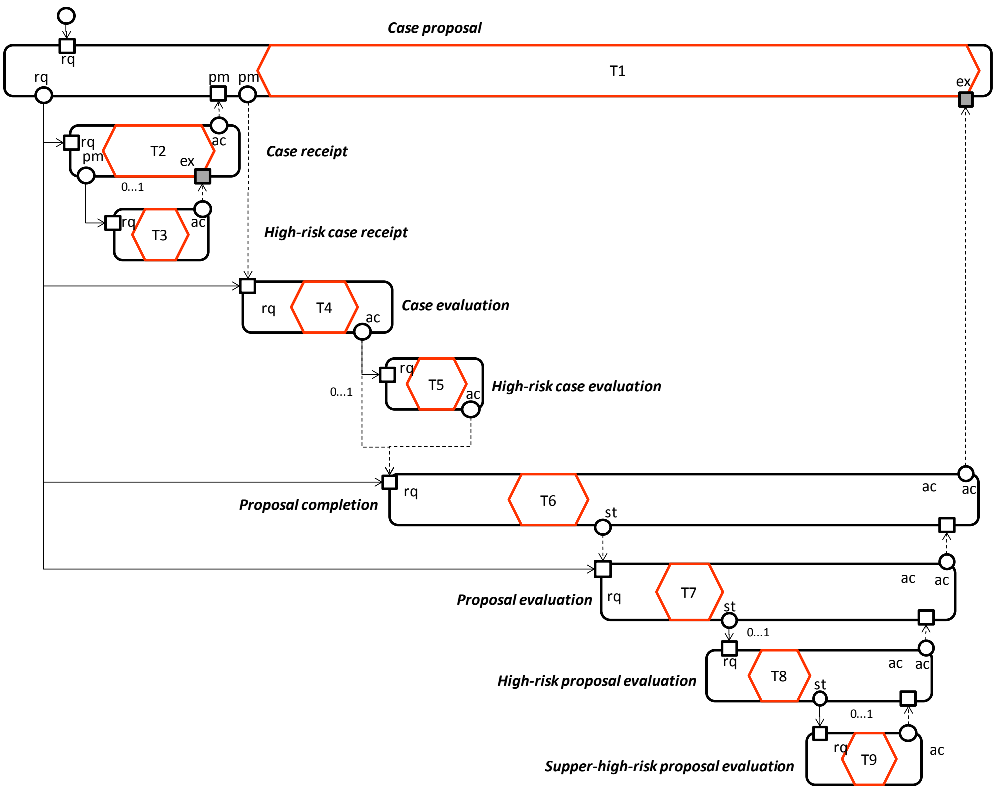


Figure 6-2 PSD of "Company D" proposal and estimation process

[4] [ ] represents act: [T1/pm] means act "to promise an instance of T1".

[5] ( ) represents fact: (T2/ac) means fact "an instance of T2 has been accepted".

- When a "case proposal" is promised (T1/pm), the case must be evaluated. Similar to a case receipt, a case evaluation must also consider high-risk cases that require additional evaluation through T5 ("High-risk case evaluation"). Thus, the act request of T6 [T6/rq] must wait until all case evaluation processes are complete, i.e., either (T4/ac) or (T5/ac).
- T6 ("proposal complementation") requires an evaluation or several evaluations according to the risk level of the proposal; thus, the acceptance of T6 [T6/ac] must wait for one or all facts [T7/ac] (normal-risk proposal), [T8/ac] (high-risk proposal) and [T9/ac] (super-high-risk proposal) according to the risk level of the case proposal.

6.3.3 FM

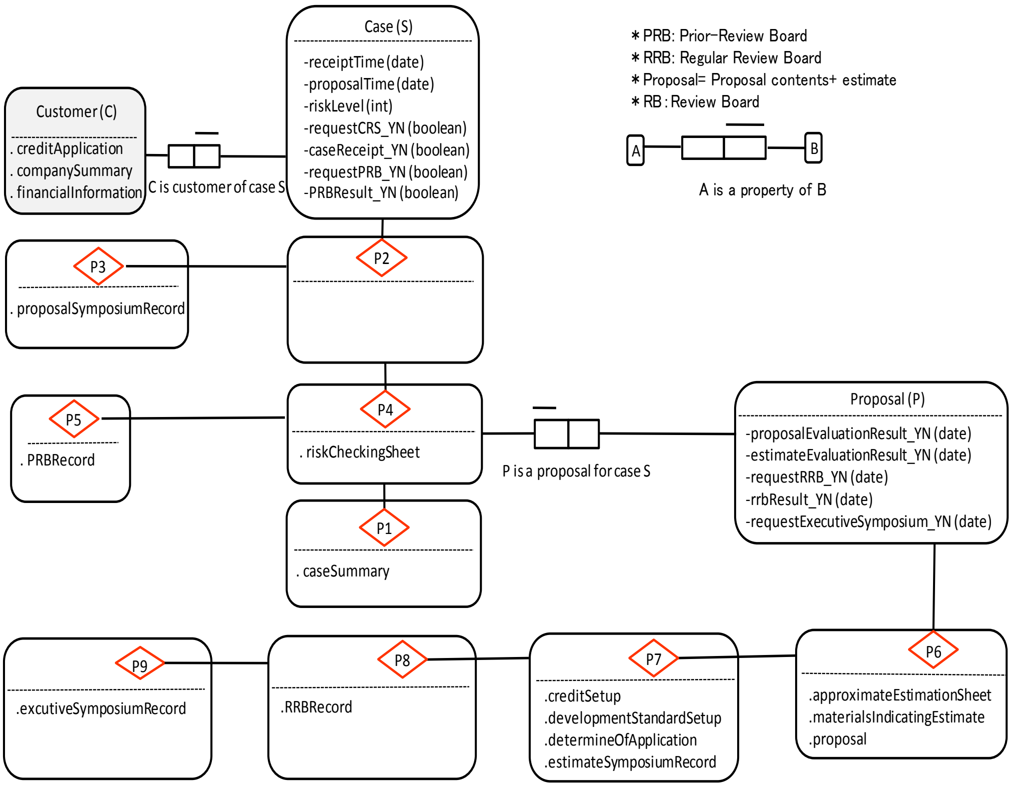


Figure 6-3 OFD of "Company D" proposal and estimation process

Another DEMO model that must be defined is an OFD, which further describes an aspect of an object. In OFD, there are three objects, the Case (S), Customer (C) and Proposal (P), whereby each of which include various defined properties. As shown in Figure 6-3, the Case(s) is (are) related to five products, i.e., P1 - P5 (cf. Table 6-1), and the Proposal (P) is related to four products, i.e., P6, P7, P8 and P9 (cf. Table 6-1).

Documents may be described as a part of a product. For example, a document of the type “Proposal Symposium Record” is produced in P3 (the products of T3). Here, the risk level could be directly assigned or calculated according to items listed in a “risk sheet”.

6.3.4 AM

AM is the most conscious model in DEMO and describes the action rules of each actor role. Each action rule is described as a CRISP (cf. Section 3.2.4) and describes the coordination between the actors that constitute a CRISP network.

As shown in Table 6-2, each CRISP has been rewritten as a table item with three components.

- “WHEN” block: The trigger event and required coordination base (defined in the “FACT” column); for example, the event “rqedT1” = (T1/rq) fits into this block.
- “THEN” block: The reaction (“REACT” column) and, if present, the decision rules for a reaction (“CONDITION” column in “THEN”). For example, deciding whether to promise [T1/rq] or decline an actor role A1 requires checking the condition. Moreover, determining whether Case.caseReceipt_YN (a property of the object “Case”) is true or false is required to make the decision.
- “WITH” block: This block includes “the assignments of the ‘ForThen block’” and “the required production state base for the When block”.

Table 6-2 Action rule table for actor role “A1”

AR:	A1														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
When	rqedT1	For	Case	is done.	If	none	Then	rqT2	For	Case	with	If	none	Then	none
When	stedT2	For	Case	is done.	If	none	Then	acT2	For	Case	with	If	none	Then	none
When	acedT2,rqedT1	For	Case	is done.	If	case.receipt_YN==true	Then	pmT1	For	Case	with	If	none	Then	none
					If	case.receipt_YN==false	Then	dcT1	For	Case	with	If	none	Then	none
When	stedT4	For	Case	is done.	If	none	Then	acT4	For	Case	with	If	none	Then	new proposal
When	acedT4	For	Case	is done.	If	PRB is required	Then	rqT5	For	Case	with	If	none	Then	requestPRB_YN=true
					If	PRB is not required	Then	rqT6	For	Case	with	If	none	Then	requestPRB_YN=false
When	acedT5	For	Case	is done.	If	none	Then	rqT6	For	Case	with	If	none	Then	none
When	stedT6	For	Proposal	is done.	If	none	Then	rqT7	For	Proposal	with	If	none	Then	none
When	stedT7	For	Proposal	is done.	If	proposal.requestRRB_YN==false	Then	acT7	For	Proposal	with	If	none	Then	none
When	acedT8,stedT7	For	Proposal	is done.	If	none	Then	acT7	For	Proposal	with	If	none	Then	none
When	acedT7,stedT6	For	Proposal	is done.	If	none	Then	acT6	For	Proposal	with	If	none	Then	none
When	acedT6,pmedT1	For	Proposal	is done.	If	none	Then	exT1	For	Case	with	If	none	Then	none
							Then	stT1	For	Case	With	If	none	Then	none
When	pmedT1	For	Case	is done.	If	none	Then	rqT4	For	Case	With	If	none	Then	none

The other action rules are listed in the appendix.

6.4 DEMO++-based AnyLogic Model and Simulation

As described in chapter 5, the Main model and the Ontological component of DEMO++ can be directly derived from the DEMO aspect models using a predefined model transformation tool. When the DEMO++ implementation component is added, these two components of DEMO++ can be translated into an AnyLogic model based on the AnyLogic-DEMO++ libraries that the author developed.

6.4.1 Main

The main model is derived from DEMO CM and PM and is used to describe the following aspects.

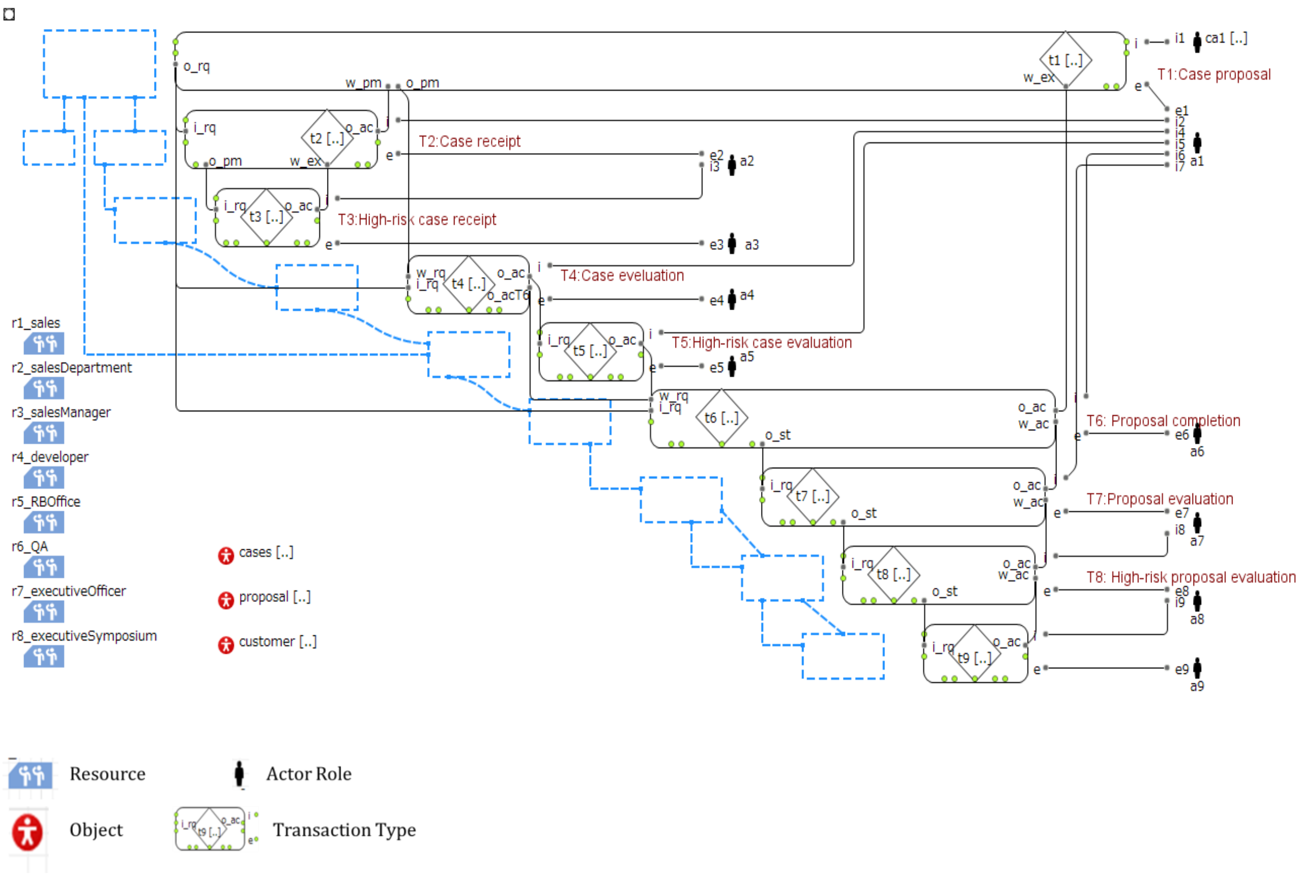


Figure 6-4 Main of the "Company D" proposal and estimation process

- The transaction types (T) and constraints between the transactions (waiting links);
- The actor roles (AR), which are related to the implementation of a transaction type (the links between AR and T);

- The objects (O); and
- The resources (R), which must be manually defined.

The details of all element types in the main model (Figure 6-4) are introduced in the following sections.

6.4.2 Ontological model

In an ontology model, the Transaction type (T), Actor Role (AR) and Object (O) must be defined in detail.

- T is defined using the AnyLogic library element "T", which was described in section 5.4.2. The parameter settings for "T" are automatically derived from the DEMO aspect models, except the "call processor" items, which are derived using the model transformation tools (cf. section 5.6). As shown in Table 6-3, all of the pink columns are automatically filled. The yellow columns are "call processor" items, which must be manually defined to determine if it is necessary to call its processor. The AnyLogic library element "T" is defined according to those parameters, which is shown in Figure 6-5: the left side denotes the standard transaction pattern. The right side shows the required parameters. "Object" is the object used in this transaction; "Ttype" is the transaction type; "description" is the description of the transaction type; "callPmDcProcessor" is used to decide whether this act must call its processor. The other seminar parameters have the seminar function; "defaultPm" is used for deciding whether the default decision is "pm", and the other "default" has the seminar functions. "cBaseConditions" is used to define the "CBase" for each act in the brace. For example, the CBase item for act "ex" is "acedT3". "pBankItems" is used to define pBank; "causalLinkSources" is used to define sources of the causal link related to this transaction, for example, "pmedT2" is the source of a causal link. "waitingLinkSources" is used to define sources of the waiting link related to this transaction; for example, "acedT2" is the source of a waiting link. "Object1" denotes the other objects that may be used in this transaction. "initiator" is the initiator of the transaction type, and "executor" is the executor of the transaction type.

Table 6-3 Parameter setting table for T of DEMO++

ttype	description	obj	obj1	initiator	executor	callRq	callPmDc	callEx	callSt	callAcRj	callQtRq	callExSp	defaultPm	defaultAc
T1	case proposal	case		CA1	A1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T2	case decision	case		A1	A2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T3	high risky case decision	case		A2	A3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T4	case estimation	case		A1	A4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T5	high risky case estimation	case		A1	A5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T6	proposal completion	proposal		A1	A6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T7	proposal evaluation	proposal		A1	A7	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T8	high risky proposal evaluation	proposal		A7	A8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
T9	supper high risky proposal evaluation	proposal		A8	A9	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE

<table>
<tr><th>defaultSp</th><th>defaultQt</th><th>cBaseItems</th><th>pBaseItems</th><th>causalLinkSources</th><th>waitingLinkSource</th></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)|(pmdc)acedT2|(ex)acedT6|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>pmedT1, rqedT1, rqed</td><td>–, pmedT1</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)rqedT1|(pmdc)|(ex)acedT3|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>–, pmedT2</td><td>acedT2</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)pmedT2|(pmdc)|(ex)|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>–</td><td>acedT3</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)rqedT1pmedT1|(pmdc)|(ex)|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>–</td><td>acedT4</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)|(pmdc)|(ex)|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)rqedT1acedT4/acedT5|(pmdc)|(ex)|(st)|(acrj)acedT7|(exsp)|(qtrq)|</td><td>–</td><td>–</td><td>acedT6, stedT6</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)rqedT1stedT6|(pmdc)|(ex)|(st)|(acrj)acedT8|(exsp)|(qtrq)|</td><td>–</td><td>stedT7</td><td>acedT7</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)stedT7|(pmdc)|(ex)|(st)|(acrj)acedT9|(exsp)|(qtrq)|</td><td>–</td><td>stedT8</td><td>acedT8</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>(rq)stedT8|(pmdc)|(ex)|(st)|(acrj)|(exsp)|(qtrq)|</td><td>–</td><td>–</td><td>acedT9</td></tr>
</table>

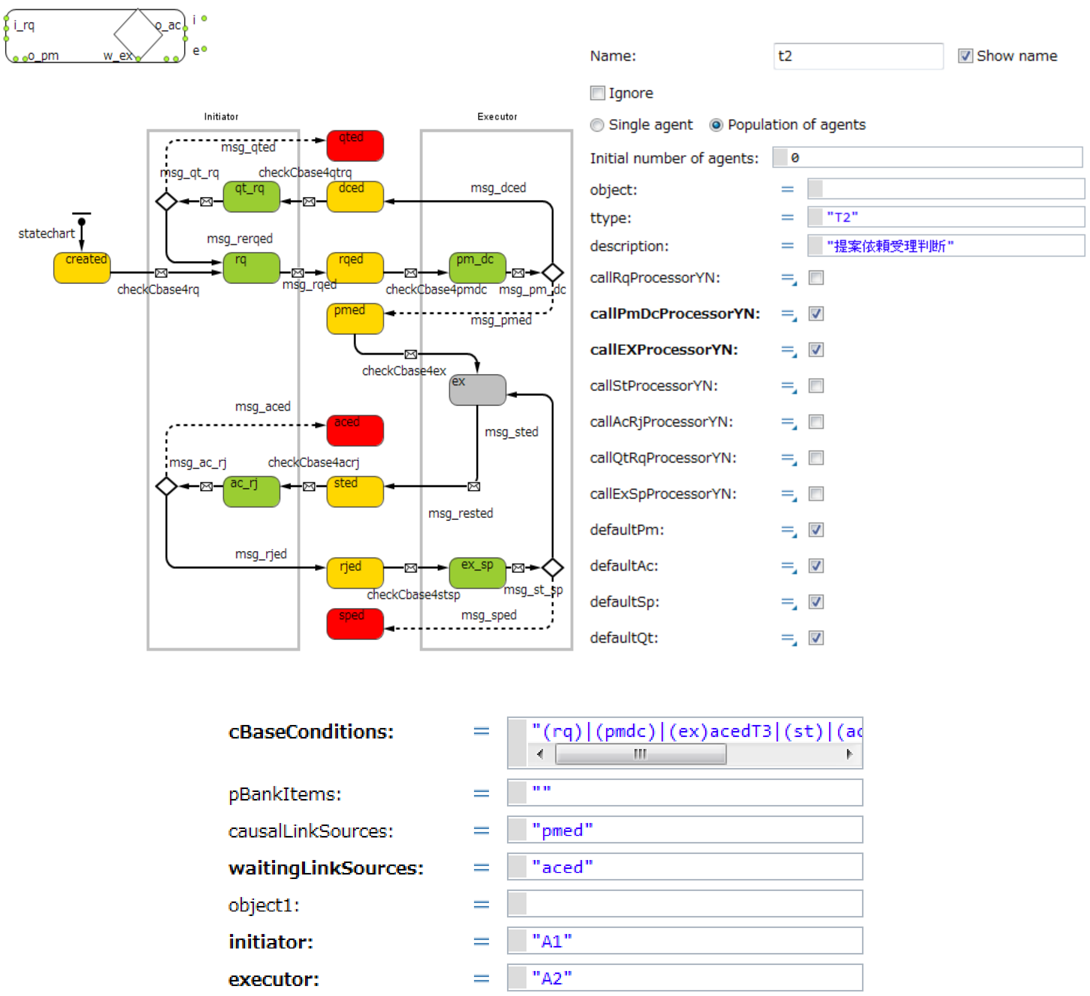


Figure 6-5 T in DEMO++ and AnyLogic

- An object model (O) is derived from an FM model of DEMO, which is shown in

Figure 6-6. There are three objects, “Customer, “Case” and “Proposal”, with corresponding properties (see section 6.2.3). The definition follows O of DEMO++, as described in section 5.4.2.1. The properties “id”, “name”, “type”, “category”, “serviceDelayTime”, and “serviceWhoDid” are default properties for all of the defined objects. The remaining properties of “Case” are derived from DEMO FM “Case”; the properties of “Proposal” are derived from DEMO FM “Proposal”.

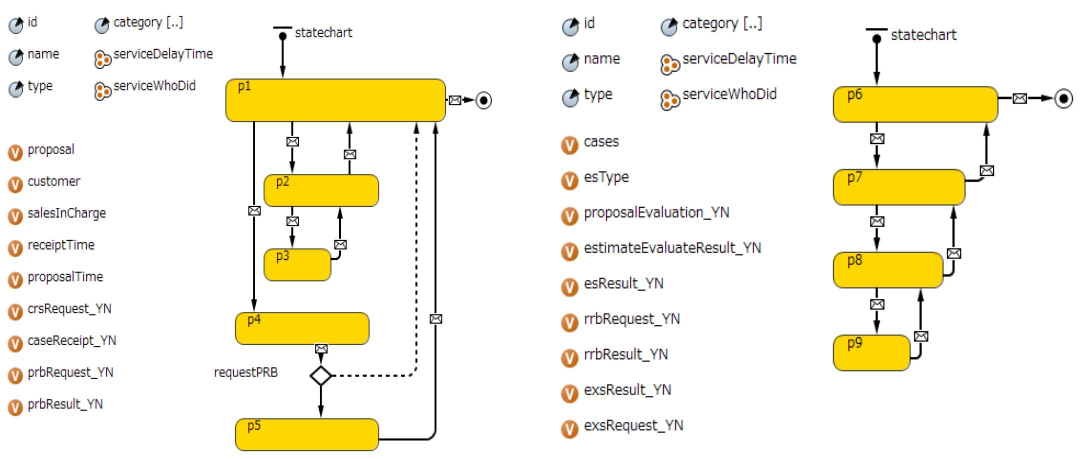


Figure 6-6 Objects of the “Company D” proposal and estimation process

6.4.3 Implementation model

For an ontological model, the implementation details, including the resources and the actor role components, must also be defined.

6.4.3.1 Resource Model

- This project defines the actors as active resource types who play the actor roles, which are shown in

Table 6-4. The following aspects are described in the table.

First, the table lists the following eight types of actors according to the organizational structure of “Company D”:

- r1, sales, who perform the works of case reception and evaluation;
- r2, sales team leader, the senior member that responds for review, management

and quality control;

- r3, sales project manager, who responds to project requests;
- r4, R&D staff, who provide technical support in case evaluation and proposal;
- r5, RB officer, who manages RB-related work;
- r6, staff in the quality assurance department, who control the risks;
- r7, executive officer, the executive managers who have more management and decision authority;
- r8, executive symposium officers, who assist executive managers in organizing the symposium.

Table 6-4 Actor-actor role-function mapping table (Case D)

Actors								Actor Roles		Functions			
Sales Dep.			R&D Dep.	RB	QA Dep.	CEO	ES Office						
R1 Sales	R2 Sales TL	R3 Sales PM	R4 R&D department	R5 RB office	R6 Quality assurance department	R7 Executive officer	R8 Executive symposium office			F1	F2	F3	F4
T1.E								A1	case responser				
T2.O													
T4.O,T4.R													
T5.O (info)		T5.O											
T6.O													
T7.O													
		T2.E						A2	case acceptance responser				
T3.O (info)		T3.O											
	T3.E	T3.E	T3.E					A3	high-risk-case acceptance responser				
		T4.E						A4	proposal risk estimator				
	T5.E	T5.E	T5.E	T5.E (info)	T5.E			A5	high-risk-proposal risk estimator				
T6.E								A6	proposal completer				
	T7.E							A7	proposal evaluator				
T8.O (info)													
	T8.E	T8.E	T8.E	T8.E (info)	T8.E			A8	high-risk-proposal evaluator				
		T9.O (info)			T9.O								
						T9.E	T9.E (info)	A9	supper-high-risk-proposal evaluator				

- Second, all these actors are mapped into actor roles in DEMO, as shown in

Table 6-4. For example, actor “r1” (salesperson) takes on the actor role, meaning that this type of staff needs to take on these responsibilities.

- A1 in the execution phase (E) of T1; the Order phase (O) of T2; the Order phase (O) of T4; the Result phase (R) of T4; the order phase of T5, which is only for information-level business (O-info); the order phase (O) of T6; and the Order phase of T7;

- A2 in the order phase of T3, which is only for information-level business (O-info for short);
- A6 in the execution phase (E) of T6; and
- A7 in the Order phase of T8, which is only for information-level business (O-info).

Third, the relationships between the actor roles and the functions that they support are defined. For example, the actor roles A1, A2 and A3 are utilized for the function F1 (case reception); A4 and A5 are for the function F2 (case risk evaluation); the actor roles A6 are for the function F3 (proposal and estimation); and A7-A9 are defined for the function F4 (proposal evaluation).

Using an actor-actor role-function mapping table, we may understand the "who coordinates" and "coordinates for what" questions.

6.4.3.2 Actor Role Components

Implementation is related to the "how to cooperate" question, which is defined by the execution steps that are associated with the acts within the corresponding actor roles. To connect an ontological act with the implementation details, it is necessary to consider whether this act must be expanded into a series of implementation details in its processor. A processor is the execution unit of an act. Typically, under the following three conditions, an ontological act is expanded into the implementation details in the processor; the other situations are ignored because they are beyond the scope of the current work.

- The act is related to resource utilization activities that must be analyzed.
- The act contains complex info-logical or data-logical steps that are within the scope of our research.
- The act contains alternative implementation plans according to different situations that are within the scope of our research.

In the case of "Company D", the acts that must be implemented in the processors are defined in Table 6-5. For example, T1pm/dc is an act that must be implemented in a processor. The processor is defined in actor role A1.

Table 6-5 Processor table

Actor Role	Transaction type	Act	Need processor?	Actor Role	Transaction type	Act	Need processor?
CA1		T1rq		A1		T6rq	
A1		T1pm/dc	TRUE	A6		T6pm/dc	
A1		T1 ex,st		A6		T6 ex,st	TRUE
CA1	T1	T1ac/rj		A1	T6	T6ac/rj	TRUE
A1		T2rq		A1		T7rq	
A2		T2pm/dc	TRUE	A7		T7pm/dc	
A2		T2 ex,st	TRUE	A7		T7 ex,st	TRUE
A1	T2	T2ac/rj		A1	T7	T7ac/rj	
A2		T3rq	TRUE	A7		T8rq	TRUE
A3		T3pm/dc		A8		T8pm/dc	TRUE
A3		T3 ex,st	TRUE	A8		T8 ex,st	
A2	T3	T3ac/rj		A7	T8	T8ac/rj	
A1		T4rq	TRUE	A8		T9rq	TRUE
A4		T4pm/dc		A9		T9 pm/dc	TRUE
A4		T4 ex,st	TRUE	A9		T9 ex,st	
A1	T4	T4ac/rj		A8	T9	T9ac/rj	
A1		T5rq	TRUE				
A5		T5pm/dc	TRUE				
A5		T5 ex,st					
A1	T5	T5ac/rj					

The details of each actor role are defined according to real-world processes. Actor role A1 is shown as an example in Figure 6-7. A1 is the actor role of the "case responder" who is the executor of T1; they respond to [T1/pm], [T1/ex], [T1/dc], and [T1/st] and are the initiator of T2, T4, T5, T6 and T7, which responds to rq and ac/rj for each transaction type. As described in Figure 6-7, the implementation details for the acts, which must be executed in the processor, are described whereby the input defines the required act (the beginning of an act), and the output defines the status after proceeding with the act (the state after act). For example, act [T1/pm] is expanded in its processor A1 between input_pmdcT1 and output_pmdcedT1. The detailed execution process includes the business-level action "b_caseReceipt" and the decision-making act "CaseReceipt_YN", which determines whether to promise or decline a request according to the action rule defined in the AM.

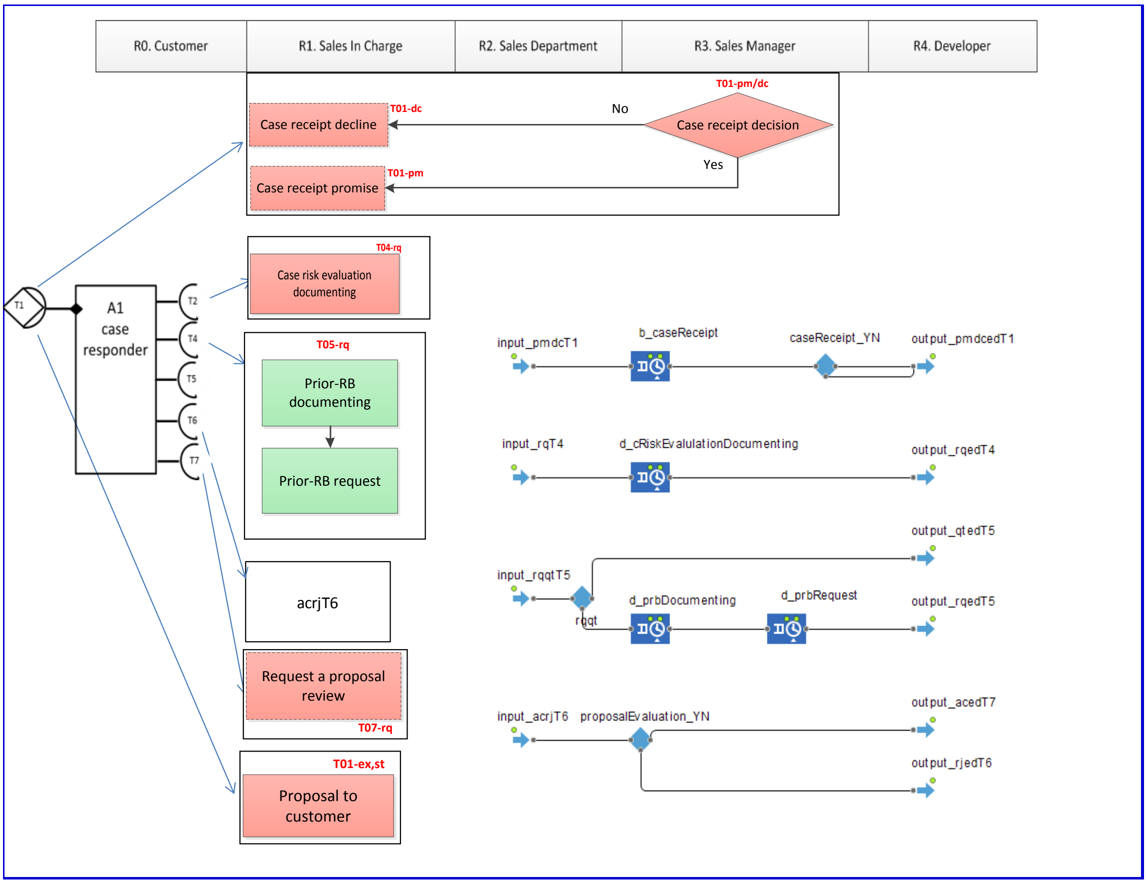


Figure 6-7 Implementation details of the actor role “A1”

The implementations of the other acts are listed in the appendix.

6.5 Simulation

6.5.1 Data collection and setup

The data was collected by (1) interviewing related staff, (2) checking documents and records and (3) assigning according to common characteristics.

6.5.1.1 Incoming case rate

The incoming case rate was determined for three months (from July 1, 2014 to September 31, 2014), covering approximately 63 days. A total of 416 cases were investigated. A total of 2 cases were indicative of risk level S (the highest level risk), 17 are at risk level A (a high risk level), 24 are at risk level B (a normal risk level), and 373 are at risk level C (a low risk level). Based on the statistical results, the author established an incoming case rate of 0.85 (pieces/hour) using a triangular distribution. The percentage of each risk level is as follows:

S: A: B: C=0.1%:5.7%:4%:90.2%

The simulation period is set as four months (from June 1, 2014 to September 31, 2014), with one month as a warm-up period.

6.5.1.2 Case handling time

In this study, the case-handling time is strongly related to the case risk level and the skill level of the actors that handled the case.

Execution time = standard time*risk level/skill level

- The rate of risk level is given as S:A:B:C= 3: 1.5 : 1.2: 1, e.g., S-level cases usually require three-times as much time as do C-level cases.
- The rate of skill level is given as A: B: C= 1.2 : 1.0 : 1.5. ‘A’ is the skilled level, ‘B’ is normal level, and ‘C’ is the primary level. The distribution for various staff members in the sales group of “Company D” is shown in Figure 6-8. For example, if the standard time is 10 min, the risk level of the case is ‘A’, and the resource that responds to the case has a skill level of ‘H’, then the staff member will require 12.5 min to complete the case.

Execution time = 10*1.5/1.2 = 12.5.

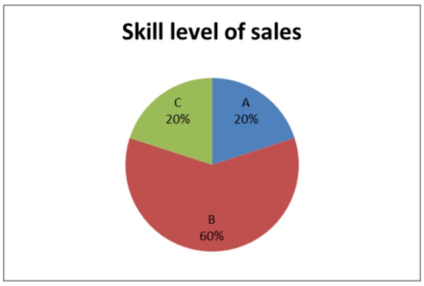


Figure 6-8 Skill level distribution

The standard time (units of hours) for each execution step and the decision-making possibilities are listed in Table 6-6. These data were obtained from reviewing existing records and interviewing staffs. The items beginning with “duration_i” indicate the info-logical time, “duration_b” indicates the ontological time, and “duration_d” indicates the data-logical time; the items beginning with “probability” indicate the possibility of choice.

Table 6-6 Simulation setup table

rateCaseComing:	=	sim_rateCaseComing
probabilityRiskLevelS:	=	(double)2/416
probabilityRiskLevelA:	=	(double)17/416
probabilityRiskLevelB:	=	(double)24/416
probabilityCustomerAcProposal:	=	0.5
probabilityReceipt:	=	0.8
duration_b_caseReceipt:	=	3
probabilityCrsReceipt:	=	0.8
duration_d_crsDocumenting:	=	triangular(2,3)
duration_i_crsPrepare:	=	triangular(4, 10)
duration_b_crs:	=	triangular(1, 2)
duration_d_crsReport:	=	0
duration_d_cRiskEvaluationDocumenting:	=	triangular(1, 2)
duration_b_cRiskEvaluation:	=	0
probabilityRequestPrb:	=	0.1
duration_d_prbDocumenting:	=	0
duration_d_prbPrepare:	=	triangular(4, 10)
duration_b_prb:	=	triangular(1, 2)
duration_d_prbReport:	=	triangular(1, 2)
duration_b_proposal:	=	triangular(32,40,64)
duration_d_esDocumenting:	=	triangular(1, 8)
duration_d_esPrepare:	=	triangular(7, 14)
duration_b_es:	=	triangular(0.5, 1)
duration_d_esReport:	=	0
duration_b_proposalEvaluation:	=	0
duration_d_rrbPrepare:	=	triangular(7, 36)
duration_b_rrb:	=	triangular(0.5, 1)
duration_d_rrbReport:	=	0
duration_d_rrbDocumenting:	=	triangular(1, 8)
duration_d_prbRequest:	=	0
duration_d_exsDocumenting:	=	triangular(1, 8)
duration_d_exsPrepare:	=	triangular(20, 36)
duration_b_exs:	=	triangular(1, 2)
duration_d_exsReport:	=	0
duration_i_exsReceipt:	=	0
probabilityProposalEvaluation:	=	1
probabilityExsModify:	=	0.1
probabilityExsNg:	=	0.1
probabilityRrbNg:	=	0.01
probabilityRrbGo:	=	0.96
probabilityRrbModify:	=	0.03
duration_b_rrbRequest:	=	0
duration_i_rrbReceipt:	=	0
probabilityRequestExs:	=	0.3
duration_d_caseDocumenting:	=	0
duration_b_CaseEvaluation:	=	triangular(8, 16)
probabilityCrsRequest:	=	0.8
probabilityPrbModify:	=	0.4
probabilityPrbNg:	=	0.2
probabilityPrbGo:	=	0.4
probabilityEsEvaluation:	=	0.6
probabilityRrbRequest:	=	0.3
probabilityExsGo:	=	0.8

6.5.1.3 Actor (active resources)

As shown in Table 6-7, the capacity of active resources is given. For example, there are 25 salespersons that are denoted by “resource type 1” (r1). For all types of actors, at most 60% of their working time can be spent performing “proposal and estimation”-related work. This is modeled by a uniform distribution.

People in sales are skilled in certain aspects but not in all aspects; for example, one salesperson may be an expert in “bank”-related areas, whereas another salesperson may not be familiar with this area but has good connections within the government. The case should be assigned to an appropriate staff member according to their field and capability. Thus, this project establishes the skill level in three different dimensions (government, finance and health care) for resource ‘r1’ (sales person). The dimensions are mapped to areas that a case belongs to in order to map different types of sales people to different type of cases.

Table 6-7 Resource capacity table

Resource		
	Name	capacity
r1	sales person	25
r2	sales TL	16
r3	sales PM	16
r4	R&D staff	10
r5	RB officer	10
r6	Q&A	3
r7	executive officer	6
r8	executive symposium	2

6.5.2 Simulation results, analysis, and suggestions

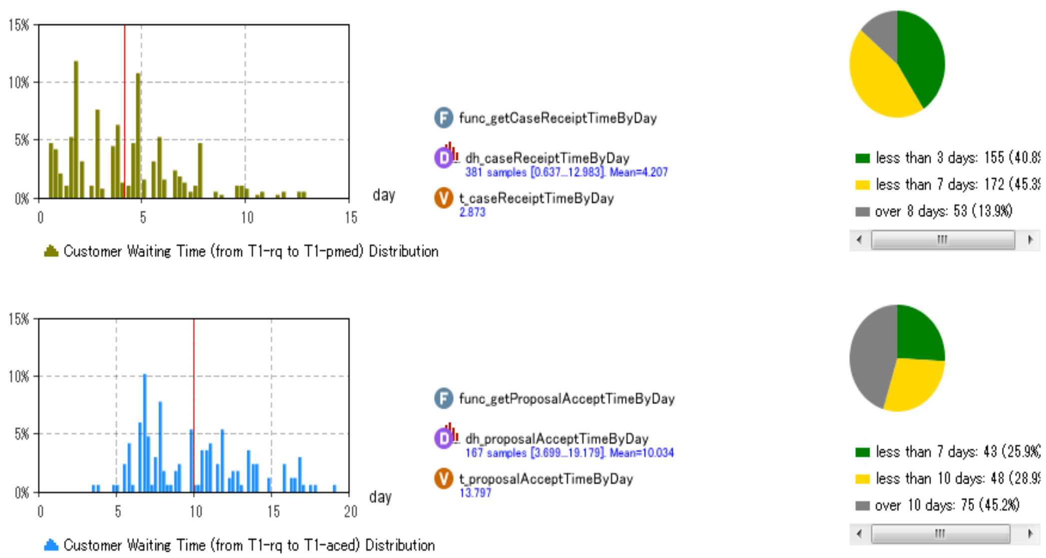


Figure 6-9 Simulation result statistics

As shown in Figure 6-9 and following the settings discussed above, we obtain average case reception times. The upper histogram describes customer waiting time until feedback is received from company D about whether they will accept the case for a proposal or not. The upper pie chart shows that 40% of the cases require less than three days; 45% require less than seven days; and 13.9% of the cases require more than eight days. The lower histogram describes customer waiting time until a proposal is received. As shown in the lower pie chart, most cases require a proposal time of longer than 10 days.

As shown in the top-left frame of Figure 6-10 (1) utilization of resources, the bottleneck is the sales department; approximately 96.6% of salespeople are occupied most of the time. Therefore, cases and proposals must wait for this resource to be released. From (2) total time spent in business level, information level and data level (B-I-D-level), we can note that only 39% of a salesperson's time is used in B-level business; the other 61% of a salesperson's time is used for information-level work and data transfer. According to "(3) Total time spent taking on different actor roles", the times r1 spent taking on different actor roles are as follows: 13% in A1, 43% in A2, 46.7% in A3, 27.83 in A6 and 19.2% in A7. As shown in "(4) delay time of each transaction", T2 and T3 are easy to delay because "r1" spends most of their time taking on these two actor roles.

To conclude, the problem of delay is mainly caused by the high utilization rate of a salesperson. However, the causes of the high utilization can be analyzed from two perspectives: (1) analyzing the resources for coordination and (2) analyzing the coordination.

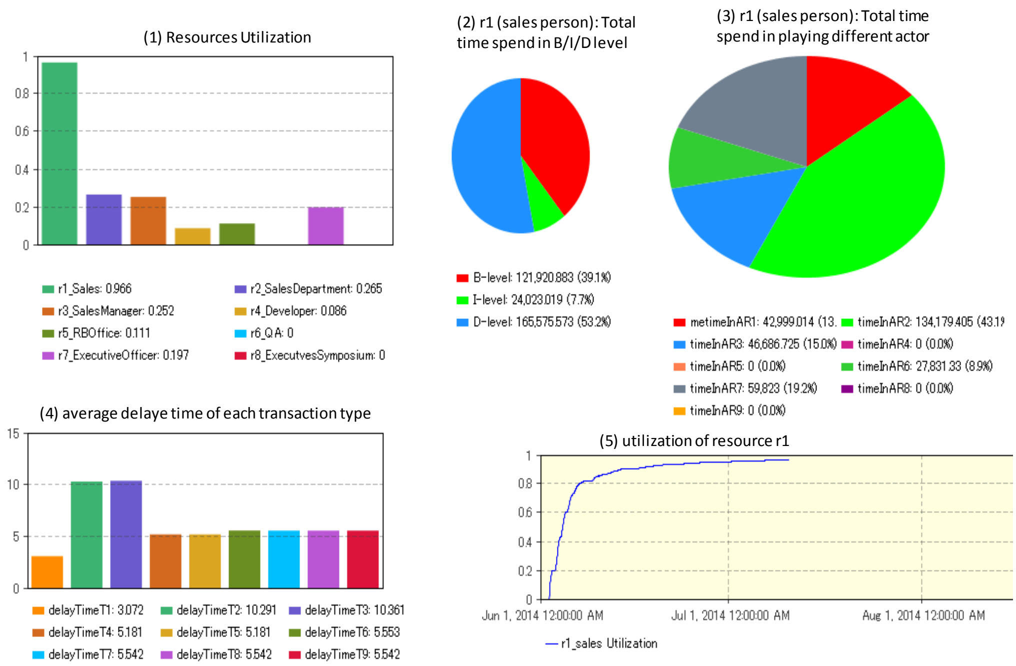


Figure 6-10 Simulation results

(1) Analyzing the resources for coordination:

- **(S1) Resource allocation:** The most basic causes of the bottleneck problem may be a result of the resource allocation problem. Such problems can be solved through an optimization analysis. For example, in the case of "Company D", through an optimization analysis, the author observed that based on the original incoming rate and skill level, the enterprise does need not 25 salespeople; instead, the use of 40 salespeople would improve the proposal time and allow most cases to be completed within 10 days.
- **(S2) Effects of resource properties:** As an active resource, the personality, characteristics, skill level, knowledge structure, and social networks of an actor influence the effectiveness and efficiency of taking on an actor role. For example, in the case of "Company D", improving the skill levels of the salespeople reduces the time required for each case (related to type 2.4); the current percentage of A-level and C-level salespeople is 20%. If some of the C-level salespeople can improve their skill level to the B-level and if some of the B-level salespeople can improve their skill level to the A-level, only 8% of the salespeople will be C-level salespeople, and 34% will be A-level salespeople. Thus, these human resources reduce the r1 occupation time by 1%-2 %.

(2) Analyzing the coordination: From the simulation results, we can see that the coordination is not sufficiently efficient; therefore, the sales people are taking on too many information-level and data-level tasks. In addition, they are simultaneously taking on several actor roles.

- **(S3) Who coordinates:** Add assistant staff to help perform the info-logical and data-logical tasks so that a time-constrained salesperson can concentrate on their value-added B-level work (which is related to type 2.2). The simulation suggests that the enterprise should assign another 10 assistant staff members to perform the info-logical and data-logical tasks, which will have the same effect on the result as solution S2.
- **(S4) How to coordinate:** Change the process to reduce the time necessary for information transfer and documentation. Based on our simulation, for example, if the data-level work can be reduced by 20%, most of the cases

can be proposed within 10 days without adding resources or within only 5 days even if more salespeople are added.

The solutions are compared and listed in Table 6-8.

Table 6-8 Change type validation table for “Case D”

Cases	Ontology				Implementation				
	Type 1.1 CM	Type 1.2 PM	Type 1.3 FM	Type 1.4 AM	Type 2.1 How to coordinate	Type 2.2 Who coordinate	Type 2.3 Passive Resource	Type 2.4 Active Resource	Type 2.5 Others
Case D					S4	S3		S1,S2	

6.5.3 Feedback

A solution was proposed to Company D. Their feedback regarding the results and the methodology is summarized as follows.

- Concerning the results: The simulation results and the bottlenecks that were found are a part of the real-world situation. This enterprise does not have sufficient skilled salespersons for the amount of work that must be completed. They are considering adding more workers, especially more skilled salespeople to remedy the situation. From this perspective, the results can help in the decision-making process. Because of the large amount of information and data-related work, they will consider hiring both salespeople and assistants to reduce the workload and the budget. Concerning solution S4, they consider the solution to be a good idea to increase efficiency; however, they still must investigate how to properly apply this solution. One of the biggest concerns is related to the confidence of the data that they provided. It is very difficult for the company to determine the time requirements or measurements according to DEMO concepts. Some of the data are not empirical and are simply theoretical in nature. Another concern is the B\I\D-level time measurements. The feedback indicates that the percentage should be holistically compared to determine if this distribution is normal. This is related to the Japanese working style—Japanese workers prefer to spend more time on documenting tasks compared to workers in other countries.
- Concerning the method: “Company D” shows great interest in this method. They agree that it is different from traditional simulations, especially in the manner

whereby it analyzes different levels of time requirements. The method appears to be more useful than other current methods and has the potential to analyze more aspects of enterprises compared to traditional simulations. However, more research and case studies are required to investigate how to implement the results. Meanwhile, there are still many components that must be improved if this method is to be commercialized, e.g., the development interface, the model transformation tool and animations.

Moreover, the company kindly provided various suggestions and their expectations for the simulation.

- It would be more helpful if the simulation could assist in investigating the relationships that exist among the key factors to determine the most criticized factor or to show more relationships between the key factors to provide additional guidance.
- The animation must be improved to show the state of the bottleneck transaction.
- The B-I-D-level analysis is very interesting; research should continue in this direction to determine if there are further results that this method can provide.

The author also gathered feedback from developers who are using this method for DEMO-based simulation. Some of their comments are provided below:

"From DEMO's perspective, this method is very useful and represents a breakthrough in the way that DEMO, which is originally static, can be transformed into a dynamic simulation model. There are several approaches for the same objective, such as transformation to another tool like Petri nets or other own simulation codes. The simulation models obtained using conventional tools are deteriorated compared to the original DEMO models with respect to the essential ideology/framework of DEMO. On the other hand, this method can transform a DEMO model into an AnyLogic model without losing almost anything important and can reconstruct the DEMO model in AnyLogic impeccably. Thus, it allows me to develop and elaborate the DEMO model further in AnyLogic even after migrating from 'purely DEMO' to 'DEMO in AnyLogic'.

From AnyLogic's perspective, this method introduces a framework or a way of thinking into AnyLogic. Because AnyLogic offers multi-method modelling, it is overfree for preparing a simulation model. Without this method, the excellent freedom of AnyLogic may cause two problems. First, it is very difficult to construct a large and/or complex model in AnyLogic from scratch. It would be a long climb. This method automatically generates the blank model from a DEMO model, and I can add details onto the blank model. The DEMO model reduces the workload dramatically. Second, its excellent freedom allows some inconsistency in the model with respect to granularity and implementation. This method allows the framework of DEMO to be incorporated in AnyLogic. By retaining the mindset of DEMO, I can develop my simulation models with less inconsistency.

Among the limitations and potential contributions of Ms. Liu's method, the DEMO model behaves very dynamically in AnyLogic. It is a great tool. However, although the original model is DEMO, the simulation is somewhat removed from DEMO. During the development process in AnyLogic, I have to restore things that were thrown away under the modeling procedures in DEMO. Actually, the things are mostly implemented (in contrast to ontology). Thus it would be great if her method had something to guide me in solving this contradiction."

--from Suga, Tetsuya, who is familiar with DEMO-related concepts and JAVA programming

"It provides a systematic way to transform DEMO to Simulation and provides a flexible environment for users.

However, the user may need a background in Java programming for easy usage, and a detailed description and tutorials for application should be provided. "

--from Zakaria, Yahia, who is not familiar with DEMO and JAVA programming.

6.6 Discussion

In this project, the proposed methodology was evaluated to investigate its ability to assist business process change. The DEMO models allowed the proposal and estimation process of "Company D" to be clarified from the construction, process, objects and rules aspects. Compared to traditional workflow perspectives, the company obtained a better understanding of their business process. Furthermore, based on the DEMO model, DEMO++ models can be derived by applying transformation tools. The corresponding simulation can be developed using DEMO++ libraries, thus avoiding development from scratch. This methodology combines high-level analysis with implementation-level measurements and evaluation. It provides all of the functions that traditional process-based simulation can provide and additional advantages. By conducting this project and interviewing the users and developers, the author observed that those experienced in the "DEMO mode" of thinking accept and use this methodology quite easily. However, for those who are not familiar with "DEMO", the most difficult part is understanding basic concepts. A summary of the evaluation is provided below.

- DEMO is a good method for describing the essence of an enterprise, however difficult to be grasped; The DEMO++-based simulation provides a different perspective for creating simulation and analyzing business processes that are consistent with the DEMO methodology;
- This method has a higher learning requirement for developers because they need to understand both DEMO theory and simulation development;
- Compared with a workflow-based method, this method is more structured. However, this methodology has an application area that is not always the best choice for all types of simulations;
- Several practical advantages exist, such as the statistics concerning the B-I-D-level and the simulation of the effects of properties of active resources. However, the potential of this method should be further investigated.
- More assistance in creating implementation models is required, as well as more instruction documents.

In addition, as an outcome of this project, the process of developing DEMO++-based simulations was standardized and demonstrated. The standard process consists of the following steps:

- Create DEMO aspects models, including CM, PM, FM and AM;
- Obtain implementation-related information, including the Actor-Actor Role-Function mapping table, processor table, and actor role comment models for the

acts that required the processor;

- Obtain simulation-related information, for example, coming rate and execution time.
- Create the simulation model.

Finally, as a result of the project, several solutions were proposed based on the DEMO analysis and DEMO++-based simulation results. These solutions provide some interesting viewpoints for analyzing business process changes. More important, the types of solutions and the analysis method can be used as guidelines for other DEMO++-based simulation projects.

The project also exposed limitations that must be improved in the future research:

- The implementation model: AM defines the rules for making decisions. However, AM is difficult for both developers and end users to define. Although we used an AM table instead of the original format to reduce complexity, defining AM remains time consuming. In addition, AM must be manually mapped into implementation models as execution steps and decision points. This process is still the most complex part of simulation development. We need to provide a solution to enable semi-automatic map processing in future research.
- The difficulties of obtaining data: DEMO analyzes an enterprise from different perspectives within a high-level abstraction. The DEMO++-based simulation can provide different levels of analysis. The objectives of the simulation and whether the B-I-D level is required must be decided in advance. In developing the simulation for case D, the author observed that it was easier to obtain ontological level data, for example, how long will a transaction type will take, than to obtain detailed data for execution steps, particularly for B-I-D level steps. In some cases, it is not necessary to define B-I-D level steps separately. In other cases, if the DEMO++ concept is desirable to obtain interesting findings, such as separating ontological works from info-logical works, e.g., documenting, and data-logical works, e.g. information transforming or information transformation, the developer must make more effort in advance to explain and confirm with the end users about the correctness of the concept.
- The simulation and animation need to be improved: The current interface is designed for automatic model transformation. Different animation interfaces can

be provided for different simulation objects. Moreover, the current simulation is still weak in analyzing enterprise changes and providing solutions, for example, the changes that caused by organization structure changes. Future research must consider how to make use of the current structure to provide more perspectives for analyzing changes.

- Only the standard transaction pattern is examined: In certain real-world cases, exceptions such as cancel or redo must be simulated. The standard transaction pattern must then be expanded to describe the complete transaction pattern (with decline, reject and cancellation) to make the simulations more realistic and comprehensive.

CHAPTER 7 Conclusions and Future Research

7.1 Summary of Main Contributions

By taking an enterprise engineering perspective that focused on coordination and construction, the author proposed a methodology for analyzing business process change. The contributions of this dissertation can be summarized as follows:

- Types that can be used to analyze business process changes were described; the study explained how one should consider changes in the context of enterprise engineering.
- A simulation methodology that includes a conceptual modeling method and useful libraries that can be used to analyze business process changes by calculations, comparisons, and evaluations.

Chapter 2 reviewed related research projects by considering the different perspectives and methods that were used; this review also explained why current methodologies always fail when supporting business process changes. What is enterprise engineering and what are the advantages of this perspective? What problems exist in current business process simulation research?

In chapter 3, the meta-model of DEMO was defined.

Chapter 4 answered our first research question, i.e., how should we analyze business process changes in the context of enterprise engineering? The two types (with nine sub-types) of changes in the defined construction levels are guidelines that can be used throughout the research project.

Chapter 5 proposed a methodology for applying an enterprise ontology in a simulation environment to support enterprise change, particularly for business process re-design and reengineering. The proposed DEMO++, which is an expanded version of the enterprise ontology DEMO with an implementation model, can be used as a conceptual modeling method for business process simulation. Two components are included in this framework: ontology and implementation. The former component describes how the enterprise is constructed using ontological models, and the latter component describes how the relevant aspect of an ontological model is implemented. This framework constitutes a method of analyzing, executing and evaluating business processes and reduces the complexity of modeling and simulating large systems yet focuses on implementing the relevant part of the process. DEMO++ is fully modularized; therefore, the generated simulation is entirely based on individual components, with features that enable controllable changes and reusability. The framework supports business process simulations, especially when the simulation is used in top-down enterprise reengineering.

This DEMO++-based simulation methodology was employed in real-world case studies in chapters 5 and 6.

- Case C (Pizza Case in chapter 5). In reality, a pizza enterprise primarily concerns resource allocation issues. These are simple and traditional problems that can also be easily solved using process-based simulation models. This case was only used to explain the basic concepts of the DEMO++ framework. In the context of enterprise engineering, DEMO++ can provide more capabilities compared to simulations alone. This method is more capable of analyzing and simulating complex business processes and cooperation compared to other traditional simulation models.
- Case D (Proposal and estimation case in chapter 6). Case D represents an application that is more related to an active resource, including knowledge, skill level and information delay. This case was used to evaluate the methodology, standardize the development process for further projects and propose improvement solutions for “Company D”.

Traditional business process simulations are inadequate for describing complex systems; it is difficult to change models to simulate reengineering, and the reengineering process cannot be repeated because of a lack of systematic approaches. These limitations are caused by the workflow perspective, by the inadequate abstraction levels that are used in conceptual modeling, and by indistinguishable opinions about design and implementation. Compared with traditional, workflow-based business process modeling and simulation methods, enterprise engineering, which is the basis of DEMO++, is not only a simulation method but also an analysis method. This method can be well integrated with management to analyze problems, find solutions and evaluate alternative plans for enterprise reengineering. In the context of enterprise engineering, DEMO++ can provide more capabilities than simply simulations. This method is better capable of analyzing and simulating complex business processes and cooperation, which other traditional simulation models cannot adequately support. The model is a modularized, component-based simulation model with increased reusability, changeability and flexibility.

Another contribution of this research is practical in nature. By clarifying the differences and dependencies between the ontological and implementation models, our methodology includes a generic framework for generating a modularized, component-based simulation model with increased reusability. DEMO++ is not only a method for simulation but also a method for analysis. This method can be well integrated with management to analyze problems, seek solutions and evaluate alternative plans for enterprise reengineering. The proposed components were developed as an AnyLogic DEMO++ library, which can be reused in other DEMO++-based simulations.

7.2 Limitations and Future Research

One limitation of this research is that the author used the standard transaction pattern to describe transactions. However, in certain real-world cases, exceptions such as cancel or redo must be simulated. The standard transaction pattern must then be expanded to describe the complete transaction pattern (with decline, reject and cancellation) to make the simulations more realistic and comprehensive.

Another limitation is the difficulty in developing an implementation model. Although the DEMO++ AnyLogic components are useable, we still must manually create the simulation models, with many repeated modification tasks. Based on the meta-model provided in this research and according to previous studies, (Liu & Iijima, 2014) converted a semi-automatic transformation tool from DEMO to DEMO++ using an AnyLogic table; however, a substantial amount of manual work is required to convert DEMO++ into an executable AnyLogic model. Our next research project will focus on reducing the complexity of the change simulation model based on DEMO++.

In addition, the simulation and animation need to be improved. The current simulation is still weak in analyzing enterprise changes and providing solutions, such as the changes that caused by organizational structure changes. Our next research objective will improve this method by combining an agent base and system dynamics with the current simulation framework to create a broader range of possibilities for business process change analysis. The following problems must be solved:

- How should we define an agent-based model using the current structure?
- How should we define the attributes of an active resource to evaluate the effects of possible changes?
- How should we simulate an organizational structure change and the effect of this change (different actor-actor role mapping)?

Enterprise-engineering-based business process simulations and business process change analysis is a new but promising research area. Considering the many substantial challenges that remain in this field, this research is just a beginning.

References

Ackoff, R. L., Gupta, S. K., & Minas, J. S. (1963). Scientific method: Optimizing applied research decisions. *Administrative Science Quarterly*, *7*(4), 551–553.

Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, *90*(2), 129–149.

Arena Simulation. (n.d.). Retrieved September 23, 2013, from https://www.arenasimulation.com/

Argyris, C., & Schon, D. A. (1987). *Organizational Learning: A Theory of Action Perspective*. Reading, MA: Addison Wesley.

Axelrod, R. (1997). Advancing the art of simulation in the social sciences. In *Simulating Social Phenomena* (pp. 21–40). Springer- Verlag, Berlin.

Barber, K. D., Dewhurst, F. W., Burns, R. L. D. H., & Rogers, J. B. B. (2003). Business process modelling and simulation for manufacturing management: A practical way forward. *Business Process Management Journal*, *9*(4), 527–542.

Barjis, J. (2007a). Automatic business process analysis and simulation based on DEMO. *Enterprise Information Systems*, *1*(4), 365–381.

Barjis, J. (2007b). Developing executable models of business systems. In J. Filipe, J. Cordeiro, & J. Cardoso (Eds.), *Enterprise Information Systems* (pp. 5–13). Springer Berlin Heidelberg.

Barjis, J. (2008). The importance of business process modeling in software systems design. *Science of Computer Programming*, *71*(1), 73–87.

Barjis, J., & Dietz, J. L. G. (1998). Business process modeling and analysis using GERT networks. In J. Filipe (Ed.), *Enterprise Information Systems* (pp. 71–80). Springer Netherlands.

Barjis, J., & Dietz, J. L. G. (2000). Language based requirements engineering combined with petri-nets. In R. Traunmüller (Ed.), *Information Systems* (pp. 1–12). Springer US.

Barjis, J., Dietz, J. L. G., & Liu, K. (2001). Combing the DEMO methodology with semiotic methods in business process modeling. In K. Liu, R. J. Clarke, P. B. Andersen, & R. K. Stamper (Eds.), *Information and Organisation Design Series Volume 1* (pp. 213–246). Springer US.

Barjis, J., Kolfschoten, G. L., & Verbraeck, A. (2009). Collaborative enterprise modeling. In W. Engelsman, H. Jonkers, H. M. Franken, & M.-E. Iacob (Eds.), *PRET 2009, LNBIP 28* (pp. 50–62). Springer.

Beer, M., Eisenbach, R. A., & Spector, B. (1990). Why Change Programs Don't Produce Change. *Harvard Business Review*, *68*(6), 158–166.

Bell, D. E., Raiffa, H., & Tversky, A. (1988). Descriptive, Normative and Prescriptive Interactions. In *Decision making: Descriptive, normative, and prescriptive interactions* (Vol. 1, pp. 9–32). Cambridge University Press, Cambridge.

Bettis, R. A., & Prahalad, C. K. (1995). The dominant logic: Retrospective and extension. *Strategic Management Journal*, *16*(1), 5–14.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Science*, *99*(3), 7280–7287.

Bosilj-Vuksic, V., Ceric, V. C., & Hlupic, V. (2007). Criteria for the evaluation of business process simulation tools. *Interdisciplinary Journal of Information, Knowledge & Management*, *2*(2007), 73–88.

Bradshaw, J. (1997). *Software Agents*. Cambridge MA: MIT Press.

Chan, W. K. V., Son, Y. J., & Macal, C. M. (2010). Agent-based simulation tutorial - Simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. In *Proceedings of the 2010 Winter Simulation Conference* (pp. 135–150).

Chen, P. (1976). The entity-relationship model: Towards a unified view of data. *ACM Transactions on Database Systems (TODS) - Special Issue: Papers from the International Conference on Very Large Data Bases: September 22–24, 1975, Framingham*, *1*(1), 9–36.

Chwif, L., Banks, J., Filho, J. P. de M., & Santini, B. (2013). Framework for specifying a discrete-event simulation conceptual model. *Journal of Simulation*, *7*(1), 50–60.

Davenport, T. H., Young, E., & Stoddard, D. B. (1994). Reengineering: Business change of mythic proportions? *MIS Quarterly*, *18*(Issues&Opinions: Myths About Reengineering), 121–127.

Deming, W. E. (1986). *Out of the Crisis*. Cambridge: Cambridge University Press.

Dietz, J. L. G. (2005). A world ontology specification language. In R. Meersman, Z. Tari, & P. Herrero (Eds.), *On the Move to Meaningful Internet Systems 2005: OTM 2005. Proceedings* (pp. 688–699). Springer Berlin Heidelberg.

Dietz, J. L. G. (2006). *Enterprise Ontology-Theory and Methodology*. Heidelberg: Springer- Verlag, Berlin.

Dietz, J. L. G., & Barjis, J. (2000). Petri-net expressions of DEMO process models as a rigid foundation for requirements engineering. In *Proceedings of ICEIS 2000* (pp. 267–274).

Dietz, J. L. G., & Hoogervorst, J. A. P. (2012). The Principles of Enterprise Engineering. In A. Albani, D. Aveiro, & B. Joseph (Eds.), *Advances in Enterprise Engineering VI: EEWC 2012 Proceedings, LNBIP 110* (pp. 15–30). Heidelberg: Springer- Verlag, Berlin.

Dietz, J. L. G., & Hoogervorst, J. A. P. (2014a). Enterprise Engineering.

Dietz, J. L. G., & Hoogervorst, J. A. P. (2014b). *Theories in Enterprise Engineering Memorandum 5 :the PSI theory*. Retrieved from http://www.ciaonetwork.org/uploads/eewc2014/EE-theories/TEEM-5 PSI v2.pdf

Dietz, J. L. G., Hoogervorst, J. A. P., Albani, A., Aveiro, D., Babkin, E., Barjis, J., … Winter, R. (2013). The discipline of enterprise engineering. *International Journal of Organisational Design and Engineering*, *3*(1), 86–114. doi:10.1504/IJODE.2013.053669

Du, X., Gu, C., & Zhu, N. (2012). A survey of business process simulation visualization. In *Proceeding of 2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering* (pp. 43–48). IEEE. doi:10.1109/ICQR2MSE.2012.6246184

Flores, F. (1991). *Offering New Principles for a Shifting Business World* (p. 21). Belmont CA: Business Design Associates.

Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *Informs Journal on Computing*, *14*(3), 192–215.

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the Social Scientist* (second., p. 295). New York; YN.: Open University Press.

Gladwin, B., & Tumay, K. (1994). Modeling business processes with simulation tools. In *Proceedings of the 1994 Wnter Simulation Conference* (pp. 114–121). Society for Computer Simulation International.

Greasley, A. (2003). Using business-process simulation within a business-process reengineering approach. *Business Process Management Journal*, *9*(4), 408–420. doi:10.1108/14637150310484481

Greasley, A., & Barlow, S. (1998). Using simulation modelling for BPR : Resource allocation in a police custody process. *International of Operations &Production Management*, *18*(9/10), 978–988.

Guha, S., Kettinger, W. J., & Teng, J. T. C. (1993). Business process reengineering: Building a comprehensive methodology. *Information Systems Management*, *10*(3), 13–22.

Guizzardi, G., & Wagner, G. (2012). Tutorial: conceptual simulation modeling with onto-UML. In *Proceedings of the 2012 Winter Simulation Conference* (pp. 1 – 15). Berlin: IEEE.

Hammer, M., & Champy, J. (1993). *Reengineering the corporation*. London: Nicolas Brealey.

Hammer, M., & Champy, J. (2006). *Reengineering the corporation: A manifesto for business revolution*. New York, NY: HarperCollins Publishers.

Harmon, P. (2007). *Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals* (Second Edi.). Burlington, Massachusetts: Morgan Kaufann.

Harrington, H. J., & Tumay, K. (2000). *Simulation Modeling Methods: To Reduce Risks and Increase Performance*. New York, NY: McGraw-Hill.

Harrison, J. R., Lin, Z., Carroll, G. R., & Carley, K. M. (2007). Simulation modeling in organizational and management research. *Academy of Management Review*, *32*(4), 1229–1245. doi:10.5465/AMR.2007.26586485

Holt, A., Ramsey, H. R., & Grimes, J. D. (1983). Coordination system technology as the basis for a programming environment. *Electrical Communication*, *57*(4), 307–314.

Hoogervorst, J. A. P. (2009). *Enterprise Governance and Enterprise Engineering*. Sogeti Nederland B.V: Springer. doi:10.1007/978-3-540-92671-9_1

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). *Introduction to Automata Theory, Languages and Computation.* (3rd ed.). Boston, Marathon: Addison-Wesley.

IDEF website. (n.d.). Retrieved September 23, 2014, from http://idef.com/

Iijima, J. (2014). *DEMO 企業活動の骨格を可視化するモデリング方法論*. Tokyo: NTT 出版.

Irani, Z., Hlupic, V., & Giaglis, G. M. (2002). Business process reengineering: An analysis perspective,. *Internal Journal of Flexible Manufaturing System*, *14*(32), 5–10.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, *203*(1), 1–13. doi:10.1016/j.ejor.2009.06.004

Kaufman, R. . (1992). Why operations improvement programs fail: Four managerial contradictions. *Sloan Management Review*, *34*(1), 83–93.

Keen, P. G. W., & Knapp, E. M. (1995). *Every Manager's Guide to Business Processes: A Glossary of Key Terms & Concepts for Today's Business Leader*. Boston, Massachusetts: Harvard Business School Press.

Kheir, N. A. (1996). *Systems Modeling and Computer Simulation* (2nd ed.). New York,YN: Mrcel Dekker.

Kleijnen, J. P. C. (2008). *Design and Analysis of Simulation Experiments*. New York, NY: Springer.

Kossiakoff, A., Sweet, W. N., Seymour, S., & Biemer, S. M. (2011). *Systems Engineering Principles and Practice*. Hoboken, New Jersey: Wiley.

Koubarakis, M., & Plexousakis, D. (2002). A formal framework for business process modelling and design. *Information Systems*, *27*(5), 299–319.

Krouwel, M., & Op`t Land, M. (2012). Using enterprise ontology as a basis for requirements for cross-organizationally usable applications. In *Proceedings of MCIS 2012* (p. paper 23).

Law, A. M., & Kelton, W. D. (2000). Simulation Modeling and Analysis. *ACM Transactions on Modeling and Computer Simulation*, *2*(2), 1370–1375. doi:10.1145/1667072.1667074

Levitt, B., & March, J. G. (1988). Organizational Learning. *Annual Review of Sociology*, *14*, 319–340.

Liu, Y., & Iijima, J. (2014). Automatic model transformation for enterprise simulation. In D. Aveiro, J. Tribolet, & D. Gouveia (Eds.), *Advances in Enterprise Engineering VIII: EEWC 2014 Proceedings*, *LNBIP 174* (pp. 136–150). Heidelberg: Springer- Verlag, Berlin.

Martin, J. (1995). *Using the Seven Principles of Enterprise Engineering to Align People, Technology and, Strategy*. New York, NY: American Management Association.

Melão, N., & Pidd, M. (2000). A conceptual framework for understanding business processes and business process modelling. *Infromation System*, *10*(2), 105–129.

Netjes, M. (2006). Business process simulation - A tool survey. In *Proceeding of Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark, October 2006*.

Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company*. New York, NY: Oxford University Press.

Op`t Land, M. (2006). Applying architecture and ontology to the splitting and allying of enterprises : problem definition and research approach. In *Proceedings of OTM Workshops 2006, LNCS 4287* (pp. 1419–1428).

Op`t Land, M. (2009). *Enterprse Engineering Framework* [PowerPoint slides]. Retrieved on March, 2012, from the lecture in Tokyo institute of technology, Japan.

Op`t Land, M., Proper, H. A. E., Waage, M., Cloo, J., & Steghuis, C. (2009). *Enterprise Architecture: Creating Value by Informed Governance*. Heidelberg: Springer-Verlag Berlin.

Paolucci, E., Bonci, F., & Russi, V. (1997). Redesigning organisations through business process re-engineering and object-orientation. In *Proceedings of the European Conference on Information Systems*. Cork, UK: IEEE.

Paul, R. J., Giaglis, G. M., & Hlupic, V. (1999). Simulation of business processes. *American Behavioral Scientist*, *42*(10), 1551–1576. doi:10.1177/00027649921957883

Paul, R. J., & Seranno, A. (2003). Simulation for business processes and information systems design. In *Proceedings of the 2003 Winter Simulation Conference*.

Pegden, C. D., Shannon, R. E., & Sadowski, R. P. (1995). *Introduction to Simulation using SIMAN* (Second Edi.). New York, NY: McGraw-Hill.

Perinforma, A. P. C. (2012). *The Essence of Organisation*. Leidschendam, The Netherlands: Sapio Enterprise Engineering.

Phalp, K., & Shepperd, M. (1999). Quantitative analysis of static models of processes. *Journal of Systems and Software*, *52*(2), 105–112.

Prahalad, C. K., & Bettis, R. A. (1986). The dominant logic: A new linkage between diversity and performance. *Strategic Management Journal*, *7*(6), 485–501.

Reijers, H. A., & Liman Mansar, S. (2005). Best practices in business process redesign: An overview and qualitative evaluation of successful redesign heuristics. *Internal Journal of Management*, *33*(58), 283–306.

Roberts, N., Anderson, D., Deal, R., Garet, M., & Schaffer, W. (1983). *Introduction to computer simulation: A system dynamic modeling approach*. Addison-Wesley.

Robinson, S. (2006). Conceptual modeling for simulation: issues and research requirements. In *Proceedings of the 38th conference on Winter simulation* (pp. 792–800). Monterey, CA, USA: Winter Simulation Conference.

Robinson, S., Nance, R. E., Paul, R. J., Pidd, M., & Taylor, S. J. E. (2004). Simulation model reuse: Definitions, benefits and obstacles. *Simulation Modelling Practice and Theory*, *12*(7-8 SPEC. ISS.), 479–494.

Rouse, W. B. (2005a). A theory of enterprise transformation. *Systems Engineering*, *8*(4), 279–295. doi:10.1002/sys.20035

Rouse, W. B. (2005b). Enterprises as systems: Essential challenges and approaches to transformation. *Systems Engineering*, *8*(2), 138–150. doi:10.1002/sys.20029

Rummler, G. A. (1984). Japanese Productivity: Observations of a Human Performance Technologist. *Performance and Instruction Journal*, *23*(4), 1–3.

Rummler, G. A., & Brache, A. P. (1995). *Improving Performance: How to manage the White Space on the Organizational Chart*. San Francisco: Jossey-Bass.

Salimifard, K., & Wright, M. (2001). Petri net based modelling of workflow systems : An overview. *European Journal of Operational Research*, *134*(3), 664–676.

Scholz-Reiter, B., Stahlmann, H. D., & Nethe, A. (1999). *Process Modelling*. : Heidelberg: Springer- Verlag, Berlin.

Seila, A. F. (2005). The case for a standard model description for process simulation. *International Journal of Simulation and Process Modeling*, *1*(1-2), 26–34.

Shannon, R. E. (1975). *Systems Simulation the Art and Science*. Englewood Cliffs, N.J.: Prentice-Hall.

Shannon, R. E. (1998). Introduction to the art and science of simulation. In *Proceedings of the 1998 Winter Simulation Conference* (Vol. 1). doi:10.1109/WSC.1998.744892

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204–210. doi:10.1057/jos.2010.14

Stoddard, D. B., & Jarvenpaa, S. (1995). Business process reengineering: Tactics for managing radical change. *Journal of Management Information Systems*, *12*(1 June), 81–108.

Strnadl, C. F. (2006). Aligning business and IT:The process-driven architecture model. *Information Systems Management*, *23*(4), 67–77.

Taylor, F. . (1911). *The Principles of Scientific Management*. New York, NY: Harper Brothers.

Technologies, X. (2009). AnyLogic home page. Retrieved October 01, 2014, from http://www.anylogic.com/

Ter Hofstede, A. H. M., van der Aalst, W. M. P., & Adams, M. (2010). *Modern Business Process Automation: YAWL and its Support Environment*. Heidelberg: Springer- Verlag, Berlin.

Toussaint, P. J., Bakker, A., & Groenewegen, L. (1997). Construct ingan enterprise viewpoint: Evaluation offour business modelling techniques. *Computing Methodsand Programsin Biomedicine*, *55*(1), 11–30.

Tumay, K. (1996). Business Process Simulation. In *Proceedings of the 1996 Winter Simulation Conference* (pp. 93–98). Coronado, CA.: IEEE.

Turnitsa, C., Padilla, J. J., & Tolk, A. (2010). Ontology for modeling and simulation. In *Proceedings of the 2010 Winter Simulation Conference* (pp. 643–651).

Valiris, G., & Glykas, M. (2004). Business analysismetrics for business process redesign. *Business Process Management Journal*, *10*(4), 445–480.

Van der Aalst, W. M. P., ter Hofstede, A. H. M., & Weske, M. (2003). Business process management: A survey. In *Business Process Management*, *LNCS 2678* (Vol. 2678, pp. 1–12). Springer-Verlag.

Vargo, S. L., & Lusch, R. F. (2008). Service-dominant logic: Continuing the evolution. *Journal of the Academy of Marketing Science*, *36*(1), 1–10.

Vargo, S. L., Maglio, P. P., & Akaka, M. A. (2008). On value and value co-creation: A service systems and service logic perspective. *European Management Journal*, *26*(3), 145–152.

Vergidis, K., Tiwari, A., & Majeed, B. (2002). Business Process Analysis and Optimization: Beyond Reengineering. *Internal Journal of Flexible Manufaturing System*, *14*(32), 5–10. doi:10.1109/TSMCC.2007.905812

Wainer, G. A., & Mosterman, P. J. (2011). *Discrete-Event Modeling and Simulation: Theory and Application*. Boca Raton, Florida: CRC Press.

Wang, W., & Brooks, R. J. (2007). Empirical investigations of conceptual modeling and the modeling process. In *Proceedings of the 2007 Winter Simulation Conference* (pp. 762–770). Washington, D.C.: IEEE.

Warren, J. R., Crosslin, R. L., & MacArthur, P. J. (1995). Simulation modeling for BPR Steps to Effective Decision Support. *Information Systems Management*, *12*(4), 32–42.

Wooldridge, M., & Jennings, N. R. (1995). Agent theories, architectures, and languages: A survey. In *Proceedings of ECAI-94 Workshop on Agent Theories, Architectures, and Languages,* (pp. 1–32). Springer-Verlag.

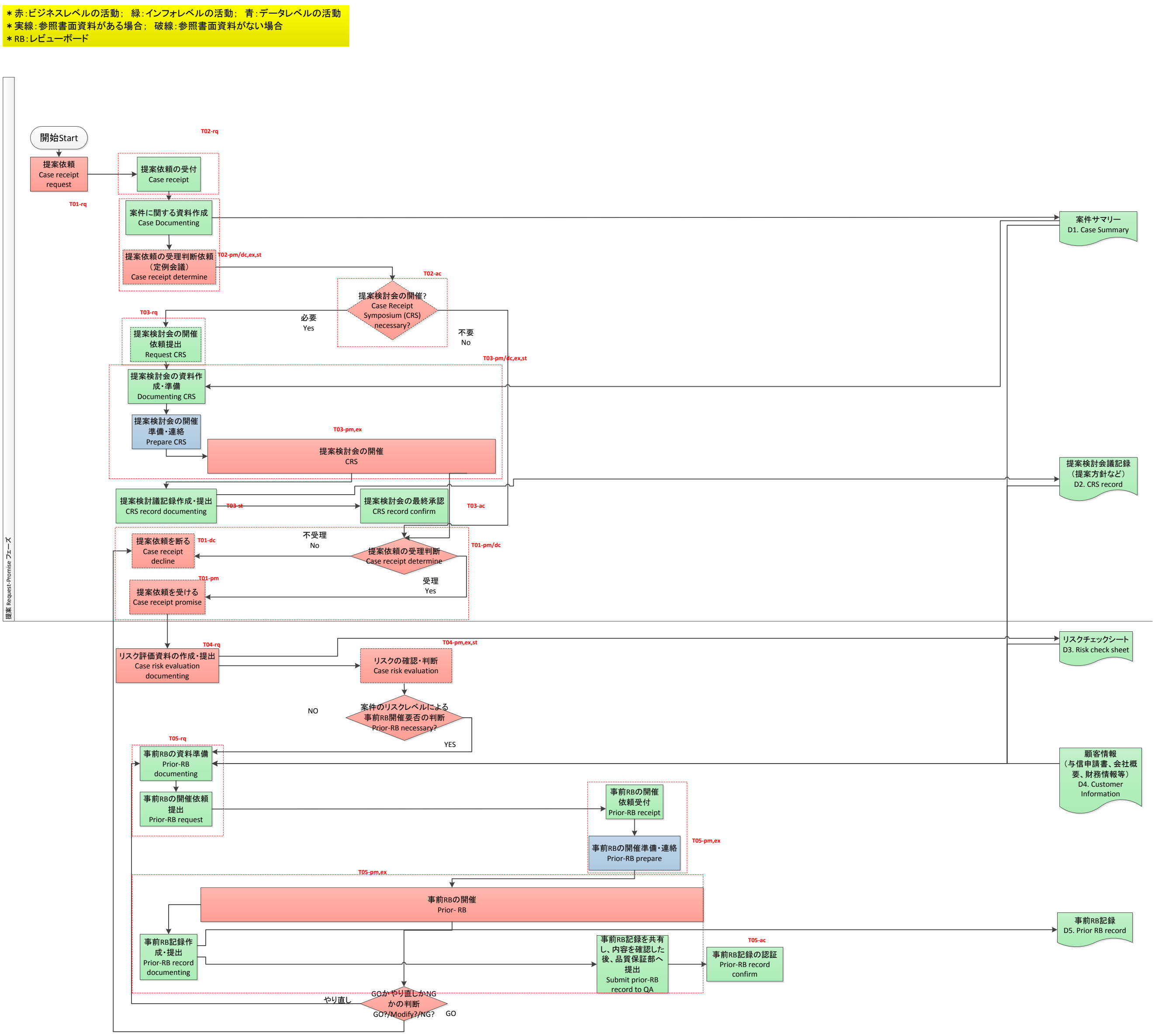
Zeigler, B. P., & Hammonds, P. E. (2007). *Modeling and Simulation-Based Data Engineering*. Waltham, Massachusetts: Academic Press.

Zeigler, B. P., Kim, T. G., & Praehofer, H. (2000). *Theory of Modeling and Simulation* (2nd Edit.). San Diego, California: Academic Press.

Appendix A. Details of "Case D"

A.1 Work Flow

The workflow of the proposal and estimation process in "Company D" is given in Figure A.1. The red boxes indicate business-level activities; green boxes indicate information-level activities, and the blue boxes indicate data-level activities. The red rectangles denote transactions in DEMO concepts. Diamonds indicate the judgment points, and the green shapes on the right denote required documents and information. The workflow was described in section 6.2.



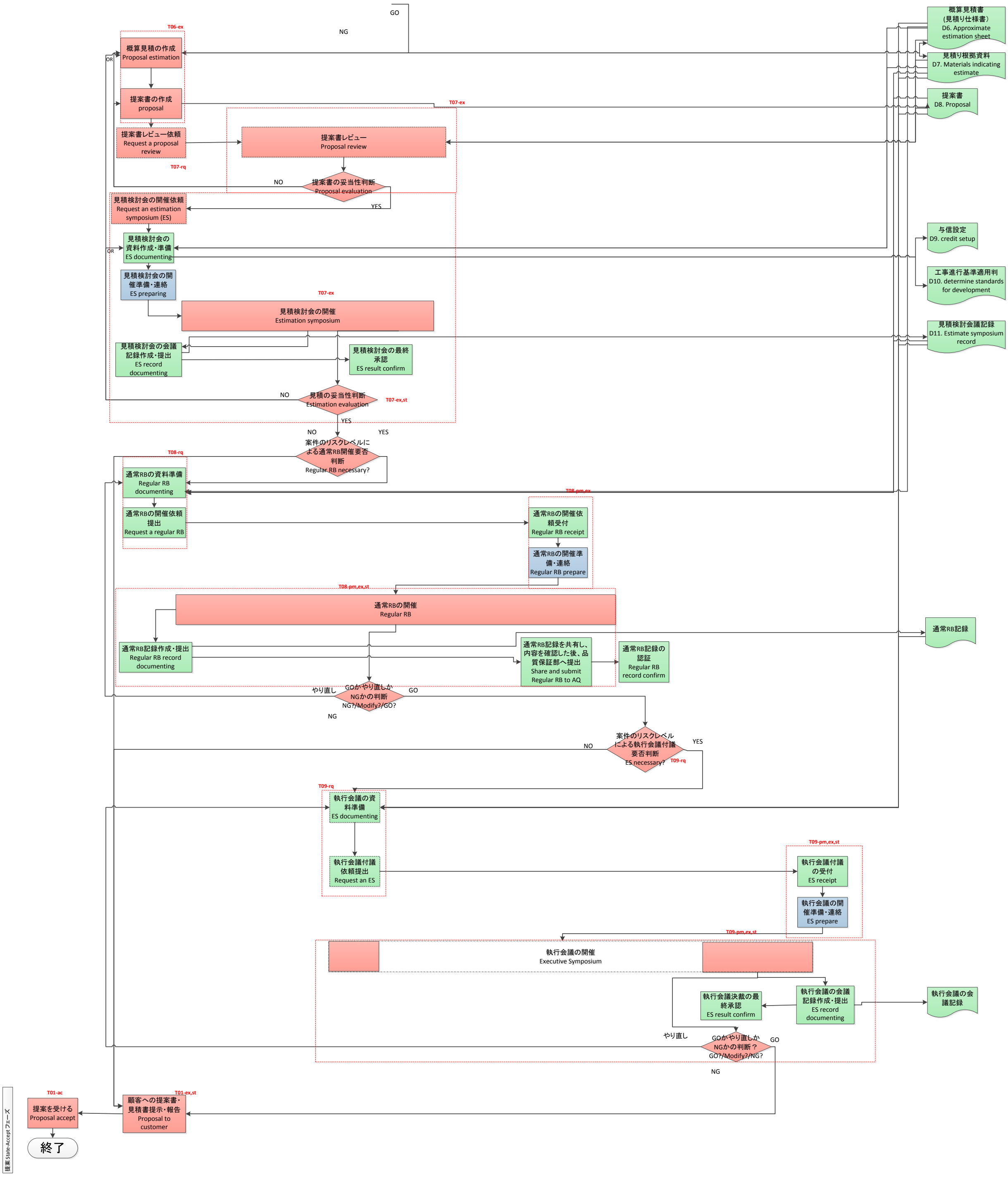


Figure A.1 Workflow of the proposal and estimation process in “Company D”

A.2 Implementation Models of Actor Roles

The remaining components of the actor roles for A2, A3, A4, A5, A6, A7, A8 and A9 are listed below.

A2 is the processor for T2pm, T2ex and T3rq. T2pm includes case activities documentation, case reception evaluation and decision-making about whether a CRS should be requested. T2ex decides whether to receive the request. T3rq requests the CRS if necessary.

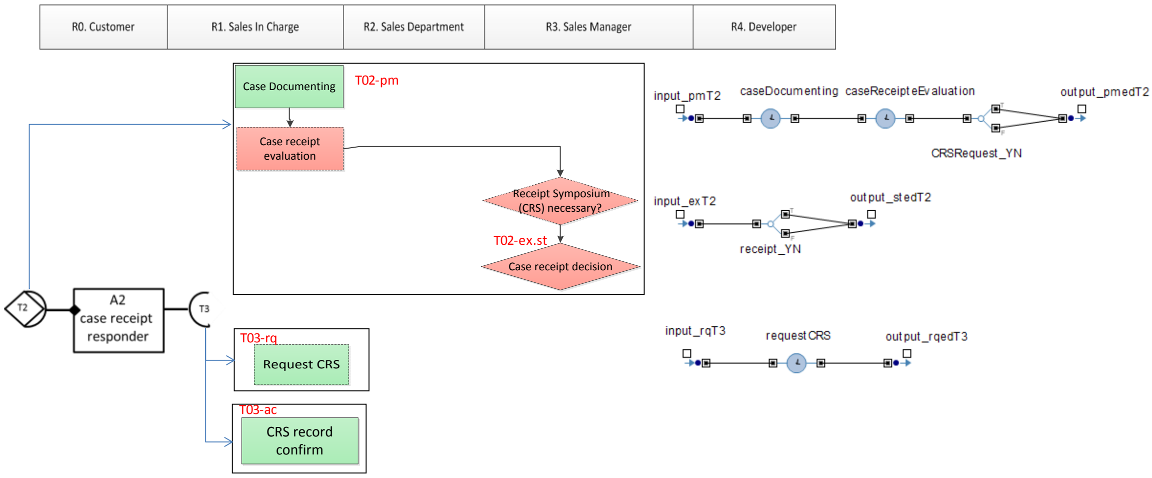


Figure A.2 Implementation Model for A2

A3 is the processor for T3ex. The execution requires resource sales to prepare documentation for the CRS and prepare for the CRS. After the CRS, the report must be documented.

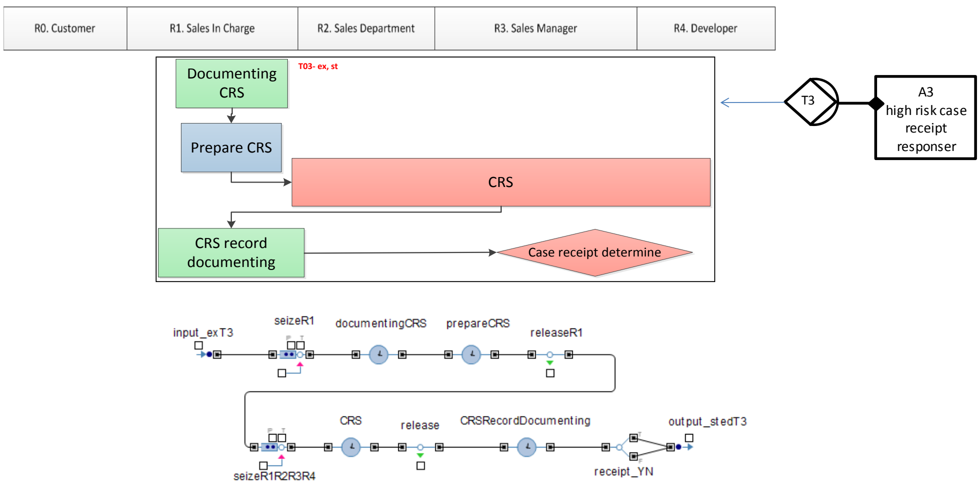


Figure A.3 Implementation Model for A

A4 is the processor for T4ex. The executor evaluates the risk of cases and decides whether to request a PRB or not.

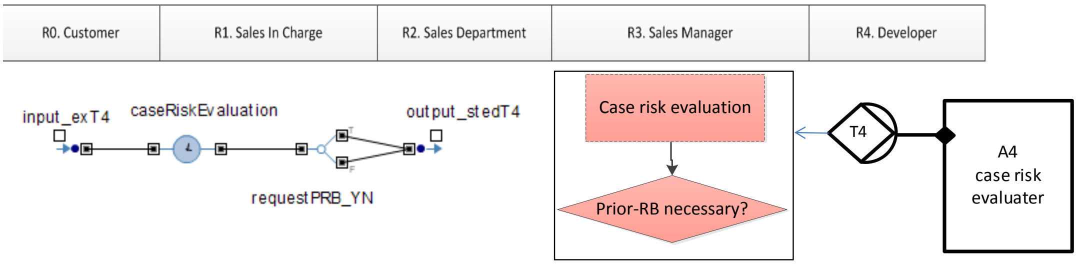


Figure A.4 Implementation Model for A4

A5 is the processor for T5pm/dc. It requires as a resource RB office staff members to receive PRB requests and prepare for the review board. After the PRB, the documents must be reported with the decision on the evaluation result.

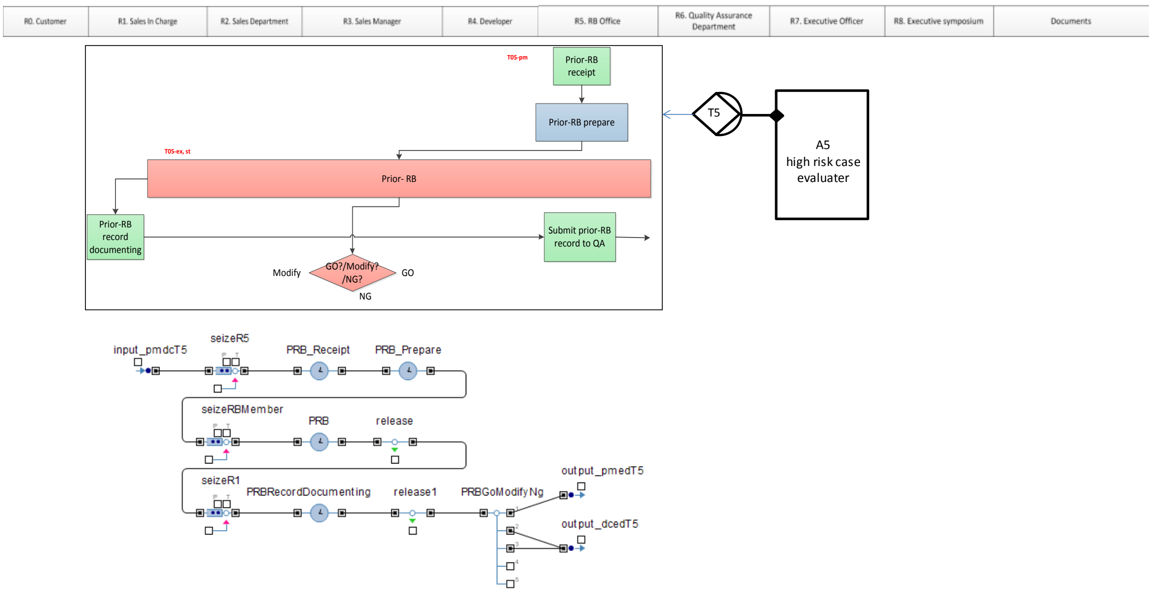


Figure A.5 Implementation Model for A5

A6 is the processor for T6ex. It requires sales people to prepare and estimate the proposal.

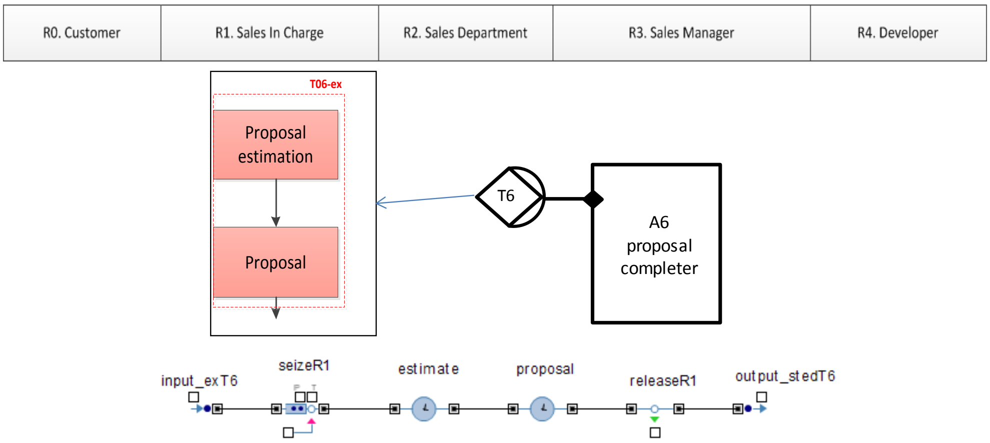


Figure A.6 Implementation Model for A6

A7 is the processor for T7ex and T8rq. The execution of T7 includes asking sales managers and development staff to review the proposal. For high-risk cases, an estimation symposium is required. In such a case, the sales managers must prepare documents and, after the ES, prepare a report on whether the proposal and the estimate are acceptable.

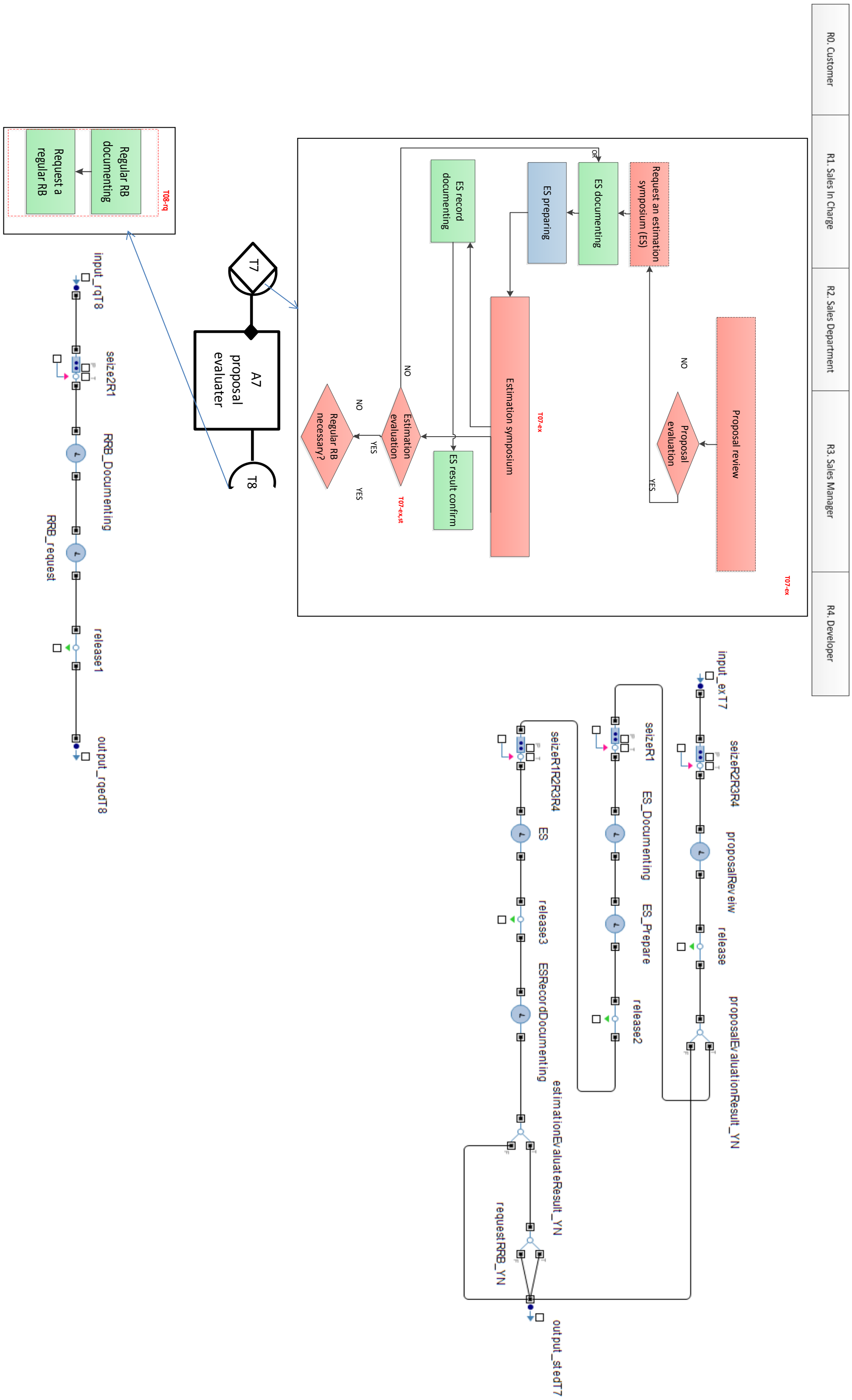


Figure A.7 Implementation Model for A7

A8 is the processor for T8pm/dc and T9rq. The promise and decline of T8 is to receive and prepare for the RPB. After the RPB, the report must be documented with the decision on whether to promise the request or decline it. The request for T9 requires the sales manager to prepare all documents and request the executive symposium.

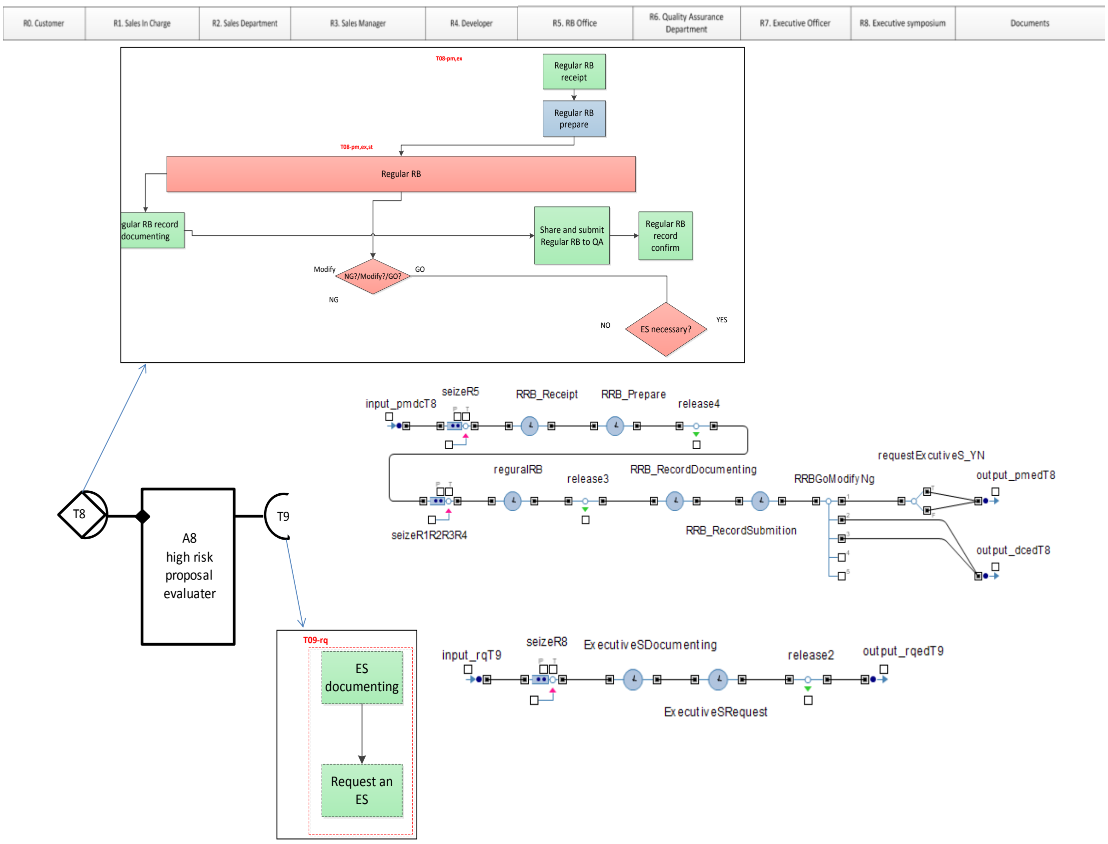


Figure A.8 Implementation Model for A8

A9 is the processor for T9pm/dc. It requires the reception of and the preparation for the ES. After the ES, the decisions must be documented and reported.

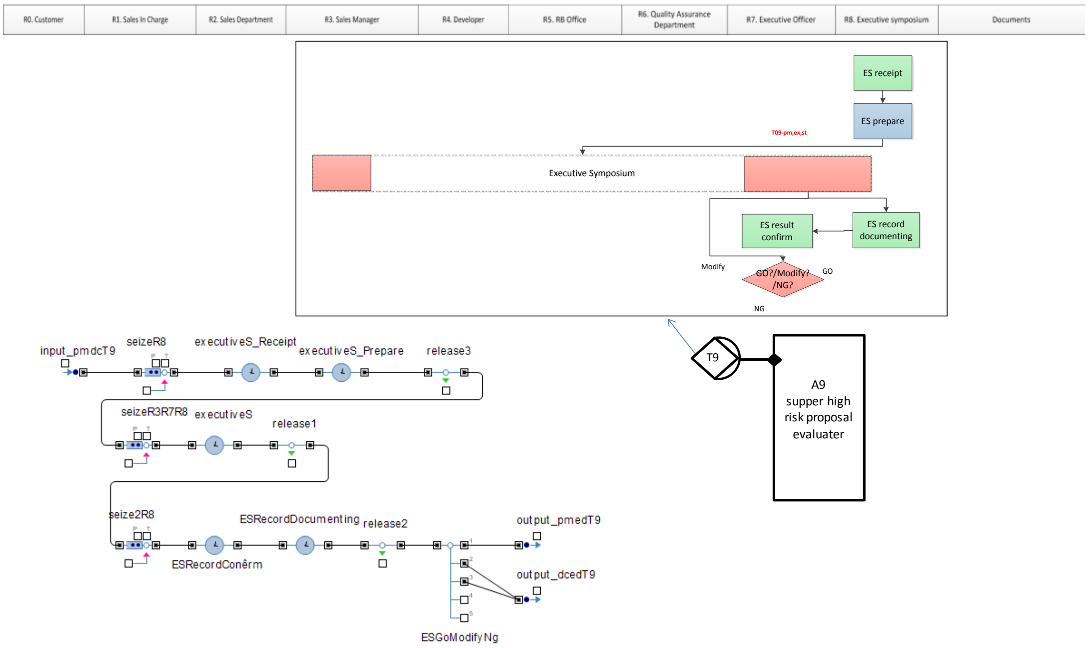


Figure A.9 Implementation Model for A9

A.3 Action Rules

The action rules for actor role A2- A9 are presented in Table A.1 and are explained in sections 3.2.4 and 6.3.4. Parts of the AM models are derived from the PM model; the condition-related items were added manually.

Table A.1 Action Rules for A2-A9.

AR:	A2														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT		CONDITION			ASSIGMENT	
When	rqedT2	For	Case	is done.	If	none	Then	pmT2	For	Case	With	If	none	Then	none
When	pmedT2	For	Case	is done.	If	case.crsRequest_YN==true	Then	rqT3	For	Case	With	If	none	Then	none
					If	case.crsRequest_YN==false	Then	exT2	For	Case	With	If	risk of the case is acceptable	Then	case.receipt_YN=true
												If	risk of the case is not acceptable	Then	case.receipt_YN=false
							Then	stT2	For	Case	With	If	none	Then	none
When	stedT3	For	Case	is done.	If	none	Then	acT3	For	Case	With	If	none	Then	none
When	acedT3,pmedT2	For	Case	is done.	If	none	Then	exT2	For	Case	With	If	none	Then	none
							Then	stT2	For	Case	With	If	none	Then	none

AR:	A3														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT		CONDITION			ASSIGMENT	
When	rqedT3	For	Case	is done.	If	none	Then	pmT3	For	Case	With	If	none	Then	none
When	pmedT3	For	Case	is done.	If		Then	exT3	For	Case	With	If	risk of the case is acceptable	Then	receipt_YN=true
												If	risk of the case is not acceptable	Then	receipt_YN=false
							Then	stT3	For	Case	With	If	none	Then	none

AR:	A4														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT		CONDITION			ASSIGMENT	
When	rqedT4	For	Case	is done.	If	none	Then	pmT4	For	Case	With	If	none	Then	none
When	pmedT4	For	Case	is done.	If	none	Then	exT4	For	Case	With	If	none	Then	none
							Then	stT4	For	Case	With	If	none	Then	none
When	stedT5	For	Case	is done.	If	none	Then	acT5	For	Case	With	If	none	Then	none
When	acedT4	For	Case	is done.	If	PRB is required	Then	rqT5	For	Case	with	If	none	Then	requestPRB_YN=true
When	dcedT5	For	Case	is done.	If	case.prbResult_YN=″MODIFY″	Then	rqT5	For	Case	With	If	none	Then	none
					If	case.prbResult_YN=″NG″	Then	qtT5	For	Case	With	If	none	Then	none

AR:	A5														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT		CONDITION			ASSIGMENT	
When	rqedT5	For	Case	is done.	If	risk of the case is acceptable	Then	pmT5	For	Case	With	If	none	Then	case.prbResult_YN=″GO″
					If	risk of the case need to be re-evaluated	Then	dcT5	For	Case	With	If	none	Then	case.prbResult_YN=″MODIFY″
					If	risk of the case is unacceptable	Then	dcT5	For	Case	With	If	none	Then	case.prbResult_YN=″NG″
When	pmedT5	For	Case	is done.	If	none	Then	exT5	For	Case	With	If	none	Then	none
							Then	stT5	For	Case	With	If	none	Then	none

AR:	A6														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT		CONDITION			ASSIGMENT	
When	rqedT6	For	Proposal	is done.	If	none	Then	pmT6	For	Proposal	With	If	none	Then	none
When	pmedT6	For	Proposal	is done.	If	none	Then	exT6	For	Proposal	With	If	none	Then	none
			Proposal				Then	stT6	For	Proposal	With	If	none	Then	none

AR: A7

WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
When	rqedT7	For	Proposal	is done.							With	If	the proposal is acceptable	Then	proposal.proposalEvaluationResult_YN =true
												Else		Then	proposal.proposalEvaluationResult_YN =false
												If	the estimate is acceptable	Then	proposal.estimateEvaluationResult_YN=true
												Else		Then	proposal.estimateEvaluationResult_YN=false
												If	a regular RB is required	Then	proposal.requestRRB_YN=true
												Else		Then	proposal.requestRRB_YN=false
					If	(proposal.proposalEvaluationResult_YN ==true)&&(proposal.estimateEvaluationResult_YN==true)	Then	pmT7	For	Proposal	With	IF	none	Then	None
					Else		Then	dcT7	For	Proposal	With	If	none	Then	none
When	pmedT7	For	Proposal	is done.	If	none	Then	exT7	For	Proposal	With	If	none	Then	none
							Then	stT7	For	Proposal	With	If	none	Then	none
When	dcedT7	For	Proposal	is done.	If	none	Then	rqT7	For	Proposal	With	If	none	Then	none
When	stedT7	For	Proposal	is done.	If	requestRRB==true	Then	rqT8	For	Proposal	With	If	none	Then	none
When	acedT9,stedT8	For	Proposal	is done.	If	none	Then	acT8	For	Proposal	With	If	none	Then	none
When	dcedT8	For	Proposal	is done.	If	proposal.rrbResult_YN="MODIFY"	Then	rqT8	For	Proposal	With	If	none	Then	none
					If	proposal.rrbResult_YN="NG"	Then	qtT8	For	Proposal	With	If	none	Then	none
When	stedT8	For	Proposal	is done.	If	proposal.requestExecutiveS_YN ==false	Then	acT8	For	Proposal	With	If	none	Then	none

AR: A8

WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
When	rqedT8	For	Proposal	is done.							With	If	Executive Symposium is required	Then	proposal.requestExecutiveS_YN =true
												Else		Then	proposal.requestExecutiveS_YN =false
					If	risk of the proposal is acceptable	Then	pmT8	For	Proposal	With	If	none	Then	proposal.rrbResult_YN="GO"
					If	risk of the proposal need to be re-evaluated	Then	dcT8	For	Proposal	With	If	none	Then	proposal.rrbResult_YN="MODIFY"
					If	risk of the proposal is unacceptable	Then	dcT8	For	Proposal	With	If	none	Then	proposal.rrbResult_YN="NG"
When	pmedT8	For	Proposal	is done.	If	none	Then	exT8	For	Proposal	With	If	none	Then	none
							Then	stT8	For	Proposal	With	If	none	Then	none
When	stedT8	For	Proposal	is done.	If	proposal.requestExecutiveS_YN ==true	Then	rqT9	For	Proposal	With	If	none	Then	none
When	acedT8, stedT9	For	Proposal	is done.	If	none	Then	acT9	For	Proposal	With	If	none	Then	none
When	dcedT9	For	Proposal	is done.	If	proposal.esResult_YN= "MODIFY"	Then	rqT9	For	Proposal	With	If	none	Then	none
					If	proposal.esResult_YN= "NG"	Then	qtT9	For	Proposal	With	If	none	Then	none

AR:	A9														
WHEN					THEN						WITH				
FACT		OBJECT			CONDITION		REACT		OBJECT			CONDITION		ASSIGMENT	
When	rqedT9	For	Proposal	is done.	If	risk of the proposal is acceptable	Then	pmT9	For	Proposal	With	If	none	Then	proposal.esResult_YN="GO"
					If	risk of the proposal need to be re-evaluated	Then	dcT9	For	Proposal	With	If	none	Then	proposal.esResult_YN="MODIFY"
					If	risk of the proposal is unacceptable	Then	dcT9	For	Proposal	With	If	none	Then	proposal.esResult_YN="NG"
When	pmedT9	For	Proposal	is done.	If	none	Then	exT9	For	Proposal	With	If	none	Then	none
							Then	stT9	For	Proposal	With	If	none	Then	none

Appendix B. DEMO++ Library T

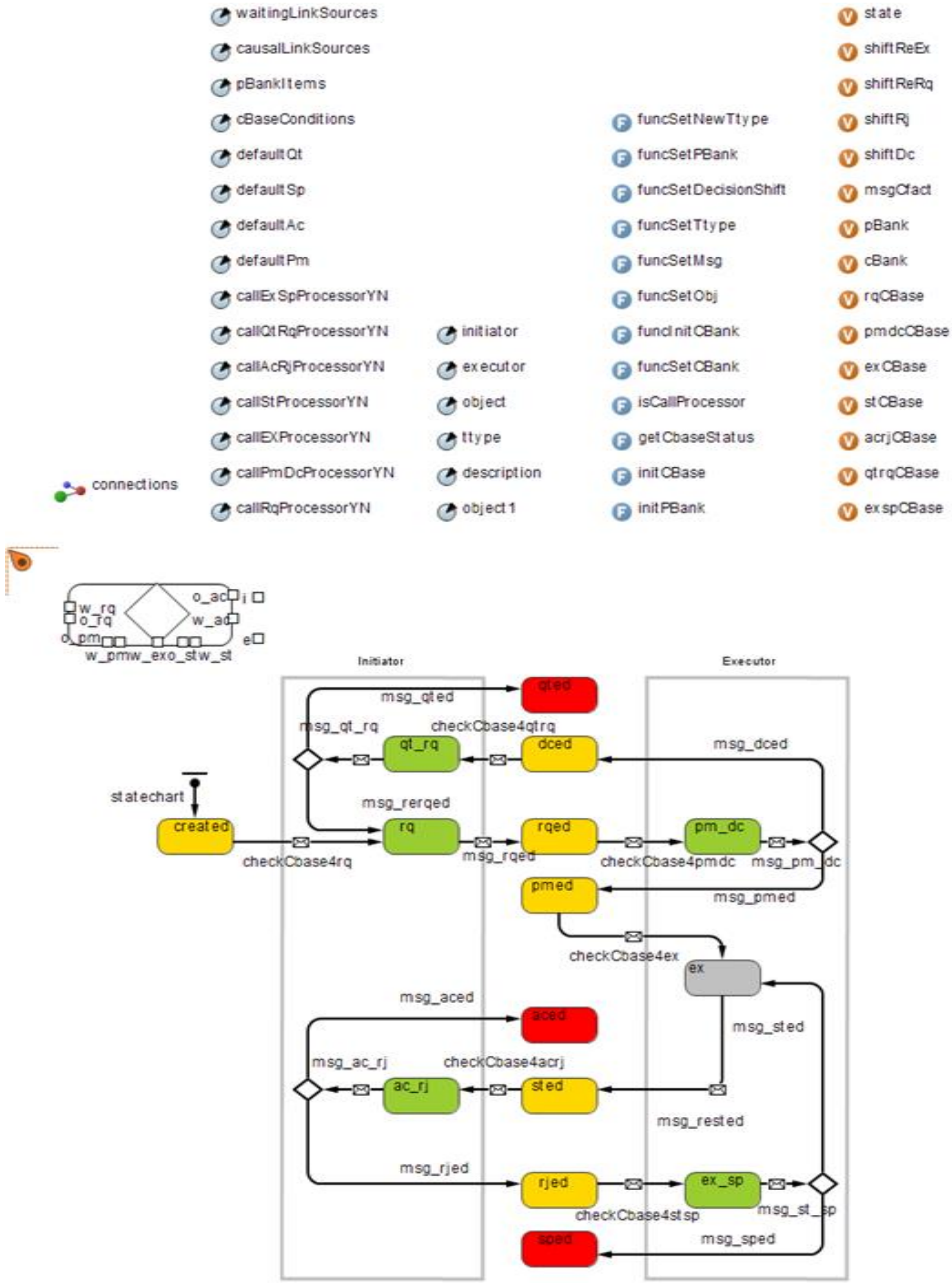


Figure B.1 DEMO++ Library T

Transaction types are defined as a library element T, as shown in Figure B.1. As described in section 5.4.2.2, each transaction type is defined as a standard transaction parent, including two types of statuses, acts and facts (ending with "ed") and two types of transitions: transition from act to fact (beginning with "msg_") and transition from fact to act (beginning with "checkCbase4"). The functions are called act, fact, transition "msg_" and transition "checkCbase4".

The class diagram is presented in Figure B.2. Act and fact are types of States and msg_ and checkCBase4 are types of Triggers.

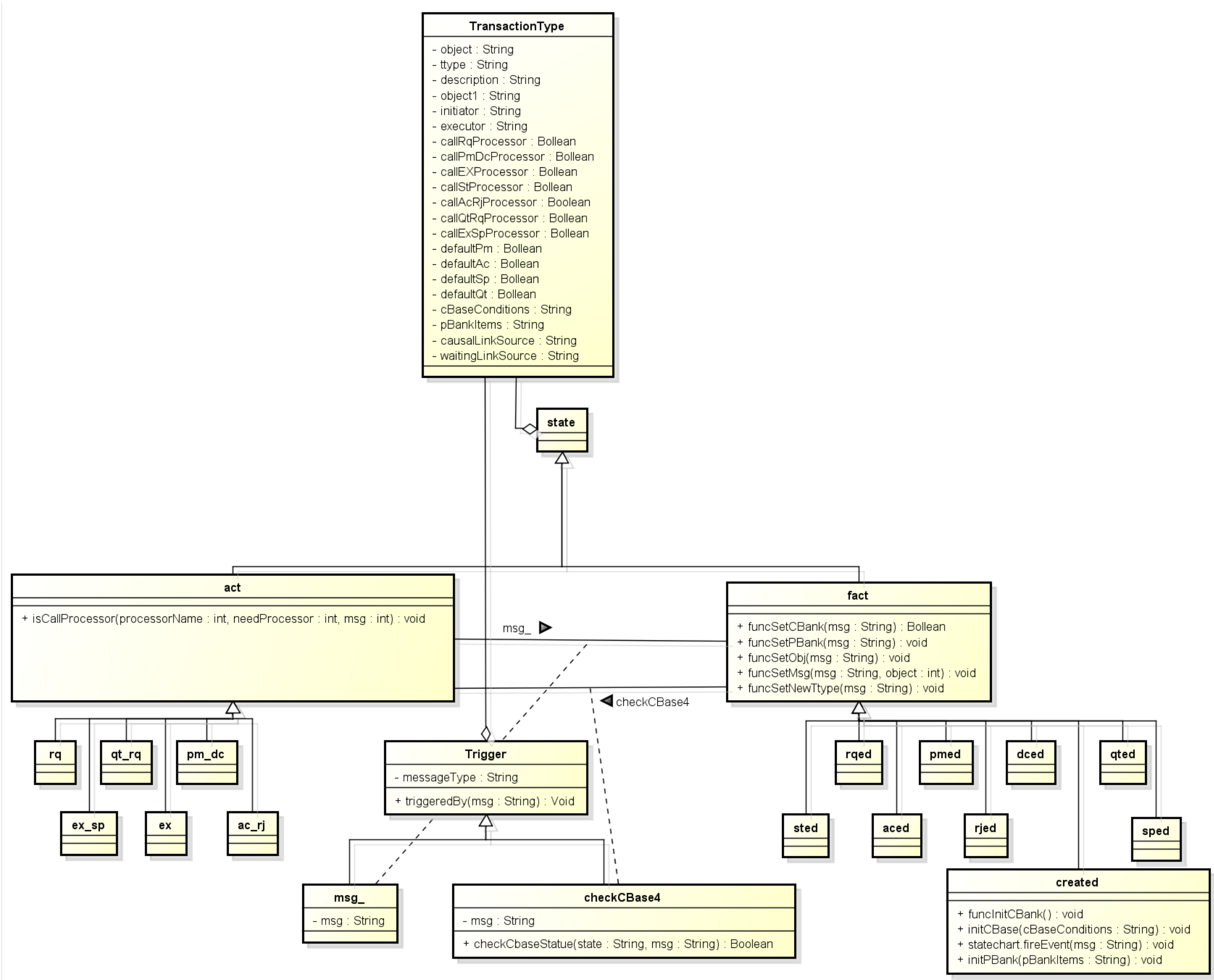


Figure B.2 Class Diagram of Transaction Type T

- Act describes the proceeding with the act. The proceeding can be described in only ontological level or in detail using its processor;
  - **In act "rq"**: an act must read perimeters to decide whether to call its

processor.

```
isCallProcessor(callRqProcessorYN, false, "rqed");
```

- Fact describes the status after proceeding with the act.
  - **In state "created"**: "created" is an initiation state in which CBank, PBank, CBase, and PBase need to be initiated.

    ```
    funcInitCBank();
    initCBase(cBaseConditions);
    initPBank(pBankItems);
    statechart.fireEvent("created");
    funcSetPBank("created");
    funcSetCBank("created");
    ```

  - **In state "rqed"**: a state must read set CBank, PBank, Object and build up messages according to the state.

    ```
    funcSetCBank("rqed");
    funcSetPBank("rqed");
    funcSetObj("rqed");
    funcSetMsg("rqed",0);
    funcSetNewTtype("rqed");
    ```

- Transition "msg_" describes the required trigger messages for changing from act to fact. For example, as the required triggers for changing from the act "rq" to the fact "rqed", a message "rqed" is required;
- Transition "checkCbase4"" describes the required conditions for proceeding with an act. For example, to proceed with act "rq", the trigger must check its Cbase to verify that all required conditions are satisfied, as shown in Figure B.2:

```
getCbaseStatus("rq", msg);
```