TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Ph.D. Thesis

Requirements Management for B2B Processes: a Worksheet Driven Approach from e3-Value and REA to UMM

Conducted for the purpose of receiving the academic title
'Doktor der Sozial- und Wirtschaftswissenschaften'

Advisors

Univ.Prof. Dipl.-Ing. Dr.techn. Hannes Werthner

and

Ao.Univ.Prof. Mag. Dr. Christian Huemer

Institute of Software Technology and Interactive Systems (E188)

Vienna University of Technology

Submitted at the

Vienna University of Technology

Faculty of Informatics

by

Rainer Schuster

0026575

Rennweg 53/25

A-1030 Wien

Vienna, 18th May, 2010

Kurzfassung der Dissertation

Die Entwicklung von B2B Systemen im Bereich e-Commerce ist von verschiedenen ökonomischen und strategischen Parametern abhängig. Aus diesem Grund können sich die Anforderungen an B2B Systeme aufgrund wirtschaftlicher Anpassungen von Unternehmen im Laufe der Zeit verändern. Ein Grund hierfür ist z.B. die Gründung neuer Partnernetzwerke oder die Änderung der Geschäftsstrategie einer Firma. Ein gängiges Problem aus der Praxis ist, dass die Business Analysten (Modellierer) bei der Entwicklung von inter-organisationalen Prozessen die Einbeziehung solcher wirtschaftlicher Parameter nicht beachten. Die Folge ist eine Anforderungsanalyse, die nur auf der Prozessebene stattfindet, nicht aber die Geschäftsmodelle des Partnernetzwerkes mit einbezieht. Diese Trennung von Geschäftsmodell und Geschäftsprozessmodel birgt hinsichtlich der Anpassung, Wartung und Weiterentwicklung des IT-Systems, das diese Modelle implementiert, ein sehr kostenintensives Risiko.

In dieser Dissertation wird ein einheitlicher Ansatz zur Spezifikation der Anforderungen von B2B Systemen unter Beachtung der wirtschaftlichen Faktoren beschrieben. Als erster Schritt wird ein Requirements Engineering Ansatz im Sinne von Service orientierten Architekturen (SOA) vorgestellt. Dabei wird eine auf sechs Phasen basierende Methode präsentiert, mit Hilfe derer der Analyst die internen Prozesse modellieren und die Schnittstellen zu den IT Systemen der Geschäftspartner spezifizieren kann. Dieser Ansatz beachtet bereits in den ersten beiden Phasen wirtschaftliche Aspekte, die in die späteren Phasen der Prozessmodellierung einfließen.

Da der oben genannte Ansatz auf die interne Geschäftsprozessmodellierung fokussiert, wird im nächsten Schritt ein auf Geschäftsmodelle basierender inter-organisationaler Ansatz vorgestellt. Dabei greifen wir auf drei im Bereich e-Business Modellierung etablierten Methoden zurück: e^{3}value, REA (Resource-Event-Agent) und UMM (UN/CEFACT's Modeling Methodology). e^{3}value hilft dabei, eine innovative e-Business Idee, die innerhalb eines Partnernetzwerkes umgesetzt werden soll, zu untersuchen. Dies beginnt bei der Modellierung des Werteaustauschs zwischen den verschiedenen Partnern und endet bei der Erhebung der Rentabilität der Geschäftsidee. Während e^{3}value eher die wirtschaftliche Nachhaltigkeit der B2B Lösung untersucht, dient ein REA Modell zur Überprüfung der Einhaltung des "Give-and-Take" Prinzips. Dieses Prinzip wird auch ökonomische Reziprozität genannt und stellt sicher, dass eine Ressource (z.B. ein Service, ein Datensatz, oder Geld, etc.), die zwischen zwei oder mehreren Partnern ausgetauscht wird, immer eine Gegenressource erfordert. Diese beiden Methoden decken den Bereich der Geschäftsmo-

delle ab und dienen als Verknüpfungspunkt zur Geschäftsprozessmodellierung mittels UMM. UMM ist eine international anerkannte standardisierte Methode zur Spezifikation von B2B Prozessen. Der derzeitige Standard dieser Methode wurde von uns als aktives Mitglied von UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) mitentwickelt.

Wir integrieren diese drei Methoden in unseren Ansatz und spezifizieren konzeptionelle Übergänge zwischen den verschiedenen Modellierungssprachen. Darüber hinaus werden diese Übergänge durch Modell-Transformationsregeln formalisiert, um eine automatisierte Generierung bestimmter Artefakte zu ermöglichen. Somit bieten wir dem Modellierer eine schrittweise Anleitung zum Erstellen der verschiedenen Modelle, die zur Spezifikation der Anforderungsanalyse von geschäftsübergreifenden Systemen notwendig sind.

Der daraus resultierende Ansatz wird von sogenannten Worksheets unterstützt, die mittels vordefinierter Formulare und einer prototypischen Tool-Entwicklung den Analysten bei der Erhebung der Anforderungen unterstützen soll. Als Proof-of-Concept wird ein realer Geschäftsfall aus dem Bereich Zeitungsdruck verwendet.

Zusammengefasst liefert die vorliegende Arbeit folgende fünf Beiträge zum Stand der Forschung im Bereich B2B: (1) eine Prozessbasierte 6-Phasen Methode zur Spezifikation der Anforderungen im Kontext von Service orientierten Architekturen; (2) ein von Geschäftsmodellen gesteuerter Ansatz zur Erstellung von inter-organisationalen Geschäftsprozessen; (3) Erweiterungen zur Verbesserung der Geschäftsmodellierungssprache REA; (4) Formalisierung der Transformationsregeln zwischen den verschiedenen Modellierungssprachen zur automatischen Modellgenerierung; (5) ein auf Worksheets basierender Ansatz zur schrittweisen Anleitung der Anforderungserhebung des Analysten.

Dementsprechend liefert diese Dissertation eine Verbesserung des gegenwärtigen Ansatzes zur Spezifizierung der Anforderungen von B2B Systemen unter Beachtung von ökonomischen Faktoren.

Abstract

Inter-organizational B2B systems are most likely tending to change their business requirements over time - e.g. establishing new partnerships or change existing ones. The problem is, that business analysts disregard the economic drivers when they start to design the business processes from scratch. Those economic drivers are captured by business models representing the business value perspective. In order to quickly adapt the B2B processes on the business process layer to changing business requirements without the need to change the overall architecture, a link between business models and business process models is needed. Traditional requirements engineering approaches for developing IT systems do not integrate business modeling techniques. Thus, a comprehensive approach is needed that is tailored for gathering the requirements without disregarding such aspects.

In this thesis we provide a requirements engineering approach, specifically designed for the development of B2B processes. First, we define a process based requirements engineering approach that consists of phases and iterations leading to a formalized and unambiguous requirements specification. Second, we propose to use business modeling techniques to ensure that business processes beneath do not violate the domain rules, i.e. to fulfill the basic economic principle for every business transaction - the give-and-take convention, called economic reciprocity. The latter one is reflected by one of the key contributions within this thesis, in which we provide a mapping between two of the most prominent business modeling ontologies - e^{3}value and REA (Resource - Event - Agent) - to a standardized business process modeling methodology called UMM (UN/CEFACT's Modeling Methodology). However, there exists already some scientific work proposing preliminary attempts to combine theses different ontologies. Most of theses approaches aim for a "horizontal" mapping between each other - i.e, the definition of a "global ontology". In this thesis we also discuss these approaches and provide distinctions to our comprehensive "vertical" approach.

e^{3}value was designed for getting a first overview of the economic values exchanged in a business network. Furthermore, e^{3}value offers the possibility to proof the economic sustainability of the business idea by quantifying the net value flow for each actor in the value web. Whereas e^{3}value concentrates more on the profitability of the IT system, an REA business model focuses on issues that may be relevant for the implementation and alignment of an IT system from an economical point of view. REA is currently in the development phase to become an official UN/CEFACT (United Nations Centre for Trade

Facilitation and Electronic Business) standard. In contrary, UMM is already standardized by UN/CEFACT for modeling the global choreography of inter-organizational business processes. The methodology is defined as a UML profile, which is specified by a set of stereotypes, tagged values and constraints. UMM is used to express and evaluate agreements and commitments between the business partners in order to provide a requirements specification for the software engineers to bind the private process interfaces to the public ones.

The formalisms introduced by business modeling ontologies and UMM's stereotypes facilitate the communication with the software engineers. However, business experts - who usually have a very limited understanding of modeling notations (e.g., UML) - tend to express their thoughts and evaluating the results by plain text descriptions. In this thesis we describe an approach that presents an equivalent of the artifacts delivered by business modeling ontologies as well as UMM stereotypes and tagged values in text-based templates called worksheets. This strong alignment allows an integration into any tool environment and ensures consistency. We show how a specially designed XML-based worksheet definition language allows customization to special needs of certain business domains. Furthermore, we demonstrate how information kept in worksheets may be used for the semi-automatic generation of pattern-based UMM artifacts.

The scientific challenge of this thesis is the combination of the three methodologies e^{3}value, REA and UMM by incorporating them into a comprehensive requirements engineering method. During the development of our proposed approach we discovered shortcomings of those methodologies. Due to the fact, that we are active co-editors of the latter two UN/CEFACT standards, we are able to integrate adequate solutions into both, the approach as provided in this thesis, and the standard specifications maintained by UN/CEFACT.

After having specified the conceptual links between the different modeling methodologies, we formalize the mapping by the use of the model-to-model ATLAS transformation language (ATL). As a proof-of-concept we use a real life business scenario out of the print media domain. At the end of the development phase, the business partners involved in this use case scenario were invited to participate in a first evaluation of our approach. The results provide valuable feedback from an industry perspective and highlight pros and cons of the methodology.

In summary, this thesis provides the following five contributions towards the development of B2B processes: (1) a comprehensive methodology for process-based requirements engineering based on phases and iterations; (2) a business modeling approach providing conceptual mapping rules between three well-established e-business methodologies; (3) improvements to REA in order to overcome the revealed limitations of the ontology; (4) the formalization of the transformation rules by means of the ATLAS transformation language; (5) a worksheet-driven approach guiding the business analyst through the requirements elicitation phase.

Contents

1 Introduction

The "blank sheet of paper" problem

The nightmare of every requirements engineer or business analyst is to conduct a new requirements elicitation (e.g. for the development of an IT system) without having any glue of the problem domain, the company, or the participating business partners. It must be even more frustrating if the requirements engineer is not supported by co-operative business experts that do not share any valuable domain knowledge. Finally, the tip of the iceberg would be, if the requirements engineer even has no guideline, recipe or idea on how to conduct the interview with the domain experts and how to structure the gathered information. As a matter of fact, the requirements engineer sits in front of a blank sheet of paper and is not able to deliver realizable specifications for software engineers. In the scientific area of requirements engineering such a phenomena is called the "blank sheet of paper" problem identified by Kilov and Simmonds [85]. The "blank sheet of paper" problem can appear when a requirements engineer is entering an uninvestigated business area, or is bombarded by a large, unstructured body of material. Following Kilov and Simmonds, elicitation of requirements demands the adoption of an appropriate frame of reference and "units of thought". In the absence of co-operative domain experts, existing showcase materials, and appropriately structured guidelines, the "unit of thought" is absent and candidates for an unambiguous requirements specification may be unclear.

Requirements engineering for inter-organizational systems

Requirements engineering for inter-organizational systems are candidates for facing such a "blank sheet of paper" problem. The reason of the possible appearance of such a phenomena is, for instance, the high number of involved business partners, the diversity of business processes in the different enterprises, or simply the different business strategies of the involved companies. In fact, the latter one constitutes a problem since economic perspectives must be aligned within the collaborating partner network before developing the IT system. Requirements engineers or business analysts often "forget" to capture such aspects in an early requirements elicitation phase, since current techniques and methodologies do not foresee to include the economic parameters.

Research question

In order to avoid such a nightmare for requirements engineers when conducting a real-world business case, a comprehensive approach tailored for the requirements engineering of B2B systems is needed. Thus, the research question of this thesis can be formulated as followed:

- How can we provide a methodology that guides the business analyst through the requirements engineering phase towards the development of inter-organizational processes without disregarding the economic drivers of the B2B system?

Method vs. Methodology

Within this thesis we aim to answer this question and provide a comprehensive methodology that delivers well-defined artifacts for managing the requirements of B2B systems. At this point we need to clarify a nomenclature issue for being in accordance with other scientific approaches. The correct terminological use of the words *method* and *methodology* within scientific publications often results in long discussions between different researchers. Some of them do not see any difference between those two terms, while others see them differently. In fact, following the definition of the New International Dictionary of the English Language [126], a method "..is a process for attaining an object as a systematic procedure, technique, or mode of inquiry employed by or proper to a particular discipline or art". In contrary, a methodology is defined as: "..the analysis of the principles of methods, rules, and postulates employed by a discipline. It is a systematic study of methods that are, can be, or have been applied within a discipline". According to these definitions, the term methodology refers to a set of different methods. Within this thesis we propose a global comprehensive methodology that consists of different methods. However, some of these methods used in our approach are referred as methodologies in their specifications. As a result and due to simplification we use both terms synonymously within this thesis.

Before we further dwell on the development of our requirements engineering methodology, we briefly introduce the fundamental research areas of the proposed approach: *e-commerce* and *e-business*.

1.1 Introducing the research area

From WWW-Presence to Smart Business Networks

Reviewing the last two decades of doing business by using the vantages of the Internet, we are facing a development consisting of four phases [153]: i) *WWW-Presence*, ii) *e-commerce*, iii) *e-business Partnerships*, and iv) *Smart Business Networks*. The first one originates from the 90's where the aim of a company was to distribute information over the Web and to be present by means of a plain Web site. The second level (e-commerce) found its hype in the year 2000. At this time, companies started to offer commercial transactions for supplying the customers with their goods or services. It was the era of web shops, where users could experience a "24/7 self service shopping". The aim of the third level (e-business Partnerships) is to build platforms for buyers and sellers to reach the critical mass. Together with different partners, companies were able to offer enhanced services supported by Web 2.0 features [143] in order to address a widespread user group. Nowadays, the level of Smart Business Networks has been reached. It deals with the "informatization" of entire value chains from the initial supplier to the consumer. Since enterprise borders are blurred, Smart Business Networks are characterized by loosely coupled network configurations requiring the development of inter-organizational business processes [200].

E-commerce and e-business: some definitions

The topics addressed within this thesis correspond with the characteristics of phase 4 - the *Smart Business Networks*. However, the *e-commerce* as well as the *e-business* level provide the fundamentals for our methodology. Thus, we give some short definitions to establish a border between the two terms. Although they are often used synonymously, there are varying definitions. Computer scientists normally refer to the technical issues and building blocks - understanding e-commerce as applied computer science, whereas the management science or information system community follows a business and transaction view. On one side e-commerce refers to sharing business information, maintaining business relationships, and conducting business transactions by means of telecommunication networks [83]. This definition focuses on the coverage of all transaction phases (from the information over the negotiation to the settlement phase). On the other side, the US Census bureau defined e-commerce as a completed transaction (i.e. an agreement as transfer of ownership) over a computer mediated network, and e-business as any process that a business organization conducts over a computer mediated network (external and internal). This is similar to the definition of Papazoglou and Ribbers [147], with e-business including e-commerce but also covering internal processes such as production, inventory management, product development, risk management, finance, knowledge management and human resources. A similar view is provided by El Sawy [167] with the statement: "..it is important to note that e-business is much more than electronic commerce. E-business involves changing the way a traditional enterprise operates, the way its physical and electronic business processes are handled, and the way people work". E-commerce is viewed as the online exchange of goods, services, and/or money, whereas - on an upper level - e-business automates all business processes and integrates them with e-commerce applications to create one seamless, digital enterprise serving customers and partners.

Classifying e-commerce

Independent of these different views, one can classify e-commerce according to several criteria [122]: (i) participating actors, (ii) phases of a trading transaction, (iii) the monetary volume of a transaction, (iv) and the economic and technical layers of a transaction. For the purpose of understanding the scope of this thesis we only classify e-commerce according to the participating actors. Thereby, we use the so-called e-commerce ABC [205], which distinguishes between Administration (A), Business (B), and Consumers/Citizens (C). There may be relationships between all of them - e.g., Business-to-Business (B2B), Business-to-Consumer (B2C), Business-to-Administration (B2A), Consumer-to-Consumer (C2C), etc. Those relationships are depicted in Figure 1.1. As one may recognize from the title of this thesis, we concentrate on the Business-to-Business category. In the following section we motivate the approach and detail the peculiarity of B2B systems in regard to requirements management.

1.2 Motivating the approach

Service Oriented Architecture

As introduced in the previous section, Smart Business Networks are characterized by offering and consuming services within the partner

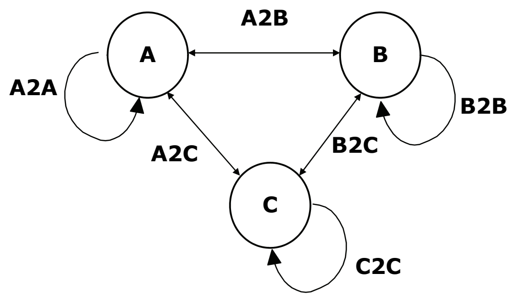


Figure 1.1
The e-commerce ABC

network. In fact, within the recent past, enterprises started to use service engineering concepts to build up a services portfolio representing their business. In this context services are economic activities offered to other business partners in order to achieve a certain benefit [216]. These services are intangible by nature and generated by business processes. Realizing the services portfolio in a technical sense results in B2B information systems according to the concept of a Service Oriented Architecture (SOA) [148].

The need for capturing the business processes from a global perspective

A successful B2B integration does not start with manually creating Web Services artifacts only, such as WSDL [208] or BPEL [131] code. Such an approach does not consider the business perspective from an economic point of view, the justification for the business processes, and the business requirements capturing the commitments and agreements between business partners. In addition, if business partners design their own interfaces in isolation, it is rather unlikely that their interfaces are complimentary to each other. Instead, an approach is needed capturing the business process from a global perspective. It is necessary to focus on the business perspective of a B2B process by gathering business domain knowledge and business requirements. These requirements must be transformed into a flow of service interactions between business partners. Such an approach is envisioned by the UN/CEFACT Modeling Methodology (UMM). We have been long-term contributors to UN/CEFACT and served as editing team for UMM's UML profile [188].

UN/CEFACT's Modeling Methodology

UN/CEFACT's Modeling Methodology (UMM) is an integrated approach for capturing the collaborative space between enterprises. It provides a modeling language and a methodology for accomplishing B2B projects. The methodology guides the analyst on his path from getting domain knowledge and requirements to designing business collaborations, executed between business service interfaces. UMM's modeling language is defined as a profile on top of the Unified Modeling Language (UML) [137]. A UML profile specifies a set of stereotypes, tagged values and constraints for customizing UML. This means, the general-purpose language UML is customized for the specific purpose of inter-organizational systems.

The Open-edi reference model

The UMM was developed according to the Open-edi reference model [78]. Open-edi distinguishes between the business operational view (BOV) and the functional service view (FSV). The BOV addresses the business aspects such as business information, business conventions, agreements and rules among organizations. The FSV is related to information technology aspects, which are necessary to support the execution of a business collaboration. Accordingly, the FSV implements the scenarios developed in the BOV. Within this thesis, our requirements engineering approach strictly focuses on the BOV layer of a B2B project. The interested reader is referred to the work of Zapletal [213], who proposed a derivation of executable artifacts from the BOV layer to the FSV layer.

Top-down vs. bottom-up methodology

However, if business analysts start to model the B2B processes by means of UMM at the BOV layer from scratch, it might lead to a business insensitivity, which implies disregarding the economic aspects that drive the business processes. In order to stay in business, companies must quickly adopt to faster and faster changing business conditions. Business models must reflect these changes, business processes must be designed supporting the value exchanges, and IT applications must adjust to changing company goals. Current business process-based requirements engineering approaches for developing inter-organizational systems focus too much on existing Web Services standards and, thus, on the technology layer. In such an approach the technology drives the business, and therefore it is referred as a bottom-up approach in this thesis. As an alternative we suggest a top-down methodology where the business/economic requirements drive the business processes. This methodology starts off with the business value perspective, leading to a trading partner perspective and resulting in a business process perspective. The latter one provides the specification for implementing the artifacts on the IT execution perspective. Thus, we can say that our approach delivers the requirements of a B2B system which is ready for implementation. We do not invent any new approaches on each of these layers, rather we outline how existing approaches are used, improved and combined into a business requirements driven approach to inter-organizational systems.

Business models are used to capture the economic drivers

As already mentioned, our top-down approach incorporates business models for capturing the economic drivers of the B2B processes. In former days business modeling was done by using standard process modeling methodologies such as UML's activity diagrams [135] or Petri Nets [125]. Since these modeling languages have been designed for modeling a sequence of activities, modelers tend to develop their business models in a workflow-oriented way. However, a major characteristic of business models is that they have no time-order. Therefore specific business modeling techniques have been introduced in order to capture the business perspective of an e-commerce information system. Presently there are four major and well-accepted business modeling ontologies - e^{3}value [48], Resource-Event-Agent (REA) [118], the Business Modeling Ontology (BMO)[144], and the Design and Engineering Methodology for Organizations (DEMO) [30].

The most prominent business modeling approaches

BMO focuses on the position of a specific business partner in the e-business network and how he can make profit. The fourth approach, DEMO, focuses on the development of a whole organization, including communication and production aspects and is therefore not applicable for our purpose. Due to the fact that BMO concentrates on the business semantics from an internal perspective, we take e^{3}value and REA as the ontologies of choice for our requirements engineering approach. e^3value is an ontology in order to depict a networked business idea. Within this network, business partners exchange things of economic value - no matter whether these things are tangible or not. The only important thing is that the value that is being exchanged has a certain monetary value for the business partner who is requesting the value. The REA (Resource-Event-Agent) ontology is an approach for gathering the rationale behind business collaborations. REA captures the declarative semantics of the collaborative space between enterprises from an economic viewpoint. It describes the involved actors (A), their resource exchanges (R) and holds the triggers for economic exchanges by the means of economic events (E). The integration of those two ontologies into one comprehensive business modeling based approach is one of the key contributions of this thesis.

1.3 The overall approach in a nutshell

The output of the approach is a UMM model

Within our business modeling based requirements engineering approach, we propose to start with e^3value for getting a first overview of the economic values exchanged in the network. Furthermore, e^3value offers the possibility to proof the economic sustainability of the business idea by quantifying the net value flow for each actor in the value web. Whereas e^3value concentrates more on the profitability of the IT system, an REA business model focuses on issues that may be relevant for the implementation and alignment of an IT system. Therefore we introduce conceptual rules for mapping an e^3value model to an REA model. Once, the business models are specified by e^3value and REA, we provide transformation rules to the business process perspective. The output of our methodological approach is the specification of the inter-organizational processes by means of UMM.

Worksheets ease the requirements engineering process

An advantage of our methodological requirements engineering approach is the fact that it puts the delivered models in a very strict corset. The resulting artifacts are well defined. Each artifact is restricted to a number of precisely defined modeling elements (stereotypes) and the relationships among them are also fixed by a set of rules. As a consequence, it is easier for software engineers to act upon the resulting artifacts in order to bind their local systems to the public process defined by UMM. However, it is rather hard for business experts to participate in the development of the whole B2B model, which includes different modeling notations, constraints and elements. They usually do not have any modeling knowledge and are not able to produce artifacts according to the constraints of our proposed approach. Communication with business experts is often based on plain text descriptions and less formal drawings. Today, a business analyst already uses some predefined templates - called worksheets - in order to gather information from the business experts. However, these worksheets are loosely connected to the business models as well as to the business process models. Accordingly, the business analyst has to connect the dots - by combining information spread all over the worksheets - in order to define the requirements for a B2B model. In this thesis we suggest a better alignment of worksheets and the models delivered during the requirements elicitation phase. In fact, the worksheets must represent an equivalent of the modeling elements, stereotypes and tagged values of the whole B2B model.

Since the combination of the different modeling methodologies by one comprehensive approach is not the only one contribution of this thesis, the next section summarizes the additional scientific contributions which were necessary for developing the approach.

1.4 Contribution of this thesis

Hypotheses of the thesis

As mentioned before, the goal of this thesis is to deliver an unambiguous requirements specification for B2B interactions. In order to reach this goals we specify two hypotheses, which will be validated at the end of this thesis. The following two hypotheses have been defined:

1. Using our business modeling approach for designing B2B processes helps the business analyst i) to design business processes from an economic point of view to ensure economic sustainability, ii) to semi-automatically generate process artifacts from business domain knowledge, and iii) to quickly adapt the B2B processes to changing requirements without the need to change the overall architecture.
2. A formalization of our approach improves the usability for the development of B2B processes i) by the definition of a unified process based on phases and iterations leading to a formalized and unambiguous requirements specification, ii) by the specification of well-defined transformation rules between the different methodologies, and iii) by the definition of worksheets for capturing and interlinking the domain knowledge.

In the following, those hypotheses are further on broken down into five contributions (c.f. Figure 1.2) that constitute the proposed modeling approach.

The roles participating in our appraoch

In order to understand Figure 1.2 and to get an overview of the "users" of our approach we shortly introduce the different user roles that are addressed in this thesis. One of the main actors within our approach is the *business analyst*. According to [150], a business analyst is an internal consultancy role that has responsibility for investigating business systems, identifying options for improving business systems and bridging the needs of the business with the use of IT. In the context of this thesis, the business analyst has modeling skills and is therefore sometimes referred as a *modeler*. A *requirements engineer* is similar to a business analyst. However, if we explicitly mention the role of a *requirements engineer* we refer to special requirements engineering skills as defined by Paech [146]. In contrary, the *business expert* has no modeling skills, but provides the business analyst with domain knowledge. Therefore, this role is also referred as *domain expert* within this thesis. Finally, the *IT expert* is restricted to technical skills (e.g, Java programmer). The IT expert can also be

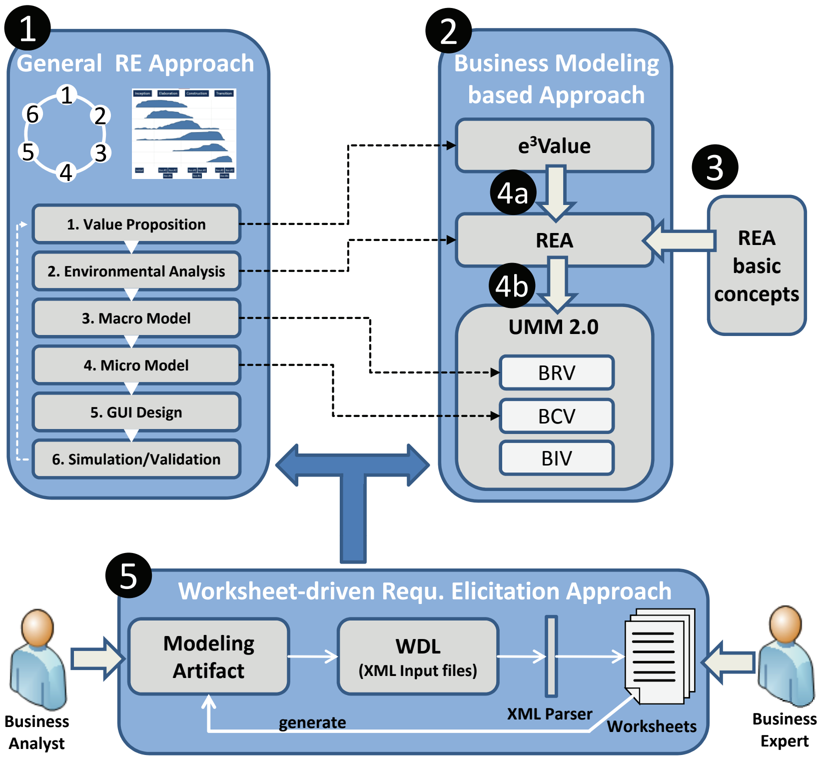


Figure 1.2 Overview of the contributions of this thesis

seen as the *software engineer* who develops the IT artifacts that are specified by our requirements engineering based approach.

Problem 1: Missing process-based requirements engineering approaches

Service-oriented architectures (SOA) aim at the alignment of business and IT by having a clear business process-centric focus. In order to reach that goal, real-world business processes are captured by business process models. These models serve as the basis for the declarative configuration of a SOA using appropriate deployment artifacts - i.e., XML-based process languages. Consequently, requirements engineering for SOAs must focus on business processes and on their integration into systems using interoperable services, which is not the case for most conventional requirements engineering approaches.

Contribution 1: A light-weight modeling approach based on phases and iterations

In this thesis we present a requirements engineering approach specifically designed for the engineering of SOAs. Requirements are captured using a unified process, based on phases and iterations leading to a formalized and unambiguous requirements specification. The final requirements specification can be used in succeeding development phases - i.e. for the model-driven generation of deployment artifacts for SOAs. The presented solution is called *general requirements engineering approach* since it delivers the fundamentals for the business modeling based approach delivered by contribution 2. In contrary to the business modeling based approach, the general requirements engineering approach is a rather light-weight methodology incorporating not so rigorous constraints and modeling rules. As denoted by the arrows from the left hand side to the right hand side in Figure 1.2, the general requirements engineering approach served as an input for the development of the business modeling approach in order to integrate more sophisticated and well-established methodologies. Due to simplification, the general requirements engineering approach is shortly referred as the *general approach* within this thesis.

Problem 2: Disregarding the economic drivers when designing B2B systems

Inter-organizational B2B systems are most likely tending to change their business requirements over time - e.g. establishing new partnerships or changing existing ones. If those business requirements are not connected to the underlying business processes, changes on both levels can not be traced back easily. The problem is that business analysts rather concentrate on the definition of the sequence and structure of the inter-organizational business processes than on the economic drivers of the partner network. If they build the business processes from scratch, they disregard important economic aspects, such as the value perspective or the trading partner perspective. Both perspectives are rarely used in traditional requirements engineering approaches.

Contribution 2: A business modeling approach from e^3value and REA to UMM

We introduce a modeling approach that considers both, the business perspective in terms of business models and the process perspective in terms of business process models. Thereby, we combine three different well-established methodologies in one comprehensive approach. The approach is called *business modeling approach*, since we start of with the two business modeling ontologies e^3value and REA that deliver the business models. Both ontologies applied in combination provide an excellent basis for developing the business process model by UMM. The output of this contribution is the definition of the conceptual mapping rules which are formalized further on by contribution 4.

Problem 3: Shortcomings of REA

The REA ontology assumes that, in the presence of money and available prices, all multi-party collaborations may be decomposed into a set of corresponding binary collaborations. The e^3value methodology illustrates e-commerce supply chains along similar lines with its graphical modeling tool [47]. In both modeling ontologies, requited economic exchanges are limited to instances between just two trading partners, even in the cases where buyers and sellers are aided in their e-commerce dealings by third parties like banks, logistics providers, or taxing authorities. The difference is that e^3value provides a notation in order to clearly depict the whole partner network constellation within its graphical modeling tool, whereas an REA model explodes geometrically in complexity as the number of trading partners grows. This results in a high number of unidentifiable relations between agents and events.

Contribution 3: Aligning REA for modeling multi-party collaborations

In order to overcome the limitations on REA we provide improvements that decreases the complexity of an REA model in a multi-party collaboration. This goal is achieved by slightly changing the original meta model. This change enables the reuse of events for multiple agents and ensures that all association between agents and events can be uniquely identified.

Problem 4: Lack of formalized mapping rules

The conceptual mapping rules between the different interlinked methodologies provide guidelines for the business analyst on how to produce models that are compliant with our approach. However, they do not support an automatic transformation of artifacts delivered by each methodology in regard to a Model Driven Engineering (MDE) approach. For instance, e^3value consists of modeling elements that map one-to-one to concepts of REA. If the business analyst only follows the conceptual mapping guidelines, he has to model all these concepts for each perspective by hand.

Contribution 4: Transformation rules for supporting Model Driven Engineering

In order to support an automated mapping between different models delivered by our approach, we formalize the conceptual mapping rules by using the ATLAS Transformation Language (ATL). ATL provides a model-to-model transformation engine which is able to transform any given source model to a specific target model. In our case, we perform a transformation from the e^3value methodology to the REA ontology and further on to a UMM compliant business process model. In order to demonstrate the transformation we use the Eclipse Modeling Framework (EMF) [182]. It is not said, that we are able to produce fully-fledged output models by our transformation rules. However, we deliver the backbones of each model as a starting point for further manual refinements.

Problem 5: Missing guidelines for conducting the requirements elicitation

As outlined in the beginning of this thesis, it is dangerous that business analysts and requirements engineers face the "blank sheet of paper" problem - especially in the area of B2B. The requirements are gathered during interviews and workshops together with the business experts in an early phase. They provide the basis for the business models and business process models delivered in later steps of the project. If there is no alignment between the gathered information and the modeling artifacts, the business analyst may either run into duplicated effort or fuzzy specifications. Thus, there are missing guidelines for conducting the requirements elicitation for B2B systems. Furthermore, the captured requirements are often represented one-to-one by modeling elements within our approach. Separating this information from the model itself is not efficient and may lead to inconsistencies between different models.

Contribution 5: A worksheet-driven requirements elicitation approach guides the business analyst

In this thesis we provide guidelines on how to integrate requirements elicitation into the development of business models and business process models. In fact, these guidelines represent an equivalent of the business modeling approach (as outlined by contribution 2) in text-based templates called worksheets. Thereby, we suggest forms for each group of modeling elements delivered by our approach. However, these forms do not follow a rigorous structure, since their layout can be configured by using an XML-based definition language. This helps the business analyst for being flexible in regard to different domains under development. The information captured by these worksheets does not only provide input for the business analyst. It is also well-understood by business people without any modeling skills, since the information is specified in natural language. Another advantage of our worksheet-driven approach is the ability to generate modeling artifacts out of text-based descriptions. This is possible, since the information between different worksheets is strongly interconnected to avoid inconsistencies. All this information is consolidated by an interactive tool-integrated worksheet editor, which is a prototypical implementation of this approach.

1.5 The methodological approach

Design Science forms the methodological approach

In order to incorporate all these five contributions, we follow a methodological approach. As we will show in our related work section of Chapter 2, requirements engineering spans over multiple research areas of *Computer Science (CS)* and *Information Science (IS)*. Following Denning and Freeman [29], Computer Science has established its "own" paradigm, crossing the traditional paradigms of math (logical / formal proofs), science (from hypothesis to experiments and validation) and engineering (from formal statements to design, implementation and test). This view comes along with one in *Information Systems (IS)* [60] with the discussion on *Design Science*, having its roots in engineering. However, the authors also put this into relationship to the science approach, with its distinct methodological steps, as the area of IS research is at the confluence of people, organizations, and technology [94]. Thus, it is not only about design, but also observing and validating the "implemented" changes. In this context one may define IS as the science dealing with information and related processes in organizations.

Design Science guidelines for ensuring a "proper" approach

In Design Science, there is a distinction between *design processes* (i.e., build and evaluate) and *design artifacts*, which are defined as constructs (vocabulary and symbols), *models* (abstractions and representations), *methods* (algorithms and practices), and *instantiations* (implemented / prototype systems). Based on these procedural and constructional "concepts", design science guidelines can be identified. Hevner et al. [60] defined seven guidelines for Design Science in Information Systems research:

1. **Design as an Artifact.** Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
2. **Problem Relevance.** The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3. **Design Evaluation.** The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
4. **Research Contributions.** Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
5. **Research Rigor.** Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
6. **Design as a Search Process.** The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7. **Communication of Research.** Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Within this thesis, we follow these seven guidelines to ensure that we specify a "proper" approach as it is generally accepted in the field of Design Science. As a proof-of-concept, in Chapter 8 we evaluate and discuss whether these guidelines have been fulfilled during the development of the approach.

1.6 Structure of this thesis

This thesis is a result of several years of research in business models, business process models, and their standardization by UN/CEFACT. The foundations of each of the different chapters have been published at international conferences and in international journals. In this thesis we provide a consolidated summary of our various contributions. For a complete reference of all relevant publications please see the publication list at the end of this thesis. The chapters of this thesis are organized in a self-contained manner. However, since a certain knowledge of the domain is generally required, it is recommended to read chapter by chapter in an ascending order. Furthermore, we need to stress that the contributions 1-5 are not strictly discussed chapter-wise. Some of them are consolidated by one chapter and some of them are treated by different ones.

Chapter 2: Related work

The thesis starts off with an overview of the state-of-the-art technologies that are relevant for the scope of this thesis. Thereby, Chapter 2 concentrates on scientific fundamentals, similar frameworks and methodologies, or simply research that coined the approach provided by this thesis.

Chapter 3: Accompanying example

Our requirements engineering approach has been applied to a real-world business scenario within the print media domain. One of the most prominent newspaper publishers in Austria served as an industrial partner to provide valuable feedback on the methodology. Chapter 3 gives an overview on the use-case scenario that is accompanying this thesis.

Chapter 4: General RE approach in a SOA context

In order to start with a "light-weight" methodology for the requirements management of B2B systems, we introduce a general approach based on phases and iterations. Chapter 4 consists of a step-by-step guide to support the business analyst during the requirements engineering phase. By using simple word-processing tools and fundamental business process modeling techniques, the general approach is easily understood by different stakeholders.

Chapter 5: Languages for engineering inter-organizational systems

As already mentioned, the business modeling approach consists of different modeling languages. In Chapter 5 we describe these languages in detail and provide the backgrounds why they are incorporated into our approach. Furthermore, we discuss the limitations of the methodologies and provide solutions toward their alignment for inter-organizational requirements engineering.

Chapter 6: Business modeling based approach

In order to describe our business modeling based approach, Chapter 6 demonstrates how e^{3}value, REA, and UMM are interconnected between each other. We specify the conceptual mapping between these methodologies and provide formalized transformation rules supporting a Model Driven Engineering approach.

Chapter 7: Worksheet driven approach

In Chapter 7, we introduce one of the key contributions of this thesis - the worksheet-driven approach. The chapter starts with demonstrating the shortcomings of traditional worksheets as already used in UMM 1.0. In order to overcome those limitations we provide a worksheet-driven guide for the business analyst that supports him during the requirements elicitation. We also introduce an XML-based Worksheet Definition Language (WDL) to integrate worksheets into any tool environment. Finally, we demonstrate, how worksheets can be used to generate modeling artifacts.

Chapter 8: First evaluation

Since the approach has been applied to the print media domain, we invited representatives of the industrial partner as well as business analysts of a business consulting company to participate in a first evaluation of the methodology. Chapter 8 evaluates whether the approach is i) in accordance with Design Science guidelines [60], ii) applicable in regard to its usability [38], and iii) valuable for the problem domain. Furthermore, we investigate the hypotheses made in the beginning of this thesis:

1. Using our business modeling approach for designing B2B processes helps the business analyst i) to design business processes from an economic point of view to ensure economic sustainability, ii) to semi-automatically generate process artifacts from business domain knowledge, and iii) to quickly adapt the B2B processes to changing requirements without the need to change the overall architecture.
2. A formalization of our approach improves the usability for the development of B2B processes i) by the definition of a unified process based on phases and iterations leading to a formalized and unambiguous requirements specification, ii) by the specification of well-defined transformation rules between the different methodologies, and iii) by the definition of worksheets for capturing and interlinking the domain knowledge.

Chapter 9: Conclusion and open issues

Finally, we conclude the thesis in Chapter 9 by providing summarizing remarks and an overview of open research issues, which are not addressed by this thesis.

2 Related Work

In this chapter we discuss the state of the art and the related work in regard to our proposed requirements engineering approach for developing B2B systems. In the area of B2B, a lot of research has been done to investigate the different methodologies and techniques [34, 199, 39, 175]. In [171], we discussed the most important technologies which are relevant for a model-driven approach towards the development of B2B IT solutions. We took this study as a basis for our evaluation within this chapter and extended its findings by additional approaches, frameworks and methodologies that are relevant in the scope of this thesis. Figure 2.1 provides an overview of the different approaches that have been identified during the literature review. The research fields that are addressed by our approach can be separated into four areas: (i) *Requirements Engineering*, (ii) *Business Models*, (iii) *Business Process Models* and (iv) *Related Approaches and Frameworks* that are well-established in the overall research context, but cannot be unambiguously assigned to any of the other three categories. As shown in Figure 2.1, the research field requirements engineering spans over the other areas. This is due to the fact that we use requirements engineering fundamentals to realize our approach. The bullets listed in each category reflect on one hand the authors that provide the scientific fundamentals and definitions, and on the other hand the approaches that have similar lines with our topic.

The different research fields related to the scope of the thesis

The remainder of this Chapter is structured as follows: Section 2.1 gives an overview of the requirements engineering techniques that are relevant for our approach. In Section 2.2 we discuss the most prominent business modeling ontologies. Section 2.3 covers the stat-of-the art modeling languages in the area of business process modeling. Finally, Section 2.4 comprises the related approaches and frameworks that cannot be unambiguously assigned to any of the other three sections.

2.1 Requirements Engineering

In our approach we use *Requirements Engineering (RE)* techniques to gather the knowledge which is necessary toward the implementation of a B2B system. However, before discussing the most prominent approaches in this area, we start with some definitions to examine the importance of requirements engineering in software and system design. Zave [215] defines requirements engineering as the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. Requirements

Fundamental definition of Requirements Engineering provided by Zave

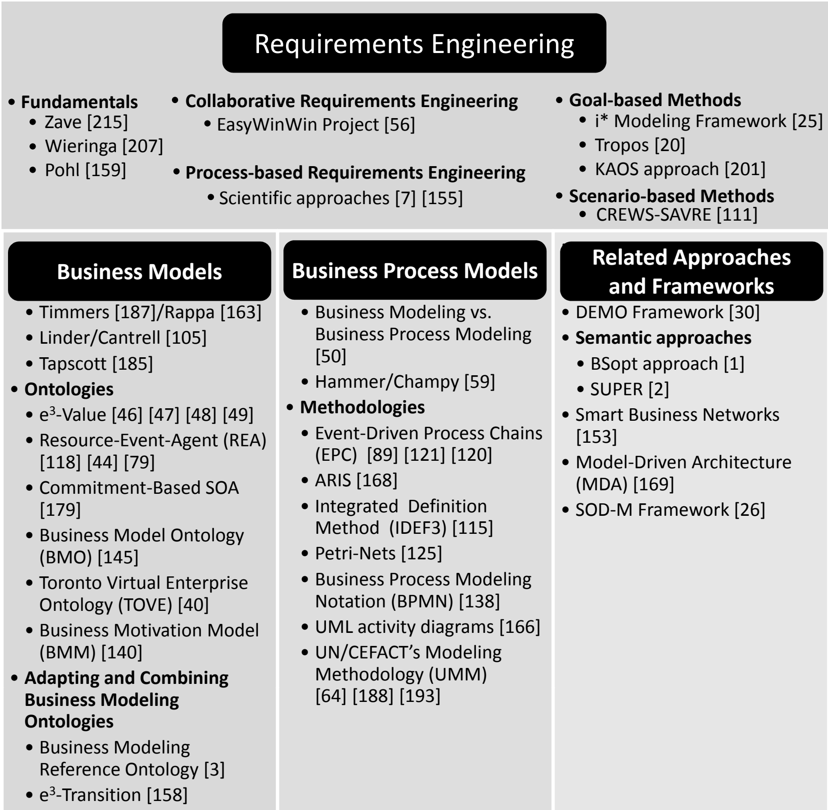


Figure 2.1 Overview of the related work

engineering is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.

Wieringa: RE characterizes the system under consideration of its development life cycle

The definition of Zave is restricted to software engineering. Nuseibeh and Easterbrook criticize this definition, since software cannot function in isolation from the system in which it is embedded, and hence requirements engineering has to encompass a systems level view [128]. Thus, the authors prefer to characterize requirements engineering as a branch of systems engineering [183], whose ultimate goal is to deliver some systems behavior to its stakeholders. In [207], Wieringa defines that this task is only possible if the requirements engineer is able to understand and characterize the system under consideration of the development life cycle. These definitions confirm our decision on using REA and UMM to capture the requirements of a dynamically changing B2B system. REA has a strong focus on the implementation of the whole system - not only on a particular software part and UMM considers the life cycle of entities relevant for the choreography of the business processes (c.f. Chapter 5).

Definition of Requirements Engineering provided by Shaw and Pohl

More general definitions of requirements engineering are mostly found in text-books observing the requirements engineering process from an economic perspective - i.e. to create cost-effective and sustainable solutions to practical problems by applying scientific knowledge [176] [159]. Since we use techniques such as e^{3}value (described in Chapter 5) in our approach that deal with the economic sustainability of the B2B system under development, this definition is of high relevance for our approach.

RE means interacting with humans

Usually, requirements engineering takes place in human activity systems, whereby the problem owners are people. Thus, requirements engineers need to be sensitive to how people understand the IT system and how the introduction of new solutions may affect their daily work. In [128], the authors state that requirements engineering is based on cognitive and social sciences to provide practical techniques for eliciting and modeling requirements. They propose four areas which should be part of the skills [146] of a requirements engineer: cognitive psychology [161], anthropology [80], sociology [67], and linguistics [22]. Our approach is based on worksheets, which incorporate aspects of these different areas. In fact, worksheets are predefined forms to guide the business analyst through the different phases of the requirements engineering process. Those worksheets are used during interviews between the business analyst and the domain expert. It is a pre-requisite that the structure of the worksheets consider those theoretical groundings mentioned above to support the business analyst most effectively during the requirements elicitation.

Having heard a lot about the theoretical backgrounds and definitions of requirements engineering, we now focus on practical approaches and applications that are related to our topics addressed in this thesis. Thereby, we strongly focus on approaches to business modeling and business process modeling techniques as used by our approach.

Collaborative requirements engineering

A first group of requirements engineering methods are *collaborative requirements engineering approaches*. In this category fall several approaches that deal with requirements elicitation under consideration of a collaborative partner network. An overview about the combination of different collaborative requirements engineering approaches and their dynamic selection depending on the project context has been presented in [123]. The survey summarizes techniques having a strong focus on the stakeholder's involvement in the requirements engineering process. Another analysis of collaborative requirements approaches as proposed by our general approach (c.f. Chapter 4) is given in [5] and [53]. The first one proposes an integrated model in order to create a graphical representation of an analysis model in an early design deliberation phase. The latter describes a method on how to filter out the necessary information from collaborative workshops with stakeholders in order to elicit the requirements. In contrary to our approach, both proposals cover only the requirements engineering techniques tailored to the use in an early development stage, in order to get a first sketch of the IT system to be designed.

A UML-based approach for collaborative requirements engineering

Gruenbacher [57] proposed a comprehensive methodology for collaborative requirements engineering. The approach is based on a UML meta model which is used to capture the different viewpoints of stakeholders. Furthermore, the author uses the meta model to investigate different tools in regard to their requirements engineering capabilities. However, since our approach uses UML concepts as well, there are some overlaps between these approaches. The approach of Gruenbacher has been applied to the *EasyWinWin* project [56]. EasyWinWin is a requirements definition methodology that builds on the win-win negotiation approach and leverages collaborative technology to improve the involvement and interaction of key stakeholders. With EasyWinWin, stakeholders move through a step-by-step win-win negotiation where they collect, elaborate, and prioritize their requirements. The result of this collaborative requirements engineering approach should be mutually satisfactory agreements between the different stakeholders [15].

From classical requirements engineering to process-based requirements engineering

A thorough examination of the gap between classical requirements engineering approaches and process based requirements engineering has been made by Arao et al. [7]. In their paper the authors provide a new requirements information model and requirements engineering process. However, the authors are missing a formalized process model allowing a model driven approach towards software artifact generation. A requirements engineering approach which focuses on the visualization of requirements has been presented by Pichler et al. [155]. Thereby the authors introduce a business process based requirements engineering approach and evaluate the tool integration of their approach. In contrast to this approach, our general approach proposed in Chapter 4 comprises a formalized process based on phases and iterations. Furthermore, the worksheet-driven approach (c.f. Chapter 7) delivers a final requirements specification which is tailored for a model-driven generation of deployment artifacts for SOAs.

Goal-based methods for requirements engineering - the i framework*

Finally, we focus on model-driven techniques that incorporate goal-based methods and scenario-based methods. The most prominent approach in the group of goal-based requirements engineering approaches is the *i* modeling framework* [25]. The framework was developed for modeling and reasoning about organizational environments and their information systems [211]. It provides a graphical notation that comprises concepts for modeling business objectives in terms of goals and softgoals. We integrate similar concepts into our general requirements engineering approach for B2B systems. For instance, the first phase of this methodology covers the value proposition by capturing goals and non-goals of the solution under development.

Extensions of the i framework*

The core concepts of the i* frameworks have been adapted or extended to fulfill special needs in the area of model-driven engineering. In [109], the authors use the i* notation as a basis for business modeling toward a service oriented design. Thereby, the approach takes advantage of i*'s agent orientation for modeling service relationships, and its goal orientation to facilitate adaptation from generic patterns to specific needs. This approach has similar lines with our business modeling based approach (c.f. Chapter 6). However, in contrary to our approach the authors focus on the reusage of business service patterns by providing a reference catalog and do not integrate business process modeling techniques. Another extension of the i* framework is the *Tropos* design process [20]. The goal of this approach is to provide an agent-oriented software development methodology which spans the software development process from early requirements to implementation for agent oriented software. The methodology is split into five main development phases: early requirements, late requirements, architectural design, detailed design and implementation.

The KAOS approach

The second well-accepted extension of the i* framework is the *KAOS approach* [201]. It is similar to the Tropos methodology in regard to the modeling of functional- and non-functional requirements by means of goals. However, the objectives of the KAOS approach differ from the Tropos methodology since KAOS allows an evaluation of the "unhappy path" - e.g. by goal-based analysis of hazards and threats, by the evaluation of trustworthiness in requirements models, or by conflict detection and resolution, etc.

Scenario-based method for requirements engineering

A prominent scenario-based method for model-driven requirements engineering that is related to our approach is *CREWS-SAVRE* provided by Maiden [111]. The acronym *SAVRE* stands for *Scenarios for Acquiring and Validating Requirements*. The aim of this approach is to generate real-world scenarios in the early requirements engineering phase, which are validated against the initial requirements specification. The approach provides a software tool that is able to generate such scenarios out of a use case specification of different actions. Fundamentals of this scenario-based method are incorporated into the guidelines for validating a model delivered by our general approach (c.f. Section 4.3.6).

2.2 Business Models

Covering the why?, what?, and how? in e-business models

In our approach, we use business models to describe the economic drivers of B2B systems. In general, business models incorporating a network of different business partners cover the fundamental question: *what* is offered by whom to whom? In contrary, business process models concentrate on *how* these offerings are selected, negotiated, contracted and fulfilled operationally. To complete this excursus of e-business modeling aims, the business strategy can be considered as a level on top of business modeling and business process modeling and deals with the questions: *why* should a company invest into the introduction of a new B2B solution. In such a view, a business model links strategy with implementation. However, the component of business strategy is not integrated into our approach and is therefore not further investigated.

The history of business models

Before going into more details of the different business modeling techniques, we give a short overview about the history of business models and discuss some definitions. The growing popularity of the term business model is strongly interrelated with the Internet hype of the late 90s [110]. At one stroke companies were able to increase sales by offering products and services 24-7 and simultaneously decrease transaction and procurement costs. For this reason the term business model quickly got popular and was used by a broad community, ranging from business people to scientists [105].

Business model fundamentals

Paul Timmers, who is one of the originators of the term business model, defines a business model as an architecture for the product, services and information flows, including a description of the various business actors and their roles. Furthermore, it describes the potential benefits for the various business actors and the source of revenues [187]. Linder and Cantrell [105] share a similar point of view and define a business model as a company's core logic in order to create value by explaining how a company acts on the market and earns money. Additionally, they specify that a business model consists of distinct components which include and represent the essential business logic building blocks. These building blocks range from revenue models and value propositions to organizational structures and arrangements for trading relationships. The authors of [154] get one step further and formulate a hierarchical structure of distinct tiers of business logic spanning from business models over business process models down to information and communication systems. Therefore, a business model can be seen as a contextual link between business strategy, business process and ICT [145].

Categorization of business models

In the literature, business models are often categorized into different types. The number of categorization types vary from five as defined by Tapscott [185] to about 30 as defined by Rappa [163]. However, most of the authors that are classifying their business models base their classification scheme on two dimensions - functional integration and degree of innovation in case of Timmers [187], economic control and value integration by Tapscott [185], and the emphasis of buyers and sellers by Pigneur [156]. Since the diversity of business model classifications shows the inadequacy of a unique classification scheme, Pigneur proposed another approach [157]. In contrary to the two dimensional frameworks of Timmers, Tapscott, and Rappa, Pigneur suggests to use a multi-category approach. Thus, a single business model could be positioned in a web of many classification schemes. He identifies twelve principal dimensions for classifying business models; user role, interaction pattern, nature of the offerings, pricing system, the level of customization, economic control, level of security, level of value integration, value/cost offerings, scale of traffic, degree of innovation, and power of buyers and sellers.

Business modeling ontologies

These fundamental business model definitions lay the cornerstone for the development of business modeling ontologies in order to create business models that are commonly understandable by means of graphical notations and well-defined meta models. To get a better understanding of the purpose of an ontology, a fundamental definition is given by Gruber: "..an ontology is an explicit specification of a conceptualization" [55]. In the following we present some of the business modeling ontologies and methods that come along with this definition. Currently, the most prominent and well established business modeling ontologies are e^{3}value [48], Resource-Event-Agent (REA) [118], and the Business Model Ontology (BMO) [145]. All of them are based on formal and semantic methods. e^{3}value and REA are a core part of this thesis and are described in more detail in Chapter 5. Thus, we only strive those two methodologies in this related work section and provide only the main facts.

e^{3}value depicts the partner network and calculates the return on investment

The *e^{3}value* methodology has been developed to model a value web consisting of actors who create, exchange, and consume things of economic value such as money, physical goods, services, or capabilities. It is an ontology-based methodology for modeling and designing business models for business networks, incorporating concepts from requirements engineering and conceptual modeling [49]. e^3value is based on the principle of economic reciprocity meaning a "give-and-take"-approach between actors exchanging objects with an economic value - e.g. if a seller delivers goods to a buyer, he gets money in return for the goods. We use e^3value to depict the B2B partner network from an economical point of view in an early requirements engineering phase. Thereby, e^3value provides the return on investment (ROI) of the B2B system under development by so-called profitability sheets before designing the business processes [46]. A similar approach has been shown in [158] where the authors propose the so-called e^3transition approach. This approach aims for deriving a process model from a value model by incorporating a step-wise transition model. Another similar approach has been demonstrated by Andersson et al. [4] where the authors propose preliminary attempts to bridge e^3value models directly with UMM models. In contrary to our approach these proposals map directly from an e^3value model to a UML activity diagram disregarding the interim step of depicting the trading partner perspective by REA.

REA concentrates on economic issues toward an implementation of the B2B system

REA was introduced by McCarthy [118] and was extended to an ontology for knowledge-based enterprise models by him and Geerts [44]. The concepts of REA reflect business accounting where the needs of managing businesses through a technique called double-entry bookkeeping was formerly the standard of use. REA uses this technique with semantic models of economic exchanges and conversions. The acronym REA comes from the core concepts Resource, Event, and Agent. The intuition behind these core concepts is that every business transaction can be seen as an event where exactly two agents exchange resources. In our approach, we use REA to ensure that business processes beneath do not violate the domain rules, i.e. to fulfill the basic economic principle for every business transaction - the give-and-take convention, called economic reciprocity. The authors of [42] propos a similar approach. They introduce a transformation approach targeting the interoperability of business process models. The authors argue that in order to gain interoperability on a business process level using business process modeling standards such as UMM or ISO/IEC 15944 [79] both business partners have to use the same business process modeling technique. To overcome this problem they propose to use REA as a shared global knowledge base for transforming model instances from UMM to ISO/IEC 15944.

Commitment-based SOA

REA also uses the concept of *commitments* that are made between business partners to promise or obligate an economic event in the future. In [178], commitments are used to model service engagements between business partners. The approach is called commitment-based SOA (CSOA). In terms of CSOA, a commitment relates three parties: a *debtor* who is committed to a *creditor*, typically within the scope of an organizational *context*. The context may be an institution, a marketplace or a legal jurisdiction. The connectors of these components are so-called CSOA patterns that support key elements of service engagements. The approach comprises three types of patterns: transactional patterns, structural patterns, and contextual patterns. These patterns are specified as state machine diagrams whereby its state transitions are used to interlink the CSOA components.

The Business Model Ontology is limited to internal economic aspects of a single business partner

The *Business Model Ontology (BMO)* has been introduced by Osterwalder and Pigneur [145]. They define a business model as a conceptual tool containing a set of objects, concepts, and their relationships with the objective to express the business logic of a specific enterprise. Therefore, it considers which concepts and relationships allow a simplified description and representation of what value is provided to customers. Furthermore, it is important to foster how this is done and with which financial consequences. Thus, base their ontology on nine concepts categorized into four main pillars: Product, Customer Interface, Infrastructure Management, and Financial Aspects. Osterwalder splits the four pillars of the business model ontology into nine interrelated business model elements. While the four areas are a rough categorization, the nine elements are the core of the ontology. A detailed description of the business model building blocks can be found in [144]. In contrast to e^3value and REA, BMO focuses on the position of a specific business partner in the e-Business network and how he can make profit. This means, it depicts a business model for a specific partner which makes the ontology unsuitable for our inter-organizational approach.

The Toronto Virtual Enterprise (TOVE)

Besides these well established methodologies, other approaches and frameworks (e.g. the *Business Engineering Model* [14], the *Edinburgh Enterprise Ontology* [195], or the *Toronto Virtual Enterprise* [40]) exist. The Toronto Virtual Enterprise (TOVE) is a project for developing an ontological framework for enterprise modeling. The goal of TOVE is fourfold: First, it is used to create an ontology to make the enterprise model understandable for each agent in the distributed network. Second, it defines the semantics of each description. Third, it implements the semantics in a set of axioms, which will cover the "common-sense" questions about the enterprise. Fourth, it comprises a graphical representation for depicting the concepts. However, TOVE's background origins from the area of knowledge engineering and was developed for computer integrated manufacturing. Due to this very specific focus within a certain domain, the TOVE project differs from our approach distinctively. Another promising approach which has recently been introduced by the Object Management Group (OMG) is the *Business Motivation Model (BMM)* [140]. The BMM specification aims at supporting the development of business plans in a structured way including the identification of necessary elements and their relationships as well as motivational aspects.

Reference ontology for business models devloped by Andersson et al.

As one can see, the area of business modeling comprises several ontologies and frameworks. This fact causes the research on combining the different methodologies. Andersson et al took up the endeavor and developed a so-called reference ontology for business models [3]. The purpose of the reference ontology was to realize a horizontal mapping between the underlying ontologies e^3value, REA and BMO. However, the authors see the three ontologies as substitutes and propose a "super-set" ontology comprising the most valuable concepts of each methodology. In contrary, our approach foresees a sequential order of the business modeling ontologies in order to deliver economic requirements for developing the business processes.

2.3 Business Process Models

Business Modeling vs. Business Process Modeling

According to the description of the methods and ontologies to reach an agreement by means of business models in the previous section, we now discuss the related work of business process models. There is a significant difference between business modeling and business process modeling. Whereas business models have no time-ordering, a business process model shows the sequence of activities to be performed to reach a certain goal [50]. Thus, one of the main goals of our approach is to link the rather static business models to the dynamic representation of business process models. Two prominent approaches for linking business models and business process models have been introduced by Weigand et al [204] and Schmitt [170]. Both, define a methodological approach for using business models as a basis for deriving business process models.

Definition of Hammer and Champy

One of the most prominent definitions of a business process is given by Hammer and Champy [59]. They define a business process as a group of related activities that together create customer value. In the context of B2B, the trading partners within a business network can be seen as customers along the value chain. With the growing importance of business processes a lot of different approaches have been developed toward a graphical representation of those related activities. In the literature, there exists several surveys discussing the utilization as well as strengths and weaknesses of these approaches [108, 179, 104]. Within this sub-section we highlight the most important ones.

Event-Driven-Process-Chains (EPC) and ARIS

Event-Driven-Process-Chains (EPC) is a modeling language to represent business processes graphically. It is used for modeling, analyzing, and visualizing business processes in an enterprise. EPC is basically a directed graph connecting events and functions through control flows providing design abilities for parallelism. The small set of modeling elements provides an easily understandable model for business analysts as well as for the management. IT specialists can use it as a basis for software development. *Events* are passive elements and represent a changing state as a process proceeds. They can either be *start events* with external changes (trigger the start of a process), *internal events* with internal changes of states (changed by a process), or *end events* with an outcome of a process that ends the chain and has external impact. On the other hand *functions* represent activities or tasks of a business process and are therefore active elements. They are triggered by one or more events. In a process chain, events and functions have to alternate, therefore an event has to follow a function and vice-versa. EPC is utilized in the ARchitecture of Integrated Information Systems (ARIS) by Scheer [168] representing the central method for information systems design. Further derivatives of EPC extending its core concepts are shown in [89, 121, 120].

Integrated DEFinition Method 3 (IDEF3)

A business process modeling approach that has its origin in the late 70s is the *Integrated DEFinition Method 3 (IDEF3)* [115]. It is a scenario-driven process flow description that captures the knowledge about how a particular system works. IDEF3 provides two kind of models: the process flow description which captures the relationships between actions and the object state transaction description capturing the description of allowable states and conditions.

Petri-Net theory

A rather mathematical approach that is used for a graphical representation of processes is based on the *Petri-Net theory* [125]. In general, the Petri-Net theory is used to provide a basis for different business process modeling approaches. It is used to model business processes [198] as well as workflow systems [196]. In the beginning, Petri-Nets were criticized to be used only for modeling internal process flows. However, the authors of [97, 106, 197, 95] propose to use Petri-Nets to model the business processes between different organizations as well.

Business Process Modeling Notation (BPMN)

Another popular modeling language is the *Business Process Modeling Notation* (BPMN) which is standardized by the *Object Management Group* (OMG) [138]. BPMN is used to graphically represent business processes in a business process diagram (BPD) and was first released in 2004 by the *Business Process Management Initiative* (BPMI). It incorporates aspects of already advanced modeling notations (e.g. UML activity diagrams [166], IDEF [116], ebXML BPSS [129], RosettaNet [165], etc.). BPMN can either be used to describe internal business processes or collaborative B2B processes. The notation was developed in order to be understandable, and is intended to be used by the management, business analysts, and developers. As described in [162], BPMN can also be used to generate the *Business Process Execution Language* (BPEL). BPEL is based on XML and describes business processes connected through Web Services.

UML activity diagrams

In recent years, UML-based approaches have become very popular for modeling business processes. The *Activity Diagram* is part of the *Unified Modeling Language* (UML) and describes what is happening in a workflow through a sequence of actions [166]. It is mostly used for business process modeling, but can also be used for system modeling. It concentrates on a couple of graphical elements and support parallelism and alternative paths through the workflow.

UML profiles are used to create new methodologies

In general, UML provides the ability to tailor its meta model for specific modeling purposes by means of a UML profile. Several approaches using UML for business process modeling have been proposed [152, 202, 107, 88]. However, these approaches focus on the modeling of business processes internal to an organization. Other approaches use UML to visualize Web Services and their choreography [43] [186]. More advanced approaches provide a development process for inter-organizational business processes. These are either driven by existing private workflows [82] or they are driven by the inter-organizational requirements instead of the private ones [90]. Another approach for representing and managing inter-organizational business processes is proposed by Kim [86]. The author defines a

UML 1.x based approach for modeling collaborative processes as a flow of transactions. The aim of this approach is to create an ebXML compliant business process specification.

UN/CEFACT's Modeling Methodology (UMM)

Also the UN/CEFACT's Modeling Methodology (UMM), which presents one of the three interlinked core methodologies of this thesis, is defined as a UML profile [64] [188] that is used for modeling the global choreography of inter-organizational business processes. It captures business knowledge independent of the underlying implementation technology, like Web Services or ebXML. UMM is used to model the choreography and data exchange commitments to be agreed upon between partners. Thus, a UMM business collaboration model becomes a kind of contract that guides a business partnership. According to the Open-edi Reference Model [79], UMM is used to create inter-organizational business process models on the business operational view (BOV) and XML is used as key concept on the implementation layer - the functional service view (FSV). Since UMM stops at the BOV layer, a transformation to an IT solution on the FSV layer is required. In [65] we describe such a mapping from UMM to BPEL. Furthermore, Hofreiter et al. [62] define a mapping from UMM models to the Business Process Specification Schema (BPSS) which is the XML-based language for describing the choreography of a business collaboration within the ebXML framework.

2.4 Related approaches and frameworks

The Zachman framework uses a taxonomy for organizing architectural artifacts

The need for modeling the business essentials and business processes of a company, analyzing and validating the developed models, designing the IT architecture and implementing the information system by the use of *one* comprehensive approach is not particularly new. However, *Enterprise Modeling* attracted a lot of attention and new frameworks and methodologies were developed (e.g. *The Open Group Architecture Framework (TOGAF)* [54], *Sherwood Applied Business Security Architecture (SABSA)* [177], or the *reference architectural styles for service-oriented computing* [33]). A further well-known framework of this area is the *Zachman framework* [212]. The Zachman framework is an approach that helps enterprises to realize the transformation of an abstract idea into an instantiation by using a taxonomy for organizing architectural artifacts. The taxonomy is represented by a so-called *Zachman Framework Matrix*. The matrix defines the different participant's perspectives in building an enterprise architecture (visionary, owner, designer, builder, implementer, and worker) by its rows and the six basic interrogatives (what, how, where, who, when, and why) by its columns. Each intersection contains a unique model giving an integrated view of the enterprise which is being modeled. The framework does not restrict the business analyst to use a certain modeling notation to design the proposed models. Thus, the framework was rather developed for analytical purposes than for modeling concrete business process models. In summary, the approach is not tailored for the need of modeling inter-organizational systems. Due to this fact, the framework differs significantly from our approach. However, an interesting point is the clear fragmentation of responsibilities for each enterprise model within the matrix. Similar lines are drawn in our general approach by using the RACI matrix [19], which will be explained in Chapter 4.

Design and Engineering Methodology for Organizations (DEMO)

Critics about such comprehensive frameworks comes from Dietz. He states that most of the enterprise modeling techniques only provide a *black-box* view of the organization that is re-engineered [31]. Dietz defines a black-box model as a system with input and output variables that are connected through transfer functions [32]. If the transfer functions are not known, the system can only be investigated through *functional decomposition*. Thus, Dietz sees a black-box model as a very powerful mental tool for understanding organizations, but it is inappropriate for understanding and changing the internal operation of a system. Instead, a *white-box* model is needed which can be refined by *constructional composition*. This concept is provided by the *Design and Engineering Methodology for Organizations (DEMO)* [30]. The DEMO framework, which implements the *Language Action Perspective (LAP)* framework [28], aims at representing the essential structure of business processes within an organization. In comparison to our business modeling approach, the scope of this framework is much broader and focuses on the development of a whole organization, including communication and production aspects.

Semantic Web technologies are used for business process modeling

In recent years, the development of frameworks that use the concepts of the *Semantic Web* emerged significantly. Especially in the field of business modeling and business process modeling toward the implementation of a service-oriented architecture, the need of an integration of these technologies becomes apparent. The Semantic Web [13] helps to overcome the limitation that most information on the Internet is only understandable by humans, but not interpretable by machines [37]. Since current approaches lack of a structured annotation with meta data, Semantic Web languages must be integrated to make the information machine-processable. Semantic Web languages are, for example, the *Resource Description Framework (RDF)* [210] or the *Web Ontology Language (OWL)* [209].

The BSopt project represents a 3-layered top-down approach

As a first example of frameworks that integrate Semantic Web technologies, the *BSopt* project is discussed [1]. BSopt - *Business Semantics on top of process technology* is funded under the Semantic Systems Program of the Austrian Research Promotion Agency and led by the Vienna University of Technology. Since we are part of the project team, research results delivered by the project serve as valuable input for the approach presented in this thesis. Thus, there are overlaps between our requirements management approach for B2B processes and the BSopt project. A first similarity is that BSopt uses UMM as an essential technique to describe the semantics of business processes. It develops a methodology and a tool set for a top-down approach for SOA where the business requirements drive the underlying IT infrastructure implemented by Web Services. In other words, BSopt defines a methodology that considers business models on the upper layer, business process models on the middle layer, and deployment artifacts for a SOA on the bottom layer [70]. This top-down approach of BSopt is depicted in Figure 2.2. BSopt integrates different approaches on each of the three layers and specifies a meta model. A model created by the BSopt methodology must be in accordance to this underlying BSopt meta model. The meta model integrates all the semantic concepts on each layer and their interdependencies. Since business models and business process models are often specified by different notations the approach is defined by a conceptual BSopt meta model. In order to cope with the different requirements on each level, BSopt uses a flexible knowledge representation language. Thus, the conceptual meta model of BSopt is expressed in OWL. However, a comprehensive implementation of this approach by means of UML is given by Zapletal [213]. The author strongly focuses on the generation of executable code, but did not provide transformation rules between the business model layer and the business process model layer.

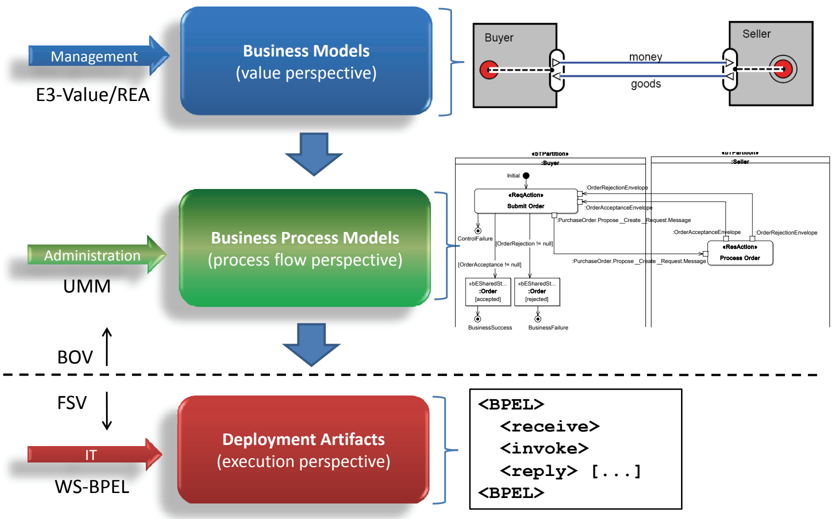


Figure 2.2 *The three-layer architecture of BSopt*

The SUPER project uses semantic annotations for its bottom-up approach

As a second approach that uses semantics for developing the business processes we discuss the EU funded project *SUPER* (IST-026850) [2]. The major objective of SUPER *(Semantics Utilized for Process Management within and between Enterprises)* is to raise business process management from the IT level to the business level. This objective can only be achieved, if business process management is accessible to business experts and business analysts without requiring detailed technical expertise. SUPER uses Semantic Web and particularly *Semantic Web Services* [119] in order to enable users to perform complex tasks without requiring an understanding of the underlying technology. SUPER aims at providing a framework that is context-aware based on Semantic Web Service technology, and which acquires, organizes, shares and uses the knowledge embedded in business processes and IT systems. Business experts and analysts can access this knowledge in an understandable format through a process modeling tool, which is tailored for the use of SUPER concepts. The tool enables them to easily analyze, change and create business processes, leading to a higher degree of agility in companies [17]. SUPER achieves this objective by adding semantic annotations to business process modeling artifacts (like process activities, services and execution artifacts), making these artifacts accessible for advanced querying and reasoning [112]. Using these querying and reasoning approaches, the tools developed in SUPER support users during business process modeling through techniques such as *Semantic Business Process Discovery* [113], *Semantic Business Process Composition* [203] and *Semantic Business Process Mediation* [127]. Figure 2.3 depicts the three essential parts of the SUPER framework. One of the deliverables of SUPER is the semantic business process repository which is depicted at the bottom layer of this figure. Business experts retrieve this repository by the help of semantic annotations and compose their executable processes semi-automatically (middle layer). Finally, semantic business process mediation services (top layer) enable the seamless integration of processes originating from different business partners in the collaboration network.

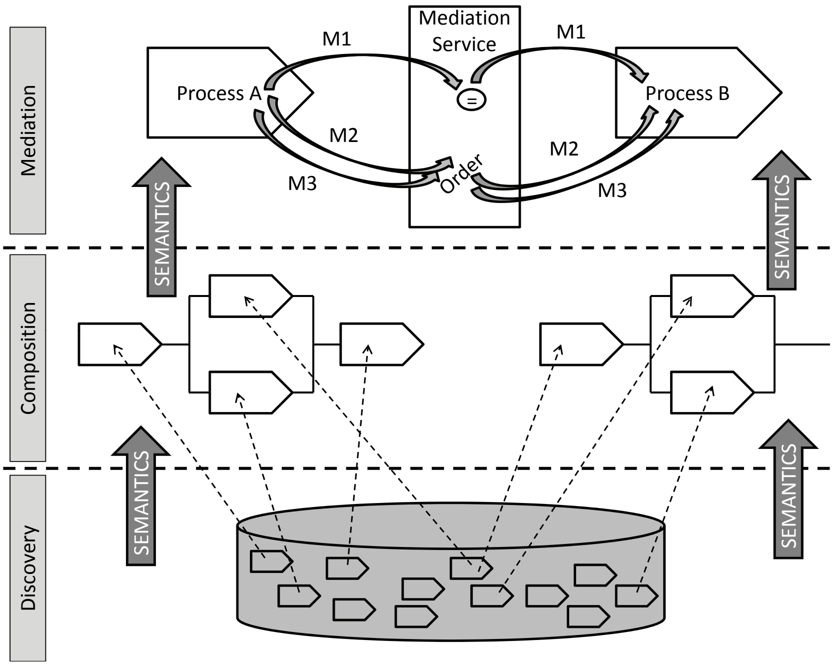


Figure 2.3
The essentials of the SUPER project

BSopt vs. SUPER

There are some overlaps as well as differences between the *BSopt* and the *SUPER* approach. As stated in the previous paragraph SUPER annotates business processes by the means of semantic concepts. The BSopt approach describes the semantics of business process models by means of business modeling techniques - e.g. e^{3}value or REA. The same concept is applied to the service implementation layer. Whereas in SUPER service compositions are semantically annotated, BSopt describes the IT layer by the business process modeling concepts of UN/CEFACT's Modeling Methodology (UMM). In contrast to SUPER, the BSopt approach considers the economic drivers of a business process and evaluates economic sustainability of the IT system to be designed.

Smart Business Networks strive for flexible inter-organizational partner networks

Most of the frameworks and approaches introduced so far consider a hard-wired business network once the participating partners are modeled by any kind of modeling language. However, business conditions and partner constellations change over time. Thus, an approach is needed that deals with the questions: how to build a business network to make the business processes more agile, with less pain and more return to all the members of the network, now and over time? The answer of this question lies in the design of *Smart Business Networks* [200]. In [153], Vervest et al give guidelines ensuring that the business processes and the partner network constellations are "smarter" than the ones of the competitors. Following the approach of Vervest et al, this goal is achieved by sharing process knowledge [11], adjusting business modularity [114], enabling of Web Services [151] and embedding business logic [149]. However, the overall message of building Smart Business Networks is to switch from static intra-organizational business constellations to flexible inter-organizational partner networks.

Model-Driven Architecture (MDA) and the SOD-M framework

A flexible design of inter-organizational networks can only be achieved, if the high-level business models have clear and well-defined interdependencies with the behavioral models and the service-oriented information system beneath. In our approach, this task is fulfilled by a *Model Driven Engineering (MDE)* approach (c.f. the mapping between business models and business process models in Chapter 6). MDE is an evolving and promising approach to software engineering [169] - especially by the OMG specification, the *Model Driven Architecture (MDA)* [136]. MDA provides an open, vendor-neutral approach for the alignment of the business view (*Computational Independent Models - CIM*) and the information systems view (*Platform Independent Models - PIM* and *Platform Specific Models - PSM*). An MDA approach describing the mapping between CIM and PIM models toward a service-oriented development of information systems is given by the *SOD-M* framework [26]. The main feature of the SOD-M (Service-Oriented Development Method) is a Domain Specific Language (DSL) [132] that represents the modeling of information systems from a service oriented perspective. The SOD-M framework has some similar lines with our approach, since the framework incorporates value models and business process models. However, a major difference is, that the framework aims at delivering service composition models by having a strong focus on the implementation layer instead of the requirements engineering process.

From an e^3value model to a SOA model by using the ATLAS Transformation Language (ATL)

In [27] the authors use the SOD-M framework to map from value models in terms of e^3value to an information system model. The latter one is represented as a use case diagram, whereby each use case in the diagram represents a service that interacts with the information system. The transformation is automated by using the *ATL (ATLAS Transformation Language)* [81] [8]. ATL is a model transformation language that provides the ability to specify transformation rules. These rules are used to define how source model elements are matched and navigated to create and initialize target model elements. We use ATL to map between the concepts of the different modeling methodologies.

3 The accompanying example: print media domain

As a proof-of-concept, a real life business scenario from the print media domain demonstrates the requirements engineering approach for B2B processes. The example used in this thesis has been taken from the use case scenario of the national funded IT project BSopt (Business Semantics on Top of Process Technology) to which we used parts of our requirements engineering approach [1].

The customer acquisition in the print media domain

In the print media domain, the success of a newspaper publisher is significantly based on the number of their readers. Not only, because the publisher makes revenue by each sold newspaper. A much bigger source of income is the revenue from the advertisement market. The more readers the newspaper publisher holds, the better is the quota for the advertisement market. This quota is specified by an independent market research institute which analyzes the customer fluctuation between newspaper publishers every year. For this reason the customer acquisition has been turned into a hot topic.

Lack of B2B supporting IT systems

In order to get new customers, a lot of different external business partners are involved (e.g. newspaper publisher, call center, address data provider, etc.). The problem is, that the information exchanges between those business partners is mostly done via paper-based documents or via manually controlled data exchanges supported by legacy systems. For example, the upload of the address data of possible candidates for test readers to an external call center is done via a daily not-automated FTP transfer. In other words, there is only little support by a sophisticated IT system. Therefore, it was the goal to replace the current methods with an efficient B2B solution. In order to implement a B2B model for the print media domain, it is important to consider economic aspects before developing the new solution. In this thesis, we introduce different approaches for specifying the requirements of inter-organizational IT systems by the example of the customer acquisition use case.

The remainder of this Chapter is structured as follows: Section 3.1 gives an overview of the problem domain. In Section 3.2 we explain the customer acquisition example in detail. Section 3.3 summarizes the facilitating assumptions that have been made in order to provide not too complex business processes for demonstration purposes.

3.1 The problem context of the business scenario

Test readers increase the quota of the newspaper publisher

In the print media domain customer fluctuation is mostly affected by competitors and their aggressive enticement of customers. The newspaper publisher under consideration has a permanent reader stock of 900.000 customers (bound to a one-year contract) paying a monthly subscription for getting their newspaper every day. It is almost impossible to increase the number of permanent readers, since the newspaper market is already saturated. Thus, one of the main tasks of the newspaper publisher's marketing and sales department is to keep this customer stock. If the competitors catch their customers with lower prices, special offers or attractive advertisement gifts (produced by *third-party vendors*), the *newspaper publisher* must acquire new *test readers* in order to keep this customer stock. E.g. if the newspaper publisher looses 50.000 permanent readers, they must acquire 200.000 new test readers to keep the stock of 900.000 readers and to compensate the customer turnover to other newspaper publishers.

Test readers are attracted by advertisement gifts

A test reader gets the same service as a *reader* having a permanent subscription but for a shorter time period and for a discounted fee. Additionally advertisement gifts should attract customers to become test readers (e.g. daily newspaper for 3 month and an iPod Shuffle for 70 EUR, whereas a one-year permanent subscription costs 240 EUR). The customer acquisition is either done in-house (e.g. by mail advertisement) or outsourced (in our case by a *call center*).

Depending on the available budget there are four channels for getting new test readers - Internet, post mail, face-to-face and telephone acquisition. Table 3.1 gives an overview about the different channels and their success quotes. The telephone channel is the most successful way for getting new customers - but the most expensive one as well, since it has been outsourced to an external company. The Internet acquisition (e.g. banner advertisements on a website, online lotteries) is the least successful one, but the cheapest one. Under face-to-face acquisition we understand the customer acquisition often done by freelancers or students (e.g. on exhibitions, sports events, or on public places). Therefore the newspaper publisher is aiming for finding the right mix of customer acquisition via a certain channel in respect to the available budget. Our business model driven approach helps to consider such factors for the definition of the partner network (see Section 5.1.2).

Channel	Acquired Readers	Test Readers	Quota	Costs
Telephone	100000	10000	0.1	++++
Postal Mail	100000	2000	0.02	+++
Internet	1000000	5000	0.005	+
Face-to-Face	1000	40	0.04	++

Table 3.1 *Success quotes for the acquisition of new customers via different channels*

As one can see, to achieve a successful customer acquisition, different partners are involved using different channels. It follows that the interaction between the different parties requires a clear and strict management of this business scenario. As mentioned before, the industrial project partner claimed, that the business processes as well as the economic partner network within their business scenario are defined too vague and the information exchange is mostly done by paper-based documents. The goal is to realize the entire customer acquisition management in an electronic manner supported by IT systems. The exchange of documents (e.g., address data of customers) should be achieved between the different participating partners in an automated manner and through standardized interfaces with as little human intervention as possible. Our approach as proposed in this thesis helps the newspaper publisher to gather the requirements for the development of such a B2B solution.

3.2 Managing the Customer Acquisition

In order to get an overview of the accompanying example, Figure 3.1 depicts the most important interactions between the newspaper publisher and the involved business partners.

Characteristics of the test reader and the reader

The reader has a permanent contract with the newspaper publisher and gets a daily newspaper for a certain yearly fee. In contrary, the test reader is acquired by the newspaper publisher or external companies and pays a non-recurring reduced fee to test the newspaper for a short period of time. At the end of this test period, the test reader should become a permanent reader with a fixed contract. From historical data we know that this applies to every fourth test reader. In order to attract test readers, the newspaper publisher offers a goody (advertising gift) in combination with a test subscription. A goody is a product that is sold by a third party vendor and ranges from multimedia equipment to household facilities and all kind of vouchers (e.g. tickets for sport events, trips, etc.). All these articles must be purchased by the third party vendor from different manufacturers. Due to the broad assortment of different articles, the management of these goodies is of importance. For example, the actual order process - in our context called *order from quote* - and the delivery process of the goodies along the supply chain comprises many non-automated intermediate steps which results in time-consuming efforts.

A test reader can be acquired by the newspaper publisher itself or by an external company. An external company is either a marketing partner or a call center. A marketing partner has a cooperation with the newspaper publisher and attracts new customers by using their own channels. For example, if another non-competing newspaper offers a combined subscription of both newspapers. A call center is an external company that acquires test readers via telephone. In both cases, the management of the customer acquisition is mostly driven by the exchange of address data sets. The address data used for the customer acquisition originates either from internal sources (e.g., former customers still existing in the CRM systems of the news-

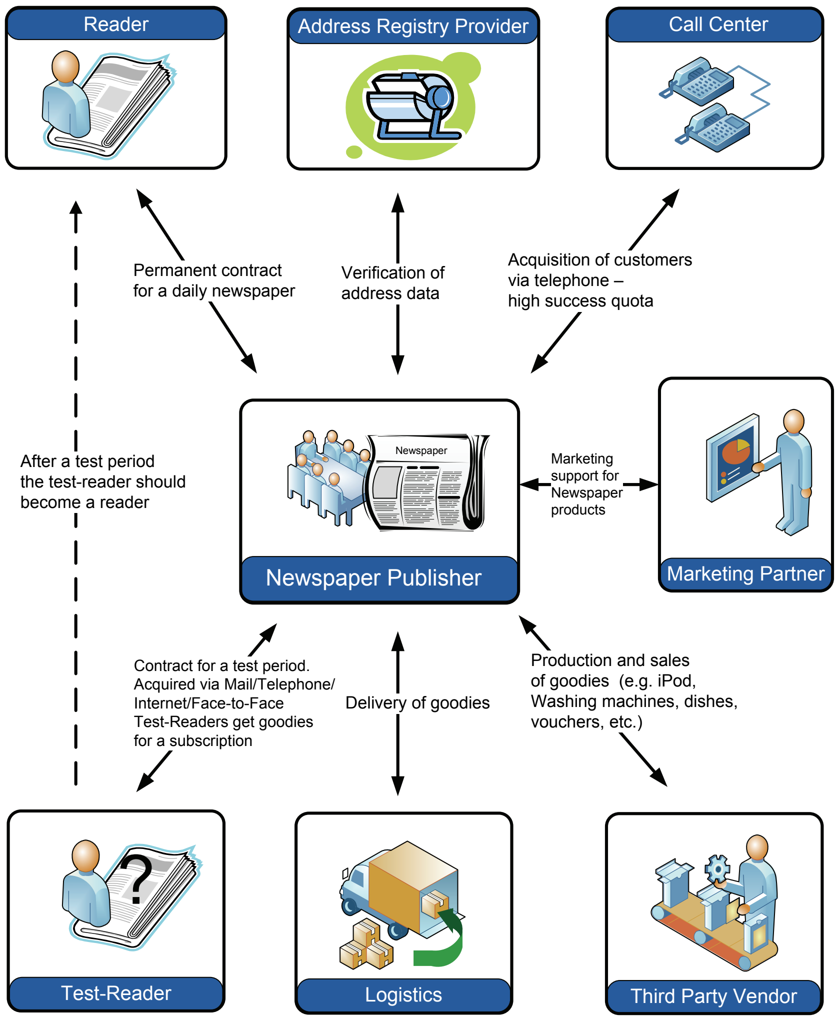


Figure 3.1 *Overview of the Use Case Scenario*

paper publisher) or from external sources (e.g. marketing companies collecting addresses from surveys or lotteries). Due to cost reduction and multiple acquisition, it is important that the address of a possible customer is still existing. In order to verify such address data, an address registry provider is used for validation.

3.3 Facilitating assumptions for demonstration purposes

Facilitating assumptions for demonstration purposes

In this thesis we concentrate on the management of the customer acquisition - i.e., the prerequisites that needs to be done in order to guarantee a flexible and economic sustainable process flow. At this point, we need to stress that we do not take the active customer acquisition as a central point - i.e., the interaction between the acquisition party and the customer.

Figure 3.1 gives a high-level overview of all the business partners involved into the whole business scenario. However, for demonstration purposes we make the following assumptions and simplifications:

1. we consider the telephone acquisition via an external call center as the only customer acquisition channel;
2. we neglect the logistics in all our business processes;
3. we concentrate on three main processes and tasks on the business process level:
   - purchasing of goodies (*request for quote* process)
   - management of address data for telephone acquisition (*call center interaction* process)
   - verification of address data (*address validation* process)

As depicted in the overview of the partner network in Figure 3.1, several industry partners were involved in this project. Due to the obligation to maintain confidentially, all monetary values used within the example of this thesis have been changed.

In Chapter 4 the customer acquisition scenario is further explained by a general approach using business process driven requirements engineering techniques.

4 General Requirements Engineering Approach in a SOA context

Service-oriented architectures (SOA) aim at the alignment of business and IT by having a clear business process-centric focus. In order to reach that goal, real-world business processes are captured by business process models. These models serve as the basis for the declarative configuration of a SOA using appropriate deployment artifacts - i.e., XML-based process languages. Consequently, requirements engineering for SOAs must focus on business processes and on their integration into systems using interoperable services, which is not the case for most conventional requirements engineering approaches. The general approach described in this chapter is a business process based requirements engineering methodology specifically designed for the engineering of SOAs [103]. Requirements are captured using a unified process, based on phases and iterations eventually leading to a formalized and unambiguous requirements specification. The final requirements specification can be used in succeeding development phases - i.e. for the model-driven generation of deployment artifacts for SOAs [65].

A requirements engineering approach based on phases and iterations

The general approach introduces a six phases, step-based, iterative solution for the precise definition of requirements. The methodology does not only specify the delivered artifacts for each phase, it also provides the methods and techniques that are used to gather the required information. The methodology is intended to be collaboratively used by all stakeholders involved in the development of a new solution for inter-organizational e-Business systems - such as business domain experts, business analysts, business process modelers, user interface designers, the management, and implementers. The general approach is further on used to provide a transition to the business modeling and worksheet driven approach described in Chapter 6 and 7. By doing this, we compare the phases of the general approach to the artifacts delivered by these approach.

The general approach is tool independent

Furthermore, this general approach itself is tool independent and can be implemented by any business process modeling tool of choice. However, in this chapter we provide tool recommendations for each delivered artifact of the different phases. Since most of the artifacts are based on the modeling languages UML and BPMN we propose to use the Enterprise Architect [181] as the main tool of choice for our approach.

In order to explain the general approach, the use case scenario of the customer acquisition has been taken as a proof-of-concept. In the highly volatile world of customer loyalty within the print media domain the presented approach enables faster application development and faster integration of solutions, thus leading to a competitive advantage over other market participants.

The remainder of this Chapter is structured as follows: Section 4.1 motivates the development and the purpose of the general approach. In Section 4.2 we give a short overview of the approach and its phases. Section 4.3 describes each phase of the approach by example. Section 4.5 concludes the approach with a final assessment and a transition to the business modeling based approach.

4.1 The motivation behind a general approach

Involvement of different stakeholders in the requirements engineering

The introduction of new inter-organizational interfaces in an enterprise contains a set of reoccurring challenges. In particular the requirements engineering process is error-prone and contains potential risks [93]. Before a solution is introduced the different stakeholders such as business domain experts, business analysts, user interface designers, developers, the management, etc must have a common understanding of the solution to-be. However, most of the stakeholders define their requirements using different concepts, languages, and tools. Thus, the final requirement specifications are often redundant and incompatible.

Requirements engineering is a dynamic process which does not consist of self contained phases and steps, but is rather executed in an iterative and repeating manner. Since professionals claim that most of the requirements engineering processes in companies are rather rigid and inflexible, changes in requirements can only be partially reflected or are not considered at all. Furthermore, when a new B2B solution is introduced in an existing system landscape there are a multitude of interdependencies to existing processes, systems, and other solutions. Most of the known requirements engineering approaches consider only the new solution architecture's requirements. Little to nothing is specified in regard to interdependencies to existing solutions and systems.

Alignment between business and IT

Finally the advent of the service orientation paradigm has brought additional challenges to the requirements engineering domain. Service oriented architectures (SOA) are expected to deliver a flexible alignment between business and IT. The goal of business/IT alignment is achieved by SOA services that realize business processes [109]. A business process is essentially a semi-formalization of business needs and requirements. Although these benefits are well known, a specialized requirements engineering process for the proper design of a SOA is still missing. Existing requirements engineering approaches do not completely bridge the gap between business and IT in order to reach a proper alignment.

By this general approach we provide a new technical solution aiming to overcome the limitations mentioned above. We introduce our business process based requirements engineering approach which consists of six distinctive phases that are constructed in iterations. In our approach we use the Business Process Modeling Notation (BPMN) [133], which is currently the state-of-the-art in both, industry and academia. BPMN provides a standardized mapping to the Business Process Execution Language (BPEL) [206]. BPEL [131] is a declarative process specification language used to configure execution engines accordingly. Consequently our approach provides the basis for the generation of SOA artifacts from the artifacts gathered in the requirements engineering phase.

4.2 The approach at a glance

The six phases of the approach

The general approach has been successfully applied for managing the B2B requirements of two industrial partners representing two different domains. The first one is an Austrian mobile network operator, and the second one is an Austrian newspaper publisher. Since the latter one is used as a proof-of-concept for all the artifacts delivered by the approach of this thesis, we use the newspaper publisher as a demonstrator for the general approach. As shown in Figure 4.1 the methodology consists of six distinctive phases: value proposition, environmental analysis, macro planning, micro planning, GUI design, and validation/simulation. Each phase has well defined objectives and delivers a set of artifacts and documents. At this point, we need to mention, that we do not use the last two phases (GUI design and simulation/validation) as an input for the worksheet driven approach in Chapter 7. Since it is not possible to capture those phases by the help of worksheets, artifacts delivered by those phases will not be mapped to our final methodology. However, to get an understanding of the complete lifecycle of the requirements management process for B2B systems, those phases are considered by detailed descriptions within this chapter.

The different phases of the methodology are not self contained but have strong interdependencies. Each phase has a distinct goal and its artifacts serve as input for the following phases:

1. *Value proposition* - Specify the value and the purpose of the B2B solution which should be introduced.
2. *Environmental analysis* - Identify the affected entities, side affected entities, affected projects and systems as well as organizational units involved in the solution.
3. *Macro planning* - Use the input of the two previous phases to construct a first coarse grained process model.
4. *Micro planning* - The coarse grained process model is gradually refined to the final process model.
5. *GUI design* - For each user interaction, a set of GUI mock-ups is assigned to the relevant process activities.
6. *Simulation & Validation* - The designed model is verified using process simulation concepts.

As shown in Figure 4.2 the six phases of the methodology are split into four steps, similar to the Rational Unified Process (RUP)

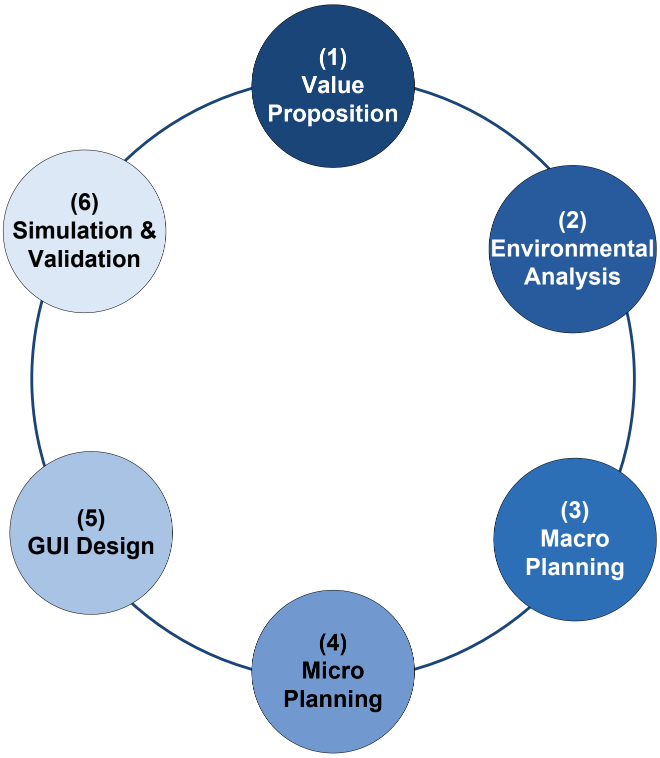


Figure 4.1
Overview of the six phases used in the general approach

[91]. Each step contains a set of iterations which are executed over and over again until the final artifacts per phase are finished. At any point within the lifecycle a return to a previous phase or step is possible. Depending on the different steps, each phase has either more or less significance throughout the overall construction of a B2B model. E.g. value proposition is important in the first two steps inception and elaboration but is of minor importance in the last two steps construction and transition.

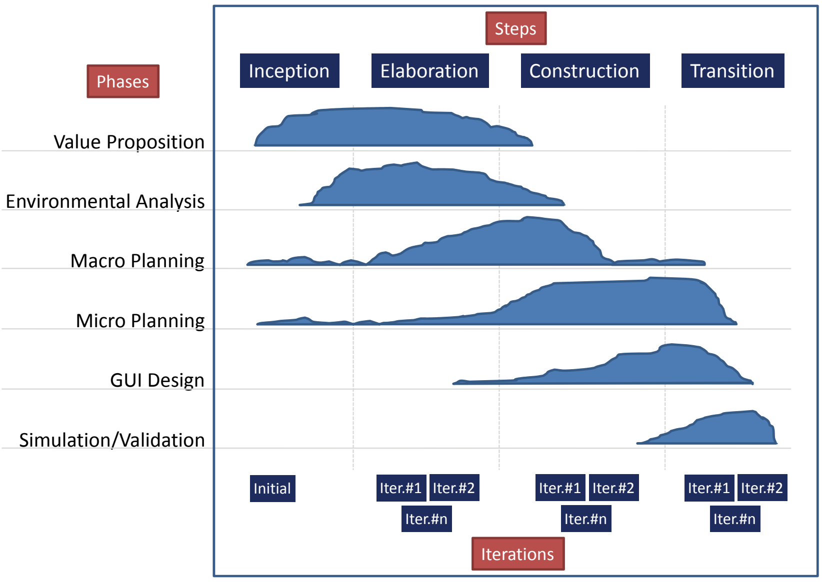


Figure 4.2
Iterations within the phases of the general approach

Each of the different steps serves its own purpose and a set of well defined actions is taken per phase:

1. *Inception*. In this step the value proposition is created collaboratively between business analysts, business domain experts, and the management. The first artifacts for the environmental analysis are constructed.
2. *Elaboration*. Following the inception, the elaboration step further refines the artifacts created in the previous step. The value proposition and the environmental analysis are finished in this phase and the first coarse grained models are constructed. In case any inconsistencies in regard to the proposed values or environmental conditions are found, artifacts are adapted accordingly.
3. *Construction*. In the construction step the macro planning artifacts are further refined and specialized resulting in a fine-grained model. The fine grained model is further equipped with GUI mock-ups. Similar to the elaboration phase the modeler has to ensure that the created fine-grained model is in accordance with the artifacts specified in the value proposition and environmental analysis phases. If necessary step 1-3 are reiterated.
4. *Transition*. During the transition phase the created artifacts are validated against the real world scenario using simulation and validation techniques. If any inconsistencies are found, the business analyst initiates another iteration in order to adapt the created model.

Iterations are used to prevent late-design-breakages

It is important to notice, that changes in the overall process design late in the development of the B2B solution are more cost intensive than in early phases of the project. This term is known as *late-design-breakage*. Especially in the early phase of a project, the requirements for a new B2B solution often and regularly change. The iterative approach provided by our methodology aims at capturing these changes at any time and at reducing the overall costs induced by the changes (late-design breakage costs).

The four areas of business process management

As shown in Figure 4.3, the business process management domain is split up into four distinctive parts: process modeling, process development, process execution, and monitoring. The general approach covers the first part of the framework - process modeling. The methodology allows the description of end-to-end flow activities across organizational units as well as inter-organizational business partners. The designed B2B model reflects the business view and is an important tool for the business owner, in order to discuss and communicate the needs of a new (or modified) process.

The final output of the process modeling phase is a validated and fine-grained process model, uniquely defining the requirements of the solution to-be. Such a process model is further refined during the process development phase where, e.g., concrete service definitions are assigned to the different process activities. The final process model of the process development phase may serve as input for execution engines during the process execution phase. Whether the

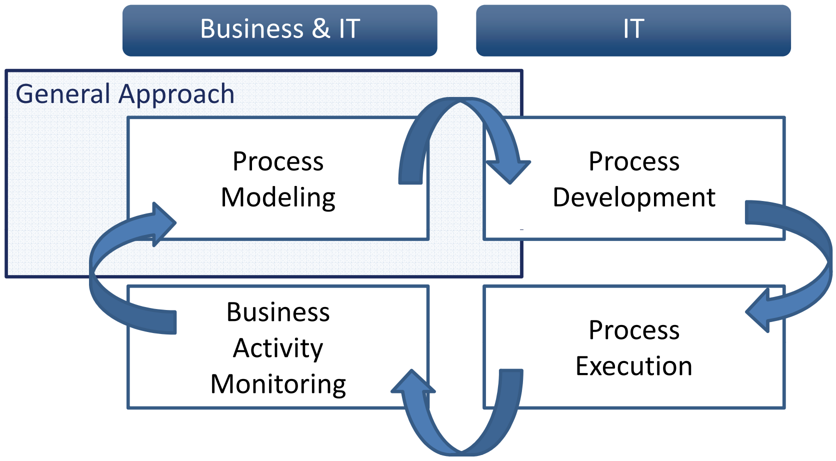


Figure 4.3 *The general approach located in the business process lifecycle*

overall goals, as specified in the value proposition phase of a developed B2B model, are met or not, is determined during the business activity monitoring phase.

In order to ensure that the difference between the actual results of the process execution and the defined goals in the value proposition phase is as low as possible, the early validation of the B2B model is of utmost importance. A first high-level simulation and validation as part of phase six of the general approach helps to apply first optimizations early in the design process.

Introducing the RACI matrix

As mentioned before, the general approach has well defined objectives and delivers a set of artifacts and documents. Furthermore it assigns clear responsibilities to each phase. Responsibilities of the different roles involved in the methodology are indicated using the RACI matrix [19]. The RACI approach splits responsibilities into four different responsibility types. The different types are then assigned to the roles within the general approach:

- *R* - responsible for producing deliverables
- *A* - accountable for quality and timeliness of deliverables and ensuring that key people are consulted
- *C* - must be consulted in the production of the deliverable
- *I* - inform, i.e., receive a copy of the deliverable

4.3 The 6 phases of the general approach

The following sections examine the six different phases of the general approach and give a detailed overview of the used objectives, scope, methods, tools and responsibilities. For every phase, an example from the accompanying use case scenario *customer acquisition* is shown.

4.3.1 Value Proposition

Objectives

Specify the value and the purpose of the model

The main objective of the value proposition phase is the business justification. In order to justify the introduction of a new B2B solution, its purpose and benefit must be specified - in a business sense. If an existing system has to be replaced, the costs for the replacement must be captured in a structured way. The replacement details in terms of the affected entities and organizational units are further specified in the next phase.

Scope

The scope of the value proposition phase covers the definition of the value delivered by the B2B solution. Furthermore the goals and non-goals of a solution are elaborated and documented. All goals defined in this phase must be in accordance to the overall IT goals of the company and in alignment to the overall IT strategy.

Methods

The value proposition phase uses a set of different methods to capture the necessary information. In order to get the required input for the value proposition phase, brainstorming sessions are held by the business analysts to get a first overview of the values and benefits of the introduction of the new IT-solution. However, the actual value propositions are not only decided by the business analyst but by the different departments and units within the enterprise. The first preliminary results of the business analyst's brain storming sessions are used to generate questionnaires and interview road-maps. Interviews and opinion-polls using the questionnaires are held together with stakeholders from the different departments and companies. Eventually this leads to a common understanding and agreement of the values the new solution is supposed to deliver for the participating stakeholders. Since all stakeholders are included in the requirements engineering process at this early phase, misunderstandings and wrong expectations are prevented. In case the new solution has direct customer interaction, the specific customer requirements are reflected accordingly. These requirements and expectations can for instance be collected by online surveys or telephone surveys.

Tools

For capturing the specific value proposition requirements the business analyst has a set of tools at his disposal including mind mapping tools, word processing tools, and project management tools.

For the customer acquisition showcase the artifacts of the value proposition phase have been elaborated using a mind mapping tool. Figure 4.4 shows a cut-out of the value proposition mind map.

The business analyst has split up the objectives of the new solution into three subcategories: value, purpose, and challenges for

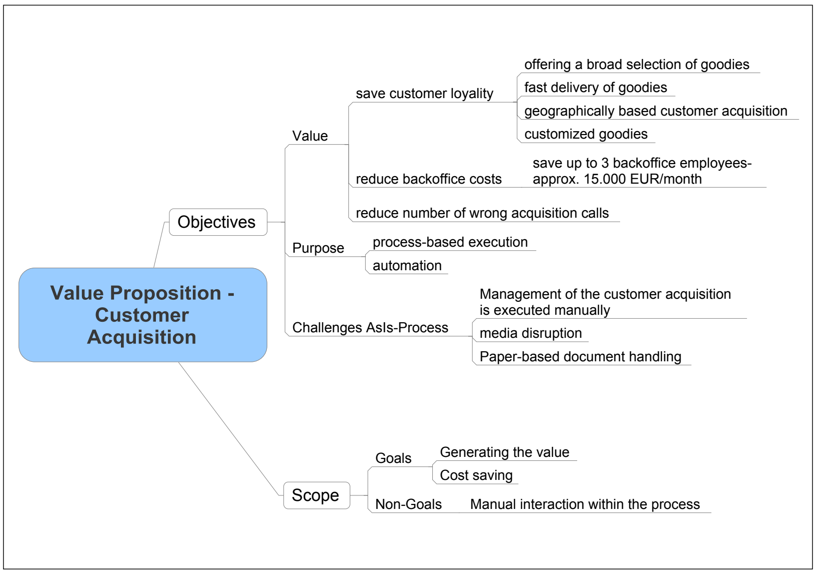


Figure 4.4
Cut-out of the value proposition mind map

the existing processes. The scope has been divided in goals and non-goals. According to the RACI specification the involved roles in this phase are: Business (R), Business Analyst (R/C) and Management (A).

After the expected values of the new solution have been captured and an agreement on the new solution in a business sense has been made, the business analyst proceeds with the environmental analysis phase.

4.3.2 Environmental Analysis

Objectives

Identify affected entities and related processes

The main goal of the environmental analysis is to define all entities which may be affected by the introduction of the new B2B solution; in other words, how the new solution fits into the existing process and system landscape. Within our proposed methodology a distinction is made between three different types of entities namely *target entities*, *affected entities*, and *side-affected entities*. Target entities are involved in the new solution, no matter if they already exist or need to be introduced. These entities are the main focus and the purpose for the introduction of the new solution. If the introduction of a new IT solution has an influence on entities which are part of other solutions, these entities are called affected entities. In contrast to a target entity, an affected entity must already exist. Entities which are part of the new solution as well as already existing solutions that remain unchanged by the introduction of the new solution are called side-affected entities. Depending on the actual context, a further sub-classification of the different entities can be made if necessary.

Target entities	Affected entities	Side-affected entities
Products	Products	Products
Customer segments	Systems	Systems
Channels	Processes	Processes
Systems	Organizational Units	Organizational Units
	Projects	Projects

Table 4.1 Classification scheme for environmental analysis phase

Scope

The scope of the environmental analysis phase includes the collection of all necessary entities and their classification by so-called worksheets. Worksheets are structured forms for gathering the requirements and capturing the domain knowledge. Figure 4.5 depicts an example worksheet for classifying the target entities of the B2B system. Although our methodology does not mandate to use a specific classification, the hierarchy as outlined in Table 4.1 is generally recommended. Note, the interactive use of worksheets for documenting B2B systems is explained in more detail in Chapter 7. Within this chapter we only give an overview about this method of capturing the domain knowledge by structured forms.

In a second step the business analyst captures the identified entities using a business process modeling tool. The representation of the different entities within the modeling tool is important, since in later steps the entities will be assigned to process activities. Thereby responsibilities of organizational units and interdependencies between the new solution and the other business environment can be shown. It is important to notice, that all decisions made in this phase have to be in accordance with the goals of the overall process strategy. This means, that the existing systems, solutions or business cases must be altered in such a way, that the overall process strategy is in alignment with them.

Methods and Tools

The methods and tools used in this phase include interviews, questionnaires and workshops. The collected information is held in worksheets and is further transferred into a formalized model representation. The cut-out in Figure 4.6 gives an overview on how UML packages can be used to structure and classify the identified entities. The business analyst has now the advantage of (re-)using those artifacts, which are already modeled within a modeling tool, in later phases of the methodology.

According to the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (R), Management (A), IT/Architecture (C/I), and Solution Designer (C).

After the business analyst has successfully completed the value proposition phase and has analyzed the environment of the new solution he starts to create a first coarse-grained model.

Target Entities	
Products	
Newspaper products	Daily Newspaper Weekly Newspaper Special Issues
Glossy paper products	TV Magazine Sports Magazine Lifestyle Magazine
Advertisement	Newspaper announcements Banner advertisements
Customer Segments	
Reader	Customer having a permanent subscription and paying a yearly fee. They get a daily newspaper.
Test Reader	Customer having a test subscription. They pay a reduced fee and get a goody for testing the newspaper.
Private Customer	Private individual
Business Customer	Retailer Tobacconists Shops, Restaurants, Gas Stations POS - Supermarket
Channels	
Written	Mail, Fax, Letter
Telephone	Customer acquisition via a call center
Online	Internet Web Services
Business Partner	Address Registry Provider (e.g. Herold) Call Center Marketing Partner (e.g. Coca-Cola, McDonalds, etc.) Third Party Vendor (e.g. Apple, Acer, OE-Ticket, etc.) Logistics (e.g. DHL, UPS)
Systems	
NewsERP	ERP (Enterprise Resource Management) tool of the newspaper publisher
NewsCRM	CRM tool (Customer Relationship Management) tool
NewsBS	The billing system of the newspaper publisher
NewsLMS	Letter-Management-System

Figure 4.5
Classification example

4.3.3 Macro Planning

Objectives

Create a coarse grained process model

During the macro planning phase the concepts of the first two phases are aggregated to a formalized model. Thus the main objective of the macro planning phase is the construction of a coarse grained model. The required processes must be identified and an overview of the most important steps and used resources is created. Furthermore the first model sketch can be used for an early management review.

Scope

The scope of the macro planning phase includes the creation of a process list. Process lists represent an ordered set of process activities and sub-activities involved in the new B2B solution. Activities and sub-activities are structured in "must-have", "should-have", and "nice-to-have" schemes. Relevant existing processes within the company and new to-be created inter-organizational processes are iden-

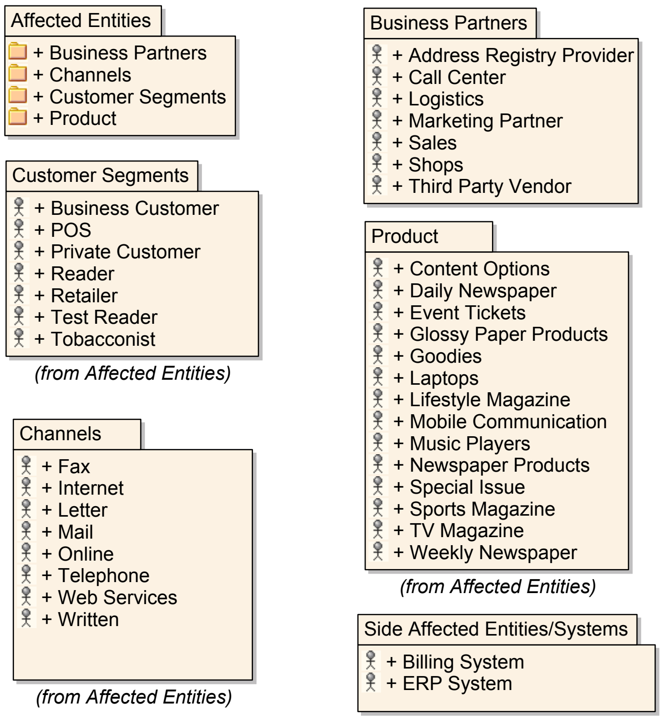


Figure 4.6 *Package structure cut-out*

tified. The delta between existing and required processes has to be analyzed in order to give a first overview of the implementation complexity. Again, it is important to notice, that all decisions made in this phase have to be in accordance with the goals specified in the first phase of our methodology.

Figure 4.7 shows a cut-out from the example macro model process list. Please note, that the categories on the right hand side follow a color code, which is also reflected in the macro planning model. The different colors denote, whether the process is classified as "must have", "should have", or "nice to have". The second column of the process list denotes the name of the (sub)-process activity. Each activity has a certain goal which is described and outlined in the third column called description and goals. An activity is finished after the goal of the activity has been fulfilled.

Methods and Tools

An XML schema is used to process worksheet information

The macro planning phase uses a set of different methods and tools to create the necessary artifacts. The process list is created using a spread sheet tool of choice e.g. Microsoft Excel. In the next step a business process modeling tool is used to create the actual process model and worksheet information artifacts. A worksheet defines the exact structure, how process meta-information is aligned, classified, and stored. For the purpose of our proposed methodology, an XML schema (XSD) is used to define the conceptual model of a worksheet. Business process management suites from tool vendors such as Oracle [142] or IBM [76] allow for the integration of XML schema artifacts into process models. The tool processes the XML schema artifacts and automatically generates the necessary forms for the busi-

ID	Process Name	Description	must have	should have	nice to have
P1	Login	The system user authenticates himself using a valid username/password combination. After a successful login the user can use all the functions of the IT-Systems			
P2	Prepare address data	The system user collects the address data from the CRM system. He can select between different groups of customers. In our case, he is only able to select the customers who are assigned to the product *newspaper*.			
P3	Request address validation	The system user sends the required data which are necessary for a validation of a set of addresses			
P4	Verify address	The addresses are checked, whether they still exist or they have been changed. The validation is done by the address registry provider by retrieving its database.			
P5	Send validation success	In case the address is correct, the address registry provider sends a list of all successfully verified addresses.			
P6	Send corrected addresses	In case the address is not actual anymore, the corrected address is sent back to the newspaper publisher's system. In case the address is not found in the database of the address registry provider, an acknowledgement flag is set for the specific address.			
P7	Check available goodies for each address	Within this sub process, the system user can check for each address, which goodie is available in respect to the geographical location			
P8	Prepare address data for telephone acquisition	In order to send the set of addresses to the call center, the system user needs to filter the addresses, e.g. by country, product (sub)type, etc.			
P9	Send data for customer acquisition	The system user sends the customer data to the call center, in order to apply for a customer acquisition via telephone			
P10	Perform call	The call center performs the call to the customer in order to get a new test subscription			
P11	Send confirmed test subscriptions	The list with all confirmed test subscriptions is sent back to the newspaper publisher			
P12	Create new test reader account	A new account for the acquired test subscription will be created			
P13	Send SMS confirmation to the test reader	This nice-to-have process sends an SMS to the test reader with a welcome message and a short summary of the contract data. Furthermore it send a reminder for a renewal two weeks before the test subscription ends.			

Figure 4.7
Process list of the macro planning phase

ness analyst where requirements information can be entered. Eventually, the requirements information is stored in XML instance documents. Since all documents have been derived from the same XML schema, interoperability between the different document instances is guaranteed. Finally the XML documents are linked to the relevant activities and sub-activities in the business process model. Therefore necessary meta-information is directly connected to the relevant artifacts in the model.

The worksheet XML schema guarantees that all worksheet instances follow the same structure and can easily be attached to process artifacts. This avoids inconsistencies between the different requirements specifications. Listing 4.1 shows an XML schema for the definition of a macro planning worksheet. Furthermore, the XML representation of requirements information allows for an automatic processing using appropriate tools.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ca="http://www.newspaperpublisher.at/CustomerAcquisition"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.newspaperpublisher.at/CustomerAcquisition"
        elementFormDefault="qualified" attributeFormDefault="unqualified">
        <!-- === ROOT element === -->
        <xs:element name="WORKSHEET" type="ca:worksheetType"/>
        <!-- === Element definitions === -->
        <!-- === Attribute definitions === -->
        <!-- === Complex Types === -->
        <xs:complexType name="generalType">
                <xs:sequence>
                        <xs:element name="processID" type="xs:string"/>
                        <xs:element name="business_process_name" type="xs:string"/>
                        <xs:element name="definition" type="xs:string"/>
                        <xs:element name="description" type="xs:string"/>
                        <xs:element name="participants" type="xs:string"/>
                        <xs:element name="automated" type="xs:string"/>
                        <xs:element name="reference" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="characteristicsType">
                <xs:sequence>
                        <xs:element name="preCondition" type="xs:string"/>
                        <xs:element name="postCondition" type="xs:string"/>
                        <xs:element name="beginsWhen" type="xs:string"/>
                        <xs:element name="endsWhen" type="xs:string"/>
                        <xs:element name="exceptions" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="relationshipType">
                <xs:sequence>
                        <xs:element name="includedBusinessProcesses" type="xs:string"/>
                        <xs:element name="affectedBusinessProcesses" type="xs:string"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="worksheetType">
                <xs:sequence>
                        <xs:element name="GENERAL" type="ca:generalType"/>
                        <xs:element name="START_END_CHARACTERISTICS"
                                                                type="ca:characteristicsType"/>
                        <xs:element name="RELATIONSHIPS" type="ca:relationshipType"/>
                </xs:sequence>
        </xs:complexType>
</xs:schema>
```

Listing 4.1
XML schema specifies the conceptual model of a worksheet

Integration of an XML schema to Microsoft Word

The tool *Enterprise Architect* allows to attach XML instance documents to activities and sub-activities in a business process, but does not support the automatic form generation based on XML schema. Thus, a different tool had to be chosen in order to capture the text based requirements by using the same XML schema. We suggest to use Microsoft Word 2007 as the tool of choice for capturing worksheet data. Figure 4.8 shows the integration of XML schema into Microsoft Word 2007.

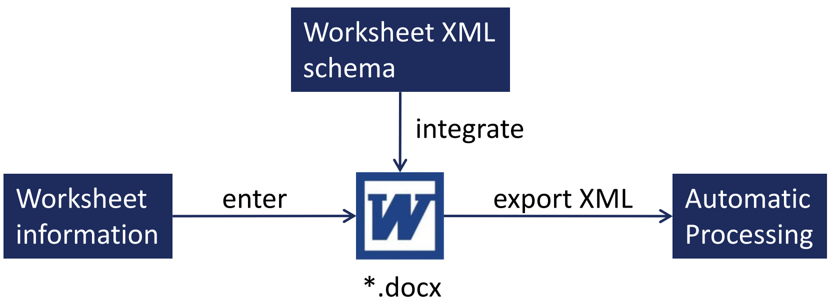
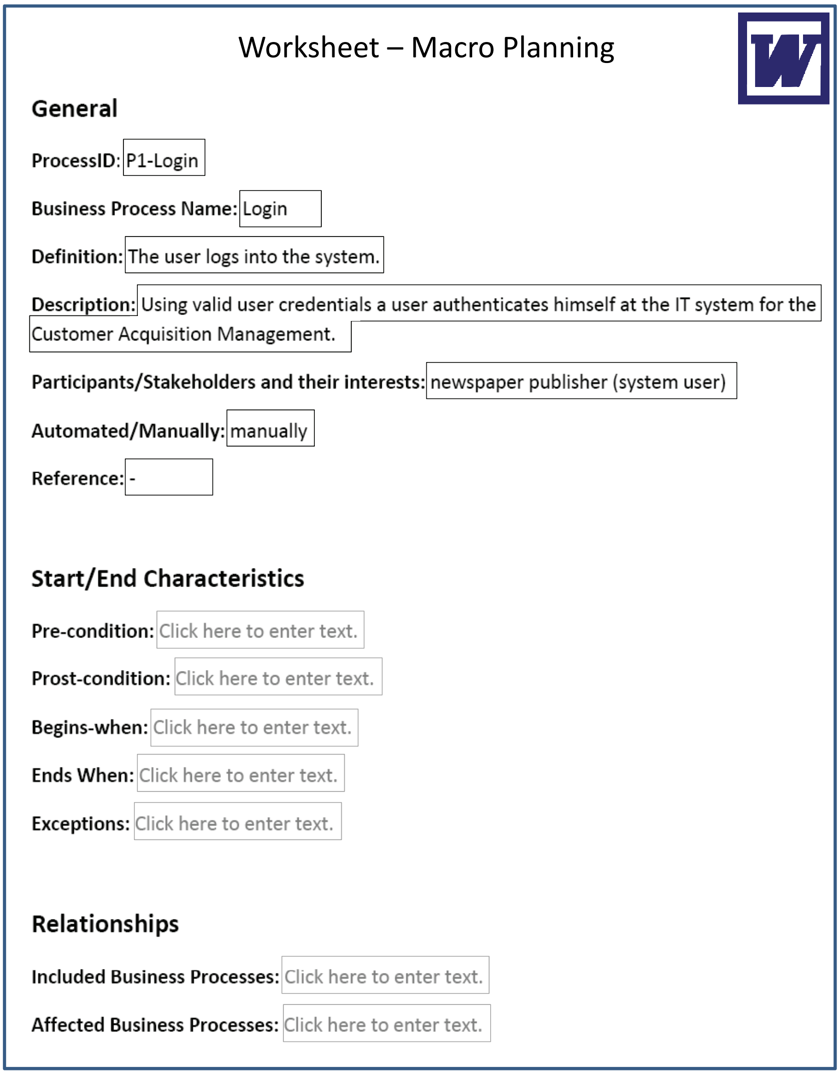


Figure 4.8
Integration of XML schema data into MS Word 2007

First an XML schema is associated with a Microsoft Word document. Due to the associated XML schema, the business analyst is presented a predefined worksheet structure where only designated fields can be filled out. The entered worksheet information can either be used as a regular Microsoft Word document (e.g. for communication purposes between business analysts) or exported as XML instance. Figure 4.9 shows an example of an XSD-generated worksheet template, which can be used as a kind of form in Microsoft Word to capture the requirements of a certain process step. In the following the exported XML instance is copied and attached to the business process model thus enabling automatic processing. Listing 4.2 shows such an XML output of the worksheet information that has already been entered to the word-processing tool as shown in Figure 4.9.

The inclusion of XML schema artifacts into Microsoft Word is currently supported by version 2007 of Microsoft Word which supports the Office Open XML standard [35]. The major advantage of this approach is the common storage of free-form text requirements information, captured by worksheets, within business process models. Inconsistencies between requirements artifacts can be avoided and automatic processing of requirements information is enhanced. The technical implementation and the use of XML-based worksheets will be explained in more detail in Chapter 7.

Worksheets are used to avoid inconsistencies between requirements information



Worksheet – Macro Planning

General

ProcessID: P1-Login

Business Process Name: Login

Definition: The user logs into the system.

Description: Using valid user credentials a user authenticates himself at the IT system for the Customer Acquisition Management.

Participants/Stakeholders and their interests: newspaper publisher (system user)

Automated/Manually: manually

Reference: -

Start/End Characteristics

Pre-condition: Click here to enter text.

Prost-condition: Click here to enter text.

Begins-when: Click here to enter text.

Ends When: Click here to enter text.

Exceptions: Click here to enter text.

Relationships

Included Business Processes: Click here to enter text.

Affected Business Processes: Click here to enter text.

Figure 4.9 *XSD-generated worksheet for gathering the requirements of a macro planning process step*

```
<?xml version="1.0" encoding="UTF-8"?>
<ca:WORKSHEET xmlns:ca="http://www.newspaperpublisher.at/CustomerAcquisition"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.newspaperpublisher.at/CustomerAcquisition␣Macro_Worksheet-Schema.
                xsd">
```

Listing 4.2 *XML schema specifies the conceptual model of a worksheet*

```
    <ca:GENERAL>
        <ca:processID>P1–Login</ca:processID>
        <ca:business_process_name>Login</ca:business_process_name>
        <ca:definition>The user logs into the system.</ca:definition>
        <ca:description>
                Using valid user credentials a user authenticates himself at the IT
                system for the Customer Acquisition Management.
        </ca:description>
        <ca:participants>newspaper publisher (system user)</ca:participants>
        <ca:automated>manually</ca:automated>
        <ca:reference>–</ca:reference>
    </ca:GENERAL>
    <ca:START_END_CHARACTERISTICS>
        <ca:preCondition/>
        <ca:postCondition/>
        <ca:beginsWhen/>
        <ca:endsWhen/>
        <ca:exceptions/>
    </ca:START_END_CHARACTERISTICS>
    <ca:RELATIONSHIPS>
        <ca:includedBusinessProcesses/>
        <ca:affectedBusinessProcesses/>
    </ca:RELATIONSHIPS>
</ca:WORKSHEET>
```

After the business analyst has finished the worksheets and the process list, the macro model must be created. In the following example the Business Process Modeling Notation (BPMN) [133] is used to depict the macro model. Figure 4.10 shows a simplified cut-out from the macro model for the management of the address data for the telephone acquisition.

Pools and lanes within the macro model

Within the macro planning model the different process activities are aggregated in so called pools and lanes. A lane indicates an organizational unit which is responsible for an activity. Lanes can be nested in order to depict intra-organizational dependencies, e.g., in Figure 4.10 the organizational units *Marketing and Sales* and *Finance* are involved in the process. Pools are used to strongly depict inter-organizational borders between different business partners - e.g. *Address Registry Provider* or *Call Center*. Furthermore, target entities, affected entities, and side-affected entities which are involved in the process are modeled by using pools as well. Remember, those entities have already been defined in the model during the environmental analysis (see Figure 4.6). In the micro planning phase, the elements are assigned as instances to the relevant process activities. This kind of reuse of the elements defined in an earlier phase is denoted by (**:**) at the beginning of the element's name. Those pools are directly connected to activities where necessary, e.g., the *CRM System* is needed to prepare the address data in process P1. It is important to notice, that the macro planning model targets to give a brief overview about the solution to-be. No activities are refined in detail nor are any pre- or post-conditions defined or GUI elements assigned in this early stage. According to the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (R), IT (C), and Management (A).

In the next step the business modeler refines the macro-model by adding additional, finer-grained requirements information to the model.

4.3.4 Micro Planning

Objectives

Finalize the process model

The main goal of the micro planning phase is the further refinement of the artifacts created during the macro planning phase until the

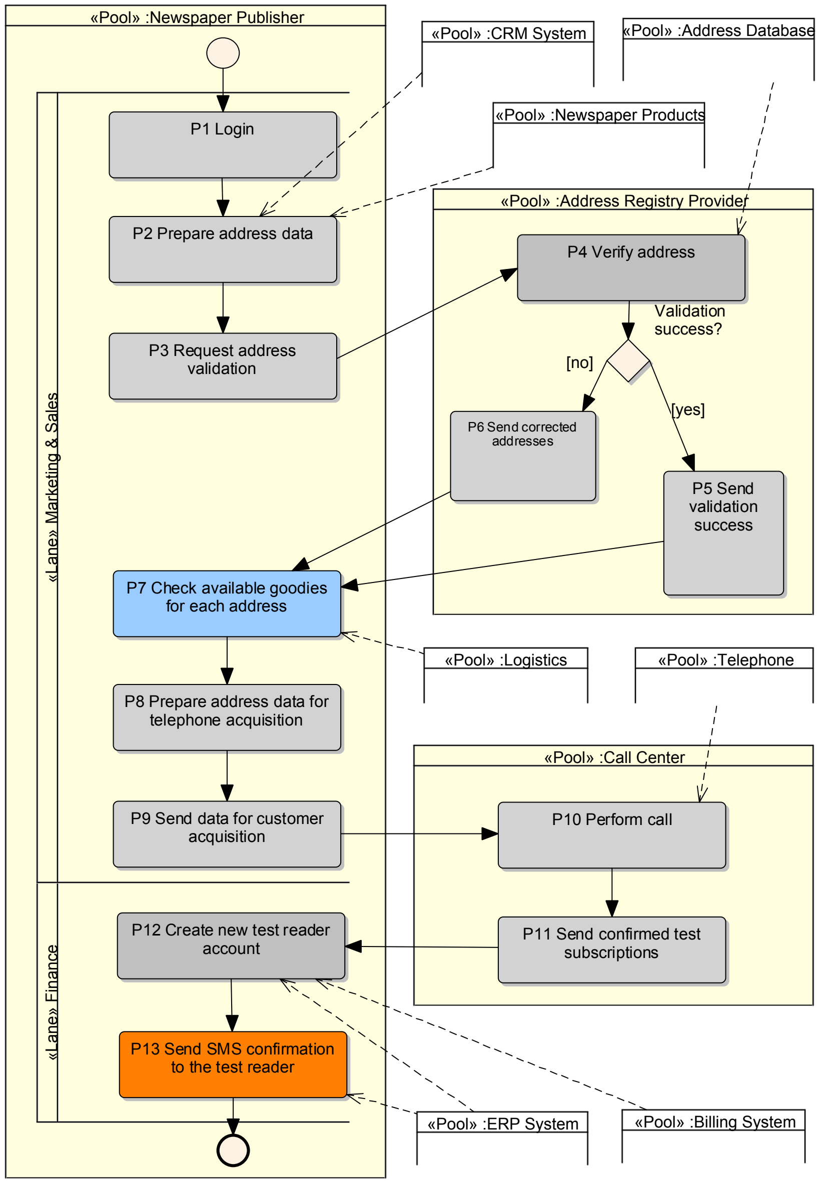


Figure 4.10 *Cut-out of the simplified Macro Model*

final business process model is finished. Furthermore, the requirements documentation using the different worksheets and the process list is completed in this phase. Thereby, the micro planning phase uses the same methods and tools as the macro planning phase - only the level of detail is more fine-grained.

Scope

The scope of the micro planning phase includes the definition of a fine-grained process model. Thereby the different business process activities from the macro planning phase are gradually refined using sub-processes. For every identified sub-process the necessary worksheets and process list entries are created as well. Furthermore, a differentiation of the identified sub-processes in terms of *automated* or *manual* processes is made.

Methods and Tools

Identify required business functions by sub-activities

The first task for the business analyst is the refinement of the different worksheets analog to the previous phase. Since most activities from the macro planning model are split into sub-activities, each sub-activity must have its own worksheet. In the next step the business analyst refines the process list from the macro planning phase. Thereby the different process activities are further elaborated using the concept of sub-activities. As shown in the process list in Figure 4.11 the activity *P2 - Prepare address data* from the macro-planning phase is split up into sub-activities *P2.1 - P2.7*.

ID	Process Name	Description	manual/ automatic	must have	should have	nice to have
P1	**Login**	The system user authenticates himself using a valid username/password combination. After a successful login the user can use all the functions of the IT-Systems	m/a			
P1.1.	Load Login page	The necessary login page is loaded.	a			
P1.2.	Logon procedure	The system checks the user credentials against the user database. If the credentials are valid the user is redirected to the requested GUIs. Otherwise an error message is shown.	a			
[...]	[...]	[...]	[...]			
P2	**Prepare address data**	The system user collects the address data from the CRM system. He can select between different groups of customers.	m			
P2.1	Choose newspaper product	There are different newspaper products within the system. The system user selects a specific product (e.g. Daily Newspaper) in order to get the customers who are assigned to this product.	m			
P2.2	Select customer group	The customers are subdivided into different customer groups (e.g. business customer). The system user is able to select one of those groups in order to process with the next step.	m			
P2.3	Assign addresses to key account business center	The addresses of business customers need a special treatment in regard to customer acquisition. If the business customer group has been selected, the system assigns the addresses to the key account business center.	a			
P2.4	Bundle retailer addresses by branch	The retailer accounts are subdivided into different branches. Within this process step, the system user selects different branches in order to restrict the set of addresses.	a/m			
P2.5	Filter addresses by customer location	The geographical location of the customer is important for a personalized customer acquisition. The system filters the addresses by the geographical location of the customer in order to process with the next step.	a			
P2.6	Assign addresses to call center	Different call centers are responsible for different groups of customers depending on their geographical location. The system automatically assigns addresses to the call center according to the location of the address.	m			
P2.7	Attach optional customer information	The system user should be able to attach optional customer information to each address. This feature is classified as "should-have".	m			
P3	**Request address validation**	The system user sends the required data which are necessary for a validation of a set of addresses	m			
[...]	[...]	[...]	[...]			

Figure 4.11 *Micro model process list cut-out*

The same principles as applied to the macro process list are also applied to the micro process list. The second column contains the name of the (sub)-activity which is performed and the third column contains the concrete description and goal of the activity. Following the update of the process list, the process model is updated accordingly. An essential functionality of the micro-planning phase are composite activities. A composite activity is refined using another BPMN activity diagram. As an example, Figure 4.12 shows the activity diagram which refines the composite activity *P2 - Prepare address data*.

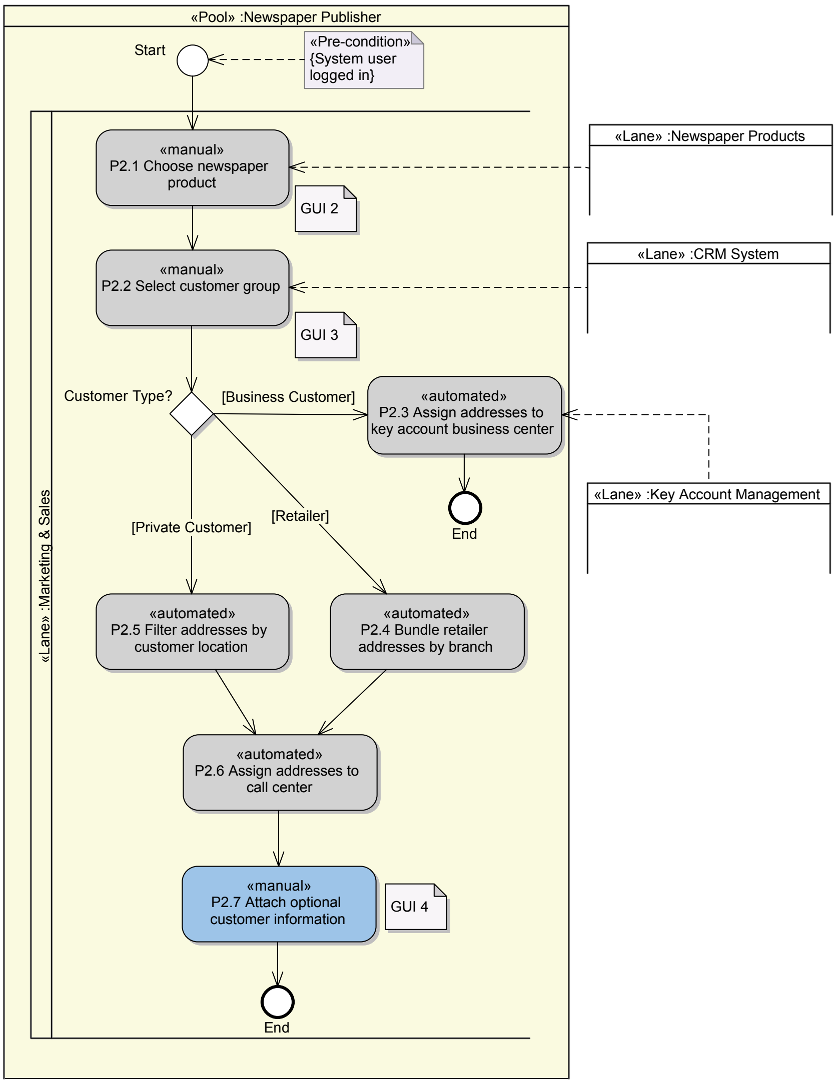


Figure 4.12
Micro model cut-out

Refining and classifying the process activities

The refined process activities in the micro model are classified into *manual* and *automated* activities. An automated activity is executed by an IT system without any human interaction. A manual activity requires human interaction. As outlined in Figure 4.12, the concept of stereotypes is used in order to determine whether an activity is manual or automatic. Again, the color codes of the different activities help to differentiate between "must-have", "should-have", and "nice-to-have" processes. Color codes are taken from the process list and are reflected accordingly in the process model.

Pre-conditions for activities or whole processes are indicated using the concept of constraints. As shown on top of Figure 4.12, the pre-condition for the whole *P2 - Prepare address data* process is a logged-in system user. Thereby the process *P1* must be executed in order to allow the *P2* process to start.

Re-using artifacts of the environmental analysis phase

An organizational unit involved in the process is denoted using the concept of lanes. E.g. process steps *P2.1* to *P2.7* are executed in the context of the *marketing and sales department*. Affected entities, target entities, and side affected entities involved in the process are denoted using the concept of pools. E.g. process step P2.3 is side-affected by the *key account management system* and process step P2.2 additionally involves the *CRM-system*. All organizational units and affected entities have been defined during phase 2 - environmental analysis.

The RACI matrix for the micro planning phase

Using this concept, the business analyst can easily refine an existing macro model to the desired extent in order to meet the requirements of a micro model. According to the RACI specification the involved roles in this phase are the same as in the micro planning phase: Business (R/C), Business Analyst (R), IT (C), and Management (A). The final micro-model is assigned with the necessary GUI mock-ups which are defined in the following phase.

4.3.5 GUI Design

Objectives

Design first GUI mock-ups

The GUI design phase concerns the definition of GUI mock-ups for relevant process steps with human interaction. This means, that there is no technical implementation of any user interfaces required in this phase. The main objective of the GUI design phase is to demonstrate the look-and-feel of manually operable interfaces of the B2B solution. GUI mock-ups can be designed using any kind of graphical editor (e.g. Microsoft Visio). The output of this phase is a set of static user interfaces and a storyboard for a GUI sequence. The GUI mock-ups also play a major role in the micro planning phase. Each GUI must be assigned to a certain process activity in the micro model.

Scope

A GUI storyboard is used to depict the interactive flow of GUI mock-ups

The scope within this phase spans from the design of the GUIs to the definition or relationships between the different GUIs. The design of the conceptual user interfaces is based on the collected information of the previous phases. According to the different process activities and sub-activities defined in the micro-planning phase, the GUI designer creates mock-ups for the different user interfaces. Naturally only manual activities which require user-interaction are assigned with GUI mock-ups. The mock-up design must consider state of the art principles e.g. accessibility standards, minimum font-size etc. Finally the GUI designer assembles the different GUI mock-ups to a final GUI storyboard as shown in Figure 4.13.

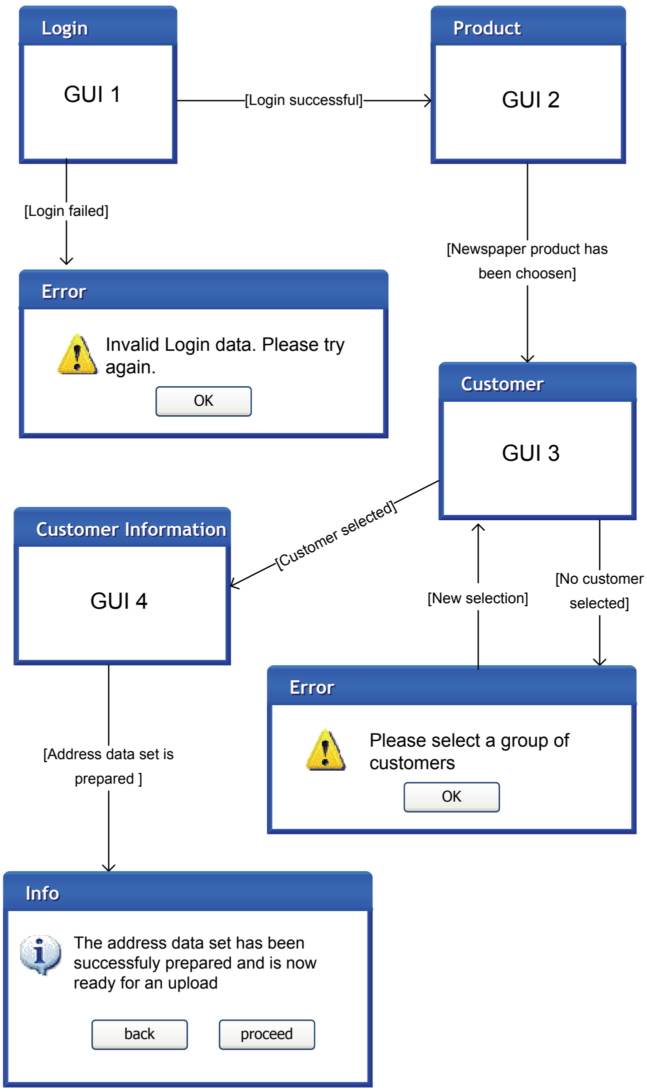


Figure 4.13 *GUI storyboard for the "prepare address data" sub-process*

Methods and Tools

The GUI storyboard for the example showcase was designed using Microsoft Visio 2007. Figure 4.13 depicts the flow of GUI, whereas the sequential flow of the different user interfaces is defined by the guarded transitions (e.g. [Login successful/failed]). This means, that the user is only able to get from GUI 1 to GUI 2, if he enters the correct login data. Each GUI symbol in the storyboard refers to a specific and accurately described GUI mock-up. As an example, the GUI mock-up of GUI 2 *Choose newspaper product* is depicted in Figure 4.14. Within such a GUI mock-up it is important to annotate restrictions for the system user (e.g., by the use of regular expressions or constraints in natural language). Note, a GUI mock-up does not necessarily need to reflect the final design of the user interface. However, it should capture the most important features the system user is interacting with.

Assign the GUI mock-ups to the process activities

Finally, the different GUI IDs are attached by notes to the micro-model activities and sub-activities. As shown in Figure 4.12 every manual activity has the relevant GUI artifact assigned. E.g. the manual activity *P2.1 Choose newspaper product* requires the GUI element *GUI 2*.

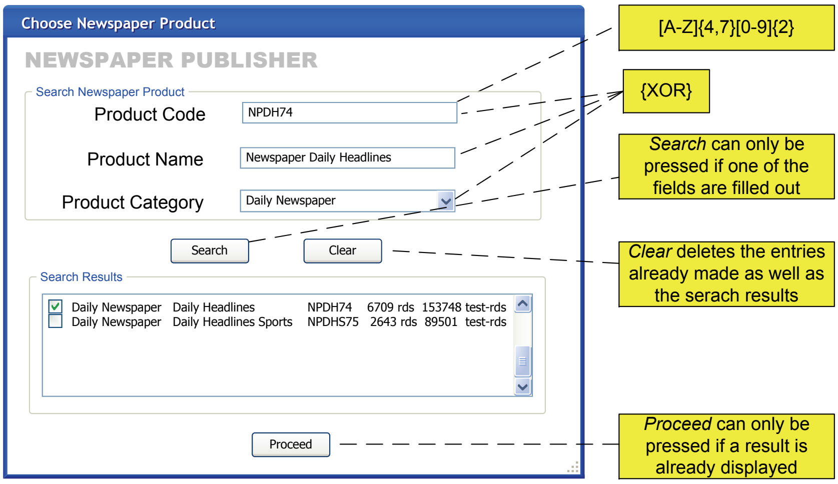


Figure 4.14 *Example of a GUI mock-up for the Choose Newspaper Product business case*

According to the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (I), IT/GUI Engineer (R), Process Owner (A), and Solution Designer (C/I).

After having defined the different user interfaces the process modeling phase is finished and the business analyst starts the final phase of the methodology - the simulation and validation of the model.

4.3.6 Simulation and Validation

Verification of the designed models

The final step in the methodology is the validation and simulation of the constructed business process model. The importance of the integration of a simulation and validation into the business process modeling lifecycle is already shown in [96] and [6]. Business process simulation is based on so called discrete event simulation [61]. Figure 4.15 gives an overview about the basic principles of a discrete event simulation. The simulation component interprets the business process model and simulates transitions of activity instances such as active or terminated. Thereby, it removes an event from the event list, processes the event and causes changes to the model. Each event has a time stamp attached to it and the processing of events can generate new events which are automatically inserted into the event list. It is important to notice, that the clock advances from one discrete time to the next (in contrary to real-time simulation).

Objectives

Simulating the processes

The main objective of the simulation phase is the evaluation of the designed model from the micro planning phase. It has to be ensured, that it is consistent with the results from the value proposition and environmental analysis phase. In regard to simulation, the objective of this phase is the evaluation of the constructed model under different scenarios. A simulation of the process on the model level

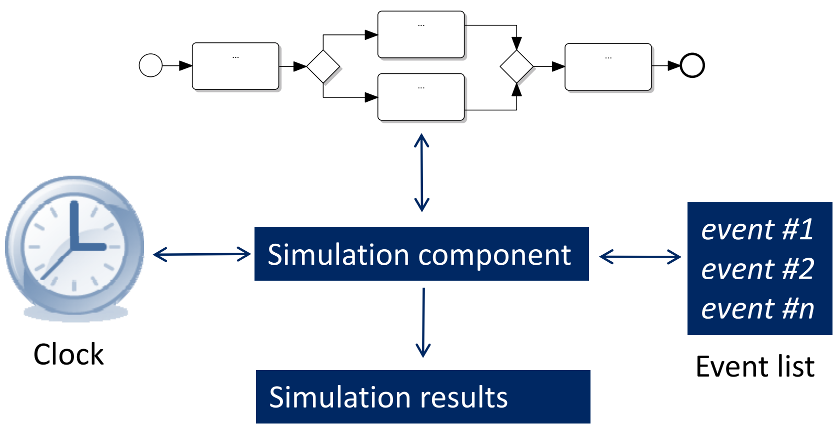


Figure 4.15 *Discrete Event Simulation for Business Processes*

helps to apply first optimizations early in the design process. Furthermore possible deadlocks or synchronization errors can be identified and possible deviations from real world processes become apparent. Moreover, first predictions can be made, e.g. what is the behavior of the model if more resources are available, the process flow has changed or any given cost of an activity (time, money, resource) changes.

Scope and Methods

Before a model validation can be initialized, the process model must be complete and ready for simulation. The specific goals for the validation must be specified and be defined in the model. The process model itself must be stable enough for a validation. This means, the necessary data annotations for the different resources and activities involved must be made e.g. time per activity, cost per resource, etc. In general, the appropriate simulation data must be available within the company. An example simulation run would at least include the following steps:

1. The different goals of the simulation must be defined precisely and it must be decided whether the simulation is feasible in terms of costs, available test data, etc.
2. The existing data within the company must be analyzed and it must be decided whether the test data is appropriate for setting the simulation data. If the business process model is very complex, the simulation run should be split up into appropriate sub-processes.
3. The necessary data must be collected and aggregated.
4. The aggregated data must be incorporated into the process model.
5. Several simulation runs with different parameters should be executed in order to allow for a broader interpretation of the results.
6. The different strengths and weaknesses of the model can be identified using the simulation results.

7. If the overall process goals and the simulation results differ significantly, the appropriate changes have to be made to the model and the simulation must re-run.

In case the simulation results do not comply with the goals specified in the value proposition phase, the business analyst has to step back in the methodology and must apply changes to the macro and micro model respectively.

Tools

Tools must support model validation of business processes

The simulation phase of our general approach depends on the capabilities of the tools used. In order to allow for a validation of the generated business process artifacts, the business process modeling tool must support validation features. For the customer acquisition example used within this thesis, Enterprise Architect [181] has been used as the modeling tool of choice to create a model which is compliant to our methodology. However, Enterprise Architect does not support the validation of a constructed model and therefore no concrete example for a model validation is given here. Tools supporting model validation of business process models include Oracle Business Process Management Tools [142], IBM WebSphere [76], etc. According to the RACI specification, the involved roles in this phase are Business (R), Business Analyst (R/C), IT (C), Management (A), and Solution Designer (C).

With the final validation and simulation of a the model, the process modeling phase as shown in Figure 4.3 is finished. The final process model is used for the process development phase where, e.g., concrete service definitions are attached to the business process model, in order to use it as input for the process execution phase.

4.4 Summarizing the approach

As a summary of the general approach we show a table with the most important artifacts delivered by the methodology: Figure 4.16 depicts the goals, artifacts, used tools and the executing roles of each phase. This table serves as a kind of "cheat sheet" for any business analyst to control whether each phase delivers the correct artifact for gathering the requirements of a B2B solution.

4.5 Transition to the business modeling based approach

By this approach we have presented a rather light-weight approach for the requirements engineering in a SOA context. The presented, process-based approach helps to overcome a set of limitations of classical requirements engineering approaches. We have introduced the six phases of our approach and have shown the application of the different phases by using the customer acquisition example. Through the process based requirements engineering the newspaper publisher

Phase	Goal	Artifact	Tools	Roles
(1) Value Proposition	Specify the value and the purpose of the model	○ "Value based" Mind Map ○ Goals and non-goals ○ Worksheets ○ Business justification	○ Microsoft Word ○ Mind Map Tool ○ Project Management Tool	○ Business (R) ○ Business Analyst (R/C) ○ Process Owner – Management (A)
(2) Environmental Analysis	Identify affected entities and related processes	○ Target, affected and side-affected entities ○ Worksheets ○ Feasibility study	○ Microsoft Word ○ Project Management Tool	○ Business (R/C) ○ Business Analyst (R) ○ IT/Architecture (C/I) ○ Process Owner – Management (A) ○ Solution Designer (C)
(3) Macro Planning	Create a coarse grained process model	○ Coarse grained process model ○ First process model sketch for management review ○ Process classification of must have, should-have, nice-to-have	○ Enterprise Architect ○ Microsoft Word	○ Business (R/C) ○ Business Analyst (R) ○ IT(C) ○ Process Owner (A)
(4) Micro Planning	Finalize the process model – identify required business functions	○ Refinement of the macro model ○ Fine-grained business process model ○ Detailed process list	○ Enterprise Architect ○ Microsoft Word	○ Business (R/C) ○ Business Analyst (R) ○ IT(C) ○ Process Owner (A) ○ Solution Designer (C)
(5) GUI Design	Design first GUI mock-ups and assign them to process activities	○ GUI mock-ups ○ Look-and-Feel of the solution ○ GUI storyboard ○ Assigned GUIs to process activities	○ Graphics Editor (Microsoft Visio) ○ Enterprise Architect	○ Business (R/C) ○ Business Analyst (I) ○ IT/GUI Engineer (R) ○ Process Owner (A) ○ Solution Designer (C)
(6) Simulation & Validation	Verification of the designed models & process simulation	○ Test data ○ Test data enhanced process model ○ Simulation plan	○ Process-Modeling tool supporting model validation (e.g. Oracle BPM Suite)	○ Business (R) ○ Business Analyst (C/I) ○ IT (C) ○ Process Owner (A) ○ Solution Designer (C)

Figure 4.16
Summary of the general approach

is able to faster deploy new solutions, thus avoiding costs and increasing customer satisfaction.

Shortcomings of the general approach

However, as a critical reflection we need to state, that the general approach has some shortcomings. First of all, the approach rather focuses on internal business processes and does not consider the business choreography of collaborating business partners. Although it captures the rough specification of the interfaces to external partners, an unambiguous specification for an IT implementation cannot be provided. Therefore, the general approach needs to be adapted to be applicable for specifying the requirements for inter-organizational systems. For example, the approach does not use existing business modeling techniques - such as e^{3}value , REA or BMO - for capturing the business justification. In order to make the general approach applicable for inter-organizational systems, e^{3}value could be used in the first phase, the value proposition, to evaluate the economic drivers for the IT solution. In contrary, REA focuses more on the implementation of the solution from an economic point of view. Therefore, some concepts of the REA ontology would improve the second phase - the environmental analysis. Finally, in our general approach we use business process modeling techniques, such as BPMN, that are not specifically tailored for the purpose of modeling inter-organizational business processes. Concepts, such as the explicit modeling of business transactions between exactly two business partners and their corresponding synchronization of business states are not a major focus in the BPMN standard. UMM is a broad-accepted B2B modeling standard and overcomes such limitations. For example, UMM's *business requirements view* can be used to capture the coarse grained business processes as delivered by the macro planning phase of our general approach. UMM's *business choreography view* can be used to model the detailed business processes (phase 4 - micro planning) that are used to interact between different business partners. Since we are part of the editing team of the current UMM 2.0 specification [193], we can integrate the B2B modeling standard into our approach. Therefore, we took the general approach described in this chapter as a starting point to introduce our business modeling and worksheet based approach which incorporates concepts from e^{3}value, REA and UMM. In the next chapter we describe those methods, their benefits, as well as their shortcomings and our proposed improvements.

5 Languages for engineering inter-organizational systems

Linking business models and process models

The approach followed in this thesis is to link business models with business process models. Currently, several languages exist for both, business modeling and business process modeling. In [173] and [174], we identified a sub-set of those methods as the most promising ones for our business modeling and worksheet-driven approach for modeling B2B systems. In this chapter, we describe the interlinked languages, which are later on used for the conceptual mapping between each other. It also needs to be mentioned, that not all of the described methods are taken as they are defined in their specification. Some of them do have some limitations and shortcomings when taking them for our approach. Therefore, we either customize parts of these methodologies for our approach or extend their base concepts. In this chapter, we also describe those changes to the original notations or standards. Furthermore, we are active members of the specification editing team of two out of the three interlinked methods, which helps us to disseminate the proposed changes to the different standards.

e^3value for calculating the return on investment

The first method concerns business models. It starts with the analysis of the economic drivers for the business collaboration. The resulting business models are described by means of the economic values that are exchanged between the business partners. These models are used to analyze who is making how much profit by exchanging objects of value with each other. An appropriate methodology on this level of abstraction is e^3value [49] which analyzes the business models in a network constellation of multiple participants for a business case.

REA for specifying the economic requirements which are implementation relevant

The second method is REA (Resource-Event-Agent), which is a business modeling ontology that originates as an economic and (especially) accounting ontology [118]. In contrary to e^3value, REA has a clear measurement and identification foundation [180]. This means that concepts that cannot be identified (with surrogates like primary keys) and that cannot be measured (for example, in an independent market) are likely candidates for eventual exclusion from the model, although they certainly should be considered at the initial, conceptual stages. The e^3value methodology seems less constrained on these issues with a strong focus on the definition of exchanged values - no matter, whether they can be considered in the implementation of the B2B system or not (e.g. customer loyalty). Therefore we propose to use REA for specifying the economic requirements which are implementation relevant, in order to have clear and unambiguous input for business process modeling. However, there are some shortcomings in order to use the REA ontology for our purpose of modeling multi-party collaborations. Furthermore, the concepts of REA are defined as informal text, table definitions and diagrams, such as ER diagrams, UML class diagrams or other graphical formalism. In other words, the REA ontology is not sufficiently explicit, and not based on a demonstrable shared conceptualization [41]. Thus, we provide some thoughts in order to overcome those limitations.

UMM model specifies the inter-organizational processes

The third method addresses the inter-organizational business processes between business partners. In order to guarantee that each partner deserves its economic value they have to agree with each other on the inter-organizational business process to realize the value exchanges. A business process in a peer-to-peer collaboration is called a choreography. Accordingly, a choreography describes the flow of interactions between the participating business partners that interlink their individual processes. A global choreography defines the inter-organizational process from a neutral perspective. It has the potential to achieve an agreement between the partners. Inasmuch, a resulting global choreography becomes a kind of contract guiding the business partnership. An approach on this level of abstraction is delivered by the United Nations Centre of Trade Facilitation and e-Business and their UN/CEFACT Modeling Methodology (UMM) [214]. The UMM specification [188] is defined as a UML profile, which we have co-edited. Usually, commitments are made on a bi-lateral basis. Accordingly, UMM models always describe business collaborations between two parties. Also similar to a contract, a UMM model describes the commitments and agreements from a neutral perspective. In summary, the UMM provides a methodology for bi-lateral and global choreographies.

The remainder of this chapter is structured as follows: the main concepts of e^{3}value are explained in Section 5.1. In Section 5.2 we discuss the basic principles of the REA ontology and we demonstrate our thoughts about an improvement and customization of the ontology. In Section 5.3 we explain the basic concepts of the UMM 2.0 standard.

5.1 e^{3}value

The e^{3}value methodology has been developed to model a value web consisting of actors who create, exchange, and consume things of economic value such as money, physical goods, services, or capabilities[48]. It is an ontology-based methodology for modeling and designing business models for business networks incorporating concepts from requirements engineering and conceptual modeling [49]. e^{3}value is based on the principle of economic reciprocity meaning a "give-and-take"-approach between actors exchanging objects with an economic value - e.g. if a seller delivers goods to a buyer, he gets money in return for the goods - see Figure 5.1.

5.1.1 The concepts of e^{3}value

The key concepts of the graphical notation of e^{3}value

The graphical notation of e^{3}value comprises a small set of concepts and relations (see Figure 5.2) that have been introduced by Gordijn in [47]. Having a deeper look at the simple e^{3}value example in Figure 5.1 *Actors* are represented as rectangles (A). They are perceived by their environment as independent economic entities engaged in a value exchange. By exchanging *Value Objects* (B) they either aim for profitability (in case of an enterprise) or maximizing their economic utility (in case of an end-consumer). Value objects do not necessarily need to be a physical good. Sometimes they represent a service, right or even a customer guarantee. A value object is always modeled in combination with a *Value Transfer* (C) and is represented as text field. A value transfer (also called value exchange) is graphically modeled as a connection between actors.

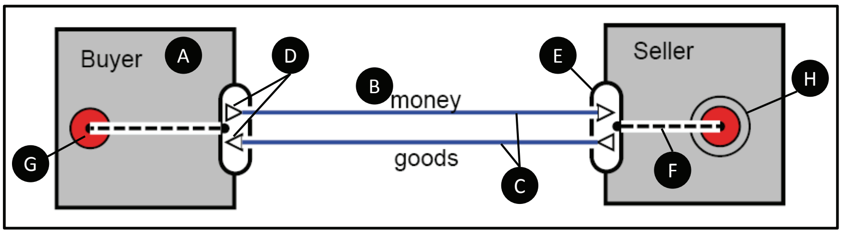


Figure 5.1
Simple example of an e^{3}value exchange

Bundling of value objects by value interfaces

Value objects are exchanges between actors using *Value Ports* (D). The concept of a value port is to signalize whether the actor offers or requests a value object. Furthermore, it enables to abstract from the internal business processes, and to focus on how external actors and other components of the e^{3}value model can be "plugged-in". Value ports are shown as small arrows pointing in the direction of the value exchange. A *Value Interface* (E) groups individual value ports. Each actor may have multiple value interfaces containing value ports for offering and requesting value objects. Value interfaces bundle the value objects an actor is willing to exchange in return for other value objects. The exchange of value objects via a value interface is atomic in order to denote reciprocity - i.e., either all exchanges occur as specified by the value interface or none at all.

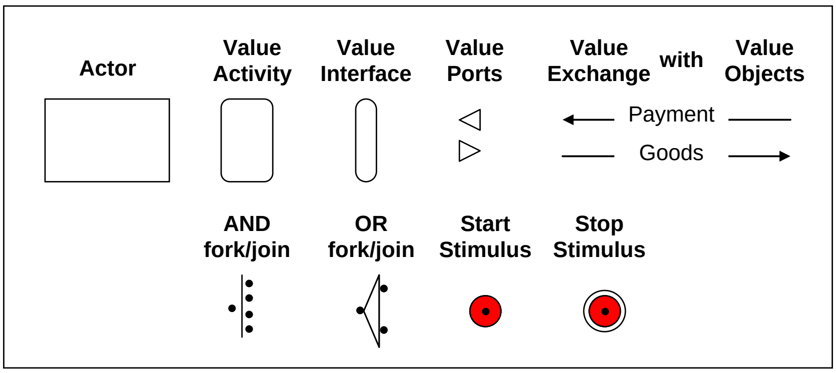


Figure 5.2
The notation of e^{3}value

The scenario path of e^{3}value

All concepts of e^{3}value discussed so far describe the inter-actor dependencies. In order to describe the intra-actor dependencies, scenarios are used to relate an actor's value interface. Such scenario techniques are described by so-called use case maps (UCMs) [21] and are used within the e^{3}value methodology in a simplified way. A scenario path (F) indicates via which value interfaces objects are exchanged. In order to keep track of a scenario path, it starts with a start stimulus (G) and ends with a stop stimulus (H). With these concepts a scenario path can pass through different actors being connected by a dotted line within an actor. AND forks as well as OR forks (and their corresponding joins) can be used to model two or more sub-paths.

Calculating the profitability sheet

The purpose of the previous mentioned scenario path is not only to depict the dependencies between different value exchanges. A more important task of this e^{3}value concept is the concrete support for a calculation of the profitability of the business network. This means, that one can analyze an e^{3}value model by quantifying the net value flow for each actor in the value web [52]. In order to calculate the profitability sheet we first have to estimate the number of value exchanges in a time period (e.g. a year) and second to valuate the value objects being exchanged in terms of monetary units. The net value flow for a specific actor is then calculated by subtracting all out-going value transfers from all in-going value transfers. The result tells us whether the network constellation could be profitable and sustainable for a specific actor. We assume that sustainability is associated with a positive long term economic performance. The generation of the profitability sheet is automated by the e^{3}value toolset [47]. Thus, we do not go into detail about the calculation of the net value flow. However, the profitability sheet is part of our requirements management approach for B2B processes and specifies a deliverable in order to create an economic sustainable e-business network. For this reason, we provide an example of a profitability sheet in Section 5.1.2.

It is important to stress that e^{3}value does not specify any order in time. This means that there is no order between the values exchanges within a value interface. Nor is there any order between the value exchanges of value interfaces connected by scenario paths. This is a very significant difference between e^{3}value representing a business (value) modeling ontology and business process modeling approaches [50].

5.1.2 e^{3}value by example

The following example has been taken from the business case described in Chapter 3. Figure 5.3 depicts the e^{3}value scenario of the acquisition of new test readers within the print media domain. In the following we will have a deeper look at the value transactions of our use case scenario. In e^{3}value a value transaction groups value transfers (the exchange of exactly one value object between two actors), which as a consequence of the value ports and value interfaces should all happen, or none at all.

The customer acquisition e^{3}value model consists of different value transactions

In Figure 5.3 the value transaction (1) denotes that a reader pays money in order to get a permanent subscription. The value of a permanent subscription for the reader is the consumption of the newspaper service - the newspaper itself, the delivery, etc. Since these values are value objects from the reader point of view we do not break down the value object of a permanent subscription into too many details. Having a look at the scenario path, the OR fork (2) denotes that the permanent reader may also get an advertising gift in order to renew his contract with the newspaper publisher (3). If he does so, the newspaper publisher must obtain an advertisement gift via the value transaction in (4). This could be traced by following the scenario path. Having a look at the test reader, he can be acquired to subscribe as a test reader either via traditional acquisition (e.g. mail, internet, face-to-face) or via telephone consultancy. This scenario is denoted by the OR fork in (5). The value transaction (6) depicts the value transfers between the test reader and the newspaper publisher in case the customer is acquired via the in-house consultancy. The test reader gets a test subscription (the service of getting a daily newspaper for a certain period of time) and an advertisement gift (e.g. an iPod Shuffle). In return, the test reader has to pay a certain fee for consuming this service. Additionally, there is an even more important value for the newspaper publisher that is exchanged between these two actors - the opportunity that the test reader turns into a permanent reader. As soon as the newspaper publisher acquires new test readers he decreases the possibility that the test reader will not become a reader of a competitor. In the print media domain this indicator is called coverage. Thus, the bigger the amount of test readers the newspaper publisher is covering, the bigger is the chance to keep the permanent customer stock and the bigger is the quota for the newspaper publisher on the advertising market.

The same value transaction having the same value transfers as in (6) is depicted in (7). The difference between these two transactions is the scenario path. In order to fulfill the value transaction in (7) the value object of an advertisement gift AND the test subscription brought by the call center is needed. This is denoted by the AND fork in (8). The value transaction (9) depicts the value exchange between the newspaper publisher and the call center. For every test subscription acquired by the call center, the newspaper publisher has to pay money. The value transaction in (10) motivates the fact, that the newspaper publisher needs to validate each address for its actuality. This is done by using a service of an address registry provider. The value for the newspaper publisher is the actual and validated address. For using this service the newspaper publisher has to pay a certain fee.

As already said before, one major output of an e^{3}value model is given by the generation of so-called profitability sheets supported by the e^{3}value tool set. These sheets are used to analyze whether the designed e-business model is economic sustainable or not. The analysis is done by defining different scenarios in regard to different occurrences of cash flows. To achieve this, the value object of each value transfer must be valuated by a certain monetary value (e.g. 29 EUR for an advertising gift). Furthermore it needs to be specified how often a value object is being transferred (e.g. 200.000 transfer occurrences of the advertising gift for all 200.000 test readers). Those

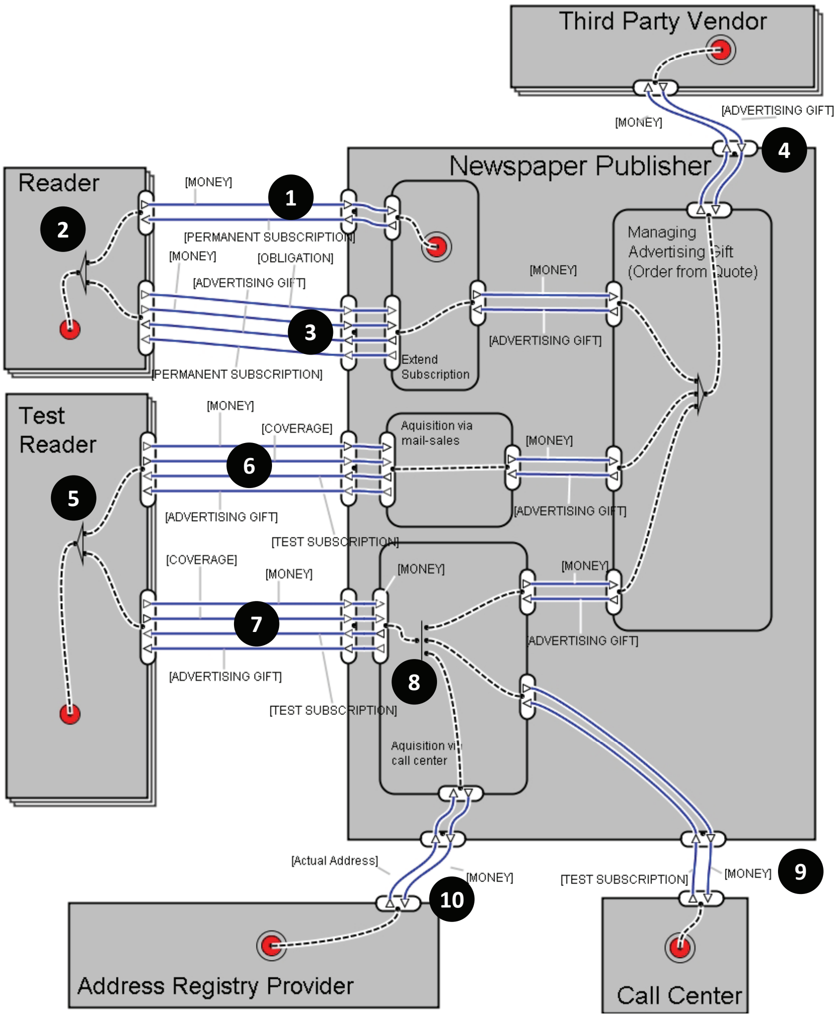


Figure 5.3
e^3value model of the Customer Acquisition use case

assumptions has to be done for each scenario in advance to simulate the profitability.

The ROI of different scenarios

Since we compare the revenue of different scenarios, we need to specify parameters which can be adjusted according to economic characteristics for each scenario (e.g. available budget, targeted number of test-readers, business strategy, etc.). In our scenario, we take the price of an advertisement gift and the number of targeted test-readers as varying parameters. From previous campaigns the project partner knows, that products out of the electronic entertainment section (e.g. MP3-players), which are in general more expensive than other types of goodies, attract more customers than e.g. household articles. Changing those parameters triggers a change of related variables - e.g. the valuation of the *coverage* value object. A decrease of the coverage leads to a decrease of the permanent reader stock caused by the declined possibility for a customer status turn-over (i.e. changing a test-reader subscription to a permanent subscription). Furthermore, in order to compensate the loss of test-readers by a cheaper advertising gift, we need to force the acquisition via call center, which is more expensive than mail acquisition but more efficient. Therefore, we consider two scenarios which are outlined in Table 5.1. Scenario A can be considered as an "high quality" enticement of customers with a more expensive goody than in scenario B. The third column in the table depicts the number of test-readers acquired according to the goody. As we can see, scenario A leads to more test-readers than in scenario B. This results in a higher fraction of the coverage for scenario A. The last column indicates the percentage of call center acquisitions in proportion to the traditional acquisition via mail-sales. For example, in Scenario A, 30% of all test readers are acquired by a call center, the rest is acquired via mail.

Scen.	Goody Price	Test-Readers	Coverage	CC Acquisition
A	52 EUR	200000	60%	30%
B	29 EUR	100000	30%	70%

Table 5.1
Different scenarios for simulating the revenue of a business model

The result is a net value sheet showing the balance (in terms of money) of each actor in the business network. For demonstration purposes we only show the profitability sheet of the newspaper publisher. The net value sheet of scenario A is depicted in Figure 5.4.

The fist column of the profitability sheet shows the value interfaces of the analyzed actor. For a better traceability, the numbers at the left side should help to find the corresponding value transaction in the e^{3}value model in Figure 5.3. The second column lists the value ports of each value interface. The prefix *in* symbolizes an in-going value port and the prefix *out* an out-going value port. The third column shows the valuation of a single value object in EUR. The last columns depicts the total amount in terms of money for each value interface. At the lower right corner, the total net value for the specific actor is shown.

In both profitability sheets (Figure 5.4 and 5.5), the parameters as specified in Table 5.1 for both scenarios are highlighted. The first parameter is the price of a goody, which is represented one-to-one in the profitability sheet (see highlighted value of business transaction 4). The second parameter is the number of acquired test-readers, which is implicitly given in the profitability sheet (see 6 and 7). The reason is, that we define a period of time for the simulation of the profitability. In our example we specified a 1-year period to prove the economic sustainability. In case of the reader this simply reflects a one-year membership with a permanent subscription. In contrary, a test-reader's subscription lasts for a shorter time period - e.g. three month. This means, that a test-reader is acquired four times a year (in total 800.000 test-readers) in order to keep the customer stock of the 900.000 permanent readers a year. Therefore, the number of quarterly test-readers of 200.000 as depicted in Table 5.1 is implicitly given in scenario A - Figure 5.4 - by adding 240.000 occurences for the test-readers acquired via telephone in (6) and 560.000 test readers acquired via mail in (7) divided by four.

The calculation of the profitability sheet is straight forward. The following steps have to be considered: (1) Counting the number of

	Value Interface	Value Port	Occurrences	Valuation	Economic Value	Total
7	{COVERAGE,MONEY ,ADVERTISING GIFT,TEST SUBSCRIPTION}		560000			€ 33.600.000,00
		in: COVERAGE	560000	€ 40,00	€ 22.400.000,00	
		in: MONEY	560000	€ 40,00	€ 22.400.000,00	
		out: ADVERTISING GIFT	560000	€ 0,00	€ 0,00	
		out: TEST SUBSCRIPTION	560000	€ 10,00	-€ 5.600.000,00	
		out: TEST SUBSCRIPTION (expenses)	560000	€ 10,00	-€ 5.600.000,00	
3	{ADVERTISING GIFT,PERMANENT SUBSCRIPTION, OBLIGATION, MONEY}		450000			€ 114.750.000,00
		out: ADVERTISING GIFT	450000	€ 0,00	€ 0,00	
		out: PERMANENT SUBSCRIPTION	450000	€ 30,00	-€ 13.500.000,00	
		out: PERMANENT SUBSCRIPTION	450000	€ 5,00	-€ 2.250.000,00	
		in: OBLIGATION	450000	€ 50,00	€ 22.500.000,00	
		in: MONEY	450000	€ 240,00	€ 108.000.000,00	
6	{COVERAGE,MONEY ,ADVERTISING GIFT,TEST SUBSCRIPTION}		240000			€ 14.400.000,00
		in: COVERAGE	240000	€ 40,00	€ 9.600.000,00	
		in: MONEY	240000	€ 40,00	€ 9.600.000,00	
		out: ADVERTISING GIFT	240000	€ 0,00	€ 0,00	
		out: TEST SUBSCRIPTION	240000	€ 10,00	-€ 2.400.000,00	
		out: TEST SUBSCRIPTION (expenses)	240000	€ 10,00	-€ 2.400.000,00	
10	{MONEY,Actual Address}		240000			-€ 12.000,00
		out: MONEY	240000	€ 0,10	-€ 24.000,00	
		in: Actual Address	240000	€ 0,05	€ 12.000,00	
4	{ADVERTISING GIFT,MONEY}		1250000			-€ 65.000.000,00
		in: ADVERTISING GIFT	1250000	€ 0,00	€ 0,00	
		out: MONEY	1250000	€ 52,00	-€ 65.000.000,00	
1	{MONEY, PERMANENT SUBSCRIPTION}		450000			€ 69.750.000,00
		in: MONEY	450000	€ 240,00	€ 108.000.000,00	
		out: PERMANENT SUBSCRIPTION	450000	€ 55,00	-€ 24.750.000,00	
		out: PERMANENT SUBSCRIPTION (expenses)	450000	€ 30,00	-€ 13.500.000,00	
9	{TEST SUBSCRIPTION, MONEY}		240000			-€ 4.320.000,00
		in: TEST SUBSCRIPTION	240000	€ 0,00	€ 0,00	
		out: MONEY	240000	€ 18,00	-€ 4.320.000,00	
	INVESTMENT				€ 0,00	
	EXPENSES				€ 90.000.000,00	
	total for actor					€ 73.168.000,00

Figure 5.4 *e^{3}value profitability sheet for scenario A*

"occurences" (indicating the number of times a value transfer take place) and "counts" (indicating the number of actors participating in the value network); (2) Assuming economic values and assigning them to value objects; (3) Calculating the profitability sheet by subtracting the value of all outgoing value objects from the value of all incoming value objects per actor.

Scenario A is more profitable than scenario B

The interpretation of the result of the two profitability sheets with different parameter sets can be done via the total revenue for the actor or by finding "economic bottlenecks" within specific value interfaces. In our case, we compared two different scenarios. The scenario with the high quality goody results in a higher revenue as the customer acquisition with the cheaper advertisement gift. Although, in scenario B, the newspaper publisher saves more money by purchasing cheaper goodies, he looses test-readers which results in a decreased coverage. The decrease of the coverage leads to a loss of permanent readers. Thus, scenario A can be considered as the right market mix with the most promising chance for economic sustain-

	Value Interface	Value Port	Occurrences	Valuation	Economic Value	Total
7	{COVERAGE,MONEY ,ADVERTISING GIFT,TEST SUBSCRIPTION}		280000			**€ 16.800.000,00**
		in: COVERAGE	280000	€ 40,00	€ 11.200.000,00	
		in: MONEY	280000	€ 40,00	€ 11.200.000,00	
		out: ADVERTISING GIFT	280000	€ 0,00	€ 0,00	
		out: TEST SUBSCRIPTION	280000	€ 10,00	-€ 2.800.000,00	
		out: TEST SUBSCRIPTION (expenses)	280000	€ 10,00	-€ 2.800.000,00	
3	{ADVERTISING GIFT,PERMANENT SUBSCRIPTION, OBLIGATION, MONEY}		350000			**€ 80.500.000,00**
		out: ADVERTISING GIFT	350000	€ 0,00	€ 0,00	
		out: PERMANENT SUBSCRIPTION	350000	€ 30,00	-€ 10.500.000,00	
		out: PERMANENT SUBSCRIPTION	350000	€ 5,00	-€ 1.750.000,00	
		in: OBLIGATION	350000	€ 25,00	€ 8.750.000,00	
		in: MONEY	350000	€ 240,00	€ 84.000.000,00	
6	{COVERAGE,MONEY ,ADVERTISING GIFT,TEST SUBSCRIPTION}		120000			**€ 7.200.000,00**
		in: COVERAGE	120000	€ 40,00	€ 4.800.000,00	
		in: MONEY	120000	€ 40,00	€ 4.800.000,00	
		out: ADVERTISING GIFT	120000	€ 0,00	€ 0,00	
		out: TEST SUBSCRIPTION	120000	€ 10,00	-€ 1.200.000,00	
		out: TEST SUBSCRIPTION (expenses)	120000	€ 10,00	-€ 1.200.000,00	
10	{MONEY,Actual Address}		280000			**-€ 14.000,00**
		out: MONEY	280000	€ 0,10	-€ 28.000,00	
		in: Actual Address	280000	€ 0,05	€ 14.000,00	
4	{ADVERTISING GIFT,MONEY}		850000			**-€ 24.650.000,00**
		in: ADVERTISING GIFT	850000	€ 0,00	€ 0,00	
		out: MONEY	850000	€ 29,00	-€ 24.650.000,00	
1	{MONEY, PERMANENT SUBSCRIPTION}		350000			**€ 54.250.000,00**
		in: MONEY	350000	€ 240,00	€ 84.000.000,00	
		out: PERMANENT SUBSCRIPTION	350000	€ 55,00	-€ 19.250.000,00	
		out: PERMANENT SUBSCRIPTION (expenses)	350000	€ 30,00	-€ 10.500.000,00	
9	{TEST SUBSCRIPTION, MONEY}		280000			**-€ 5.040.000,00**
		in: TEST SUBSCRIPTION	280000	€ 0,00	€ 0,00	
		out: MONEY	280000	€ 18,00	-€ 5.040.000,00	
	INVESTMENT				€ 0,00	
	EXPENSES				€ 90.000.000,00	
	total for actor					**€ 39.046.000,00**

Figure 5.5
e³value profitability sheet for scenario B

ability. Such a scenario comparison by profitability sheets is not only useful when applying the same e³value model for all scenarios. It is often useful to compare different network constellations by adjusting a set of parameters. This helps to find the right business partner who fits at most to the e-business network.

5.2 REA

REA originates from accounting theory

The REA (Resource-Event-Agent) ontology was developed by William E. McCarthy [118]. The concepts of REA originates from business accounting where the needs are to manage businesses through a technique called double-entry bookkeeping. By the help of this technique every business transaction is registered as a double entry (a credit and a debit) in a balance sheet. REA is using this technique in order to fulfill the basic economic principle for every business transaction - the give-and-take convention, called economic reciprocity. The acronym REA comes from the core concepts *Resource*, *Event*, and *Agent*. The intuition behind these concepts is that every business transaction can be seen as an event where exactly two agents exchange resources. As one may have figured out, e^{3}value concentrates more on the profitability of the e-business system, and REA focuses on issues that may be relevant for the implementation and alignment of the system from an economical point of view.

5.2.1 Basic principles of the REA ontology

The basic REA concepts are illustrated in the cutout of the simplified REA meta model. Figure 5.6 illustrates the simple Resource-Event-Agent structure at the MOF M2 level [134] from a conceptual point of view. We base this meta model on UML class diagrams and use it for defining the concepts of the REA ontology itself. Note, the original REA ontology [118] did not follow a strict meta model. It was constructed by a mixture of UML class diagrams, ER modeling techniques [24] and plain-text table descriptions [41]. Thus, there are no clear specifications on how to develop an REA model at the MOF M1 level. In order to overcome this limitation we formalize REA by means of a UML profile as outlined in Section 5.2.4. All the diagrams shown in the following are already based on UML class diagrams representing cut-outs of the UML profile.

The REA constellation

An *economic exchange* is a collaborative business process where the goal is an exchange of *economic resources* between two *economic agents* where both parties derive higher utility after completion. An economic agent is an individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other individuals or organizations. An economic resource is a good, right, or service of value that has utility for economic agents and is something users of business applications want to plan, monitor and control. An economic event is an occurrence in time wherein ownership of an economic resource is transferred from one economic agent to another economic agent. An economic commitment is a promise or obligation of economic agents to perform an economic event in the future. For the sake of simplicity, concepts such as *typification*, *business locations*, and *economic claims* have been skipped for explaining the basic principles of REA. Those concepts are explained in more detail in the Section 5.2.4.

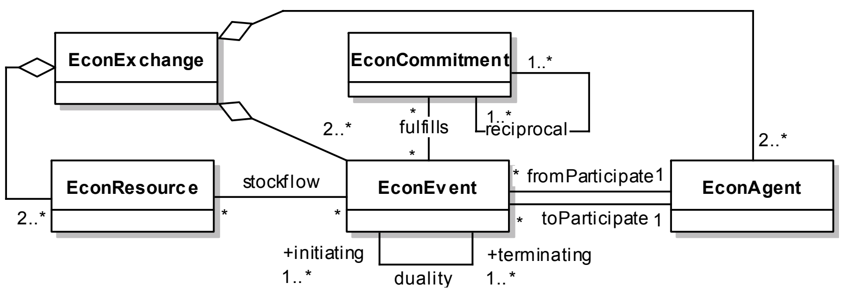


Figure 5.6 *Cutout of the REA meta model*

Reciprocal relations denoting the give-and-take principle

At the instance level, a business transaction or exchange has two

REA constellations (see left part and right part of Figure 5.7) joined together. It denotes that the two parties to a simple market transfer expect to receive something of value in return when they trade. For example, a seller, who delivers a product to a buyer, expects a requiting cash payment in return. In other words, in order to get a resource an agent has to give up another resource. The two REA constellations are connected by a number of relations defined by REA. For instance, economic events resulting in a paired inflow or outflow of economic resources are connected via duality relations. Participation relations between the economic agents and the economic events denote that the economic agent plays either the role of the requester (toParticipant) in the event or the role of the economic agent that has to give up an economic resource (fromParticipant). Finally, reciprocity relations relate commitments with each other denoting that all related commitments must be fulfilled to execute the business exchange.

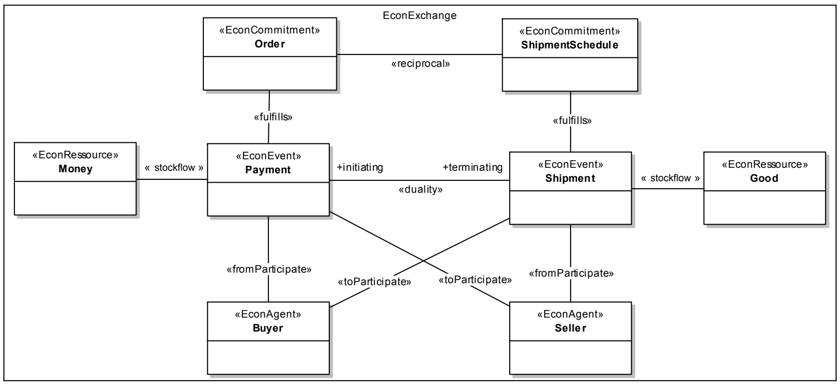


Figure 5.7 Simplified REA example of a buyer-seller scenario

To sum up the basic principles of REA, Figure 5.7 depicts a simplified REA example of the buyer-seller scenario covering the four fundamental questions of a business collaboration:

- **Who** is involved in the collaboration (*Economic Agents* - `Buyer, Seller`)?
- **What** is being exchanged in the collaboration (*Economic Resources* - `Money, Good`)?
- **When** (and under what trading conditions) do the components of the exchange occur (*Economic Events* - `Payment, Shipment`)?
- **Why** are the trading partners engaged in the collaboration (duality relationships between resource flows)?

The answers of those questions help us to specify the requirements for a B2B collaboration model. If the business analyst would disregard such fundamental business rules in large-scale e-business solutions, it is most likely to develop business processes that violate the basic give-and-take principle. REA helps to prevent such failures and ensures the adherence of those domain rules.

5.2.2 REA by example

Within this section, we use again the customer acquisition example to demonstrate how REA can be used on the instance level. However, this time we only use the interaction between the newspaper publisher and the call center for demonstrating the REA - pattern (Resource-Event-Agent constellation). The interaction deals with the acquisition of test subscriptions via telephone calls. In order to trace this interaction back to the e^{3}value model, we refer to (9) of Figure 5.3.

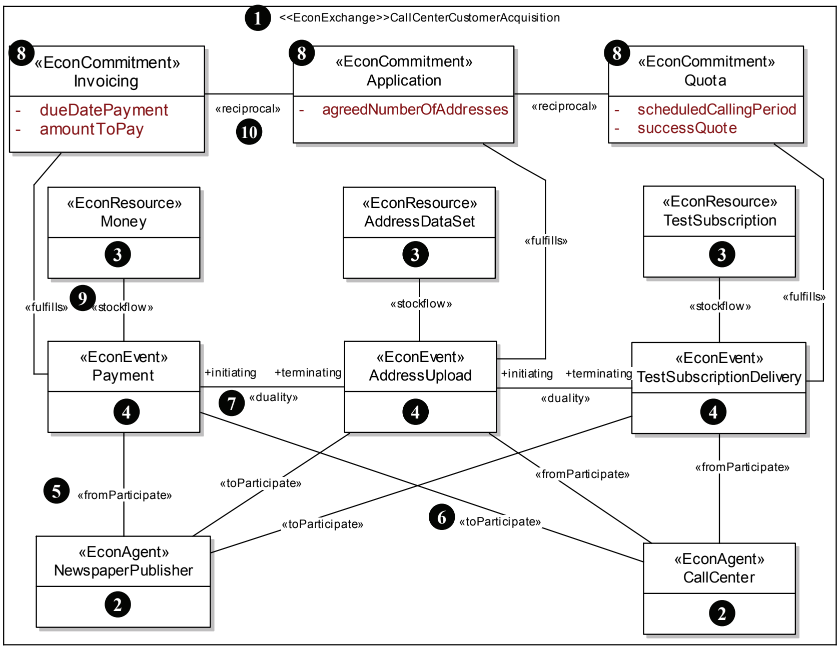


Figure 5.8 *REA example of the Newspaper Publisher and Call Center interaction*

The REA class diagram in Figure 5.8 depicts the economic exchange called *CallCenterCustomerAcquisition* (1). As explained above the exchange happens between the economic agents (2) *newspaper publisher* and *call center*. Both agents have the intention to exchange economic resources (3). These economic resources are the actual objects of exchange. In our scenario the *newspaper publisher* pays the *call center* for the acquisition of new customers. To execute the economic exchange both agents have to participate in adequate events. The economic events denoted by (4) are used to fulfill a resource-flow from one partner to the other one. Since a resource-flow is always a directed association, indicating the role of each participant, the involved agents are connected to an event via (5) *fromParticipate* and (6) *toParticipate* associations. The first one points out that one agent gives up the control over a specific resource, whereas the latter one defines who receives the resource. The *payment* event, for example, provokes that *money* will be transfered *from* the *newspaper publisher to* the *call center*. Since the REA ontology follows the give-and-take principle, each event must have at least one opposite event, which will be fulfilled in return. According to our scenario the *call center* has to fulfill two economic events: *address upload* and *test-subscription delivery*. Both are connected by so called (7) *duality* associations with the *payment* event. This association denotes that if the *newspaper publisher* fulfills a *payment* event, the *call center* has to fulfill *address upload* and *test-subscription delivery*, too. The detailed specifications of the economic events are specified by the means of (8) economic commitments. A commitment can be seen as an obligation to fulfill specific events in the future (9). In the *invoicing* commitment, for example, the *newspaper publisher* is committed to pay a fixed amount of money to the *call center*. The execution of this obligation is done by the *payment* event. The same concept applies for the other commitments - *application* and *quota* and their corresponding events *address upload* and *test-subscription delivery*. The give-and-take principle, as it has already been used with economic events, also applies for economic commitments. The resulting reciprocity is formalized by *reciprocal* associations (10).

All the concepts provided in this example play a major role for our mapping between the different methodologies as provided in Chapter 6. However, REA comprises additional concepts, such as *economic agreements*, *economic claims*, *business locations*, etc. For the sake of completeness, these concepts will be outlined in the next sub-section by presenting the limitations of REA and the UML profile for REA.

5.2.3 Shortcomings of REA

Integrating the original REA concepts into the development process of B2B models is certainly a big challenge. The reason is twofold: first, REA lacks of a precise and formal description on the meta-model layer; second, in contrary to e^{3}value, REA is not made for depicting network constellations in one diagram where more than two business partners are involved having multiple value exchanges. In the following we detail these two limitations and provide a solution in Section 5.2.4 and 5.2.5.

REA has no formalized meta model

Limitation 1 deals with the lack of a formalized meta model. As already said, REA is a business ontology that has been proposed for ontology-driven enterprise systems development. A famous definition of an ontology is given by Gruber: "an ontology is an explicit specification of a conceptualization" [55]. According to Fensel, an ontology is used to represent explicitly the semantics of structured and semistructured information enabling automatic support for maintaining and accessing information [36]. Borst took the definition of Gruber and added, that the conceptualization must be formal and machine-readable [18]. The concepts of REA in its original version are defined at the MOF M1 and MOF M2 level as informal text, table definitions and diagrams, such as ER diagrams, UML class diagrams or other graphical formalism. In other words, the REA ontology is not sufficiently explicit, and not based on a demonstrable shared conceptualization [41].

REA Specialization Module for UMM

However, the REA concept found its place in some standardized specifications. The ISO Open-edi specification [78] uses REA as an ontological framework for specifying the concepts and relationships involved in business transactions and scenarios in the Open-edi sense of those terms. This specification is part of the ISO/IEC 15944-4 standard [79]. Furthermore, the REA ontology definitions are considered by the standardization body UN/CEFACT, which is the originator of UN/CEFACT's Modeling Methodology UMM. It is proposed that the REA ontology should be used to specify the economic requirements for modeling a UMM compliant collaboration model [189]. In order to cope with UMM, the REA ontology must be specified in a formal manner similar to the UMM 2.0 specification. Since UMM is defined as a UML profile and some REA concepts are already specified by UML artifacts (e.g. UML class diagrams), it is straight forward to define REA as a UML profile as well. In Section 5.2.4 we give an overview of the UML profile for REA by means of an excerpt of the REA specialization module for UMM [189].

Multi-party collaborations in REA

Limitation 2 deals with the complex depiction of multi-party collaborations modeled in REA. As portrayed in Figure 5.8, the REA translation for the binary collaboration between the newspaper publisher and the call center becomes already quite extensive. However, the complexity is still manageable, since there are only two business partners involved who exchange only two economic resources. Imagine a network constellation of more than two business partners, which leads to a high number of value exchanges. It is almost impossible to depict the partner network in one diagram, as it is possible by using the e^3value notation. We came to the conclusion, that multi-party collaborations explode geometrically in complexity as the number of trading partners grows. The e^3value methodology (e.g. the diagram of Figure 5.3) obviously provides a more understandable overview of this complexity than the REA class diagrams. In Section 5.2.5 we provide a solution on this limitation by introducing new REA constructs as well as notational adaptions.

5.2.4 A UML Profile for REA

This sub-section deals with limitation 1, which addresses the lack of a formalized notation of REA. A first attempt to formalize the REA ontology has been done by Gailly and Poels [41]. The authors base their approach on the development of a new representation of the REA business ontology by means of OWL. Since OWL is machine-readable their proposed representation is of practical value for those who want to explore REA as a run-time ontology [58]. However, UN/CEFACT decided to integrate the concepts of REA into UMM by means of a well-defined UML profile. This endeavor led to a further specification maintained by the Techniques and Methodologies Group (TMG) of UN/CEFACT. Within this specification the implementation neutral concepts of REA are mapped to a formal UML model. This helps the business analyst to develop an REA model with any UML modeling tool of choice. The specification is called *REA Specialization Module for UMM* and is currently available as a draft version 1.0 [189]. A specialization module for UMM is defined as an add-on concept to the foundation module, which includes the core concepts of UMM [193]. Each specialization module addresses a specialized type of analysis that extends the foundation module at a well-defined extension point for a specific topic. In contrary to extension modules, specialization modules are adding features that are created and maintained by internal members of the UN/CEFACT. We are part of the editing team and developed the UML profile together with the project team leader William E. McCarthy. In the following we introduce the most important parts of this specification, which are expected to become an official UN/CEFACT standard soon.

The REA UML profile is specified by a set of stereotypes

Due to space limitations and for the sake of readability we only provide an excerpt of the UML profile for REA by showing the most important parts. The interested reader is referred to UN/CEFACT's *REA Specialization Module for UMM* document [189]. In Figure 5.9 the conceptual meta model of the UML profile for REA is depicted. The UML classes within this diagram represent the stereotypes of the REA profile. The associations and dependencies between these classes specify the conceptual constraints which have to be considered on instance level. In the following we describe the additional REA concepts specified by the meta model - i.e. the REA stereotypes we did not explained so far in the previous sections.

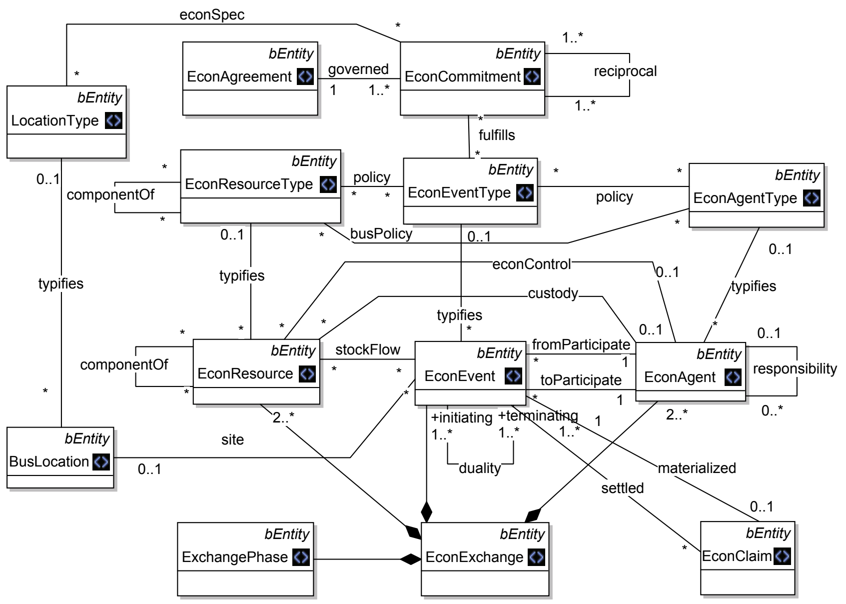


Figure 5.9 *The conceptual meta-model of the UML profile for REA*

Typifying REA concepts denoting the "economic future"

One of the most interesting parts of this meta-model is the introduction of an abstraction layer for the REA pattern (Resource-Event-Agent). This is done by typifying the economic resource, the economic event and the economic agent. The typification is denoted in the meta model by adding the term *Type* to the according REA concept. For instance, the economic resource is abstracted by the stereotype *EconResourceType*. The reason for this abstraction is to represent the abstract structure of economic phenomena. In accounting theory economic phenomena are entities (e.g. policies, real things, etc.) that could be or should happen in terms of the abstract economic future [45]. Abstraction works by abstractly typifying the grouped properties of real things and policies which can then be derived by associating those abstract entities [79]. For example, the type level for an economic resource *advertising gift* might include its designated *package volume* or its *maximum durability*. An *economic event type* is an abstract specification (or type image) of an economic event where its typed or grouped properties can be designated without attachment to an actual, specific occurrence in time. Example of attributes at the type level for events might be *expected duration* or *standard pricing percentage*. An economic agent type is an abstract specification (or type image) of economic agent where its typed or grouped properties can be designated without attachment to an actual agent. An example of an economic agent type might be *qualified buyer* or *approved shipper*.

There are two associations connecting the REA typification concepts. The first one is the *typifies* association, which is the connection between a concrete entity and the abstract specification of its typed or grouped properties. The second one is called *policy* that connects types images and specifies an organizational policy. When types are connected with each other, policy artifacts often emerge, such as the association between an event type and an agent type (such as the managerial position needed to authorize such a class of transactions). This kind of abstract specification is especially important when planning or negotiating an economic exchange. For example, newspaper publisher often specify in advance the types of advertising gifts they desire to be shipped under different delivery categories by different types of shipping agencies.

Economic commitments and agreements

As already mentioned, an *economic commitment* is a promise to execute an economic event at some point in the future. All economic commitments that each in turn individually fulfill compensating events are connected via *reciprocal* associations. In the meta model of Figure 5.9 we also introduce the concept of *economic agreements*. An economic agreement bundles one or more economic commitments similar to a contract. However, an economic agreement is not legally enforceable. It is connected to the economic commitment via the *governed* association.

Introducing the business location and the economic claim

The *business location* is used to specify the site where an economic event takes place. At the first sight, considering such issues in an REA model may appear as an overflow of information. However, the business location indicates the target delivery point of economic commitments and is therefore used to ensure legal aspects of business transactions. In our case, a business location may be an approved kind of delivery warehouse of the newspaper publisher. A further stereotype covering legal aspects of a REA constellation is the *economic claim*. An economic claim is materialized by an economic event which has caused a temporal imbalance in a duality relationship. For example, if an economic event has occurred without its requited correspondence to another economic event. The requited economic event is then responsible for the settlement of the imbalance. The *materialized* association is connected to the initial economic event and the *settlement* association to the terminating one.

The economic exchange phase uses the Open-edi business transaction phases

Finally, the stereotype *economic exchange phase* captures the information about the relating phase of the economic exchange. Thereby, we suggest to use the five different Open-edi business transaction phases (planning, identification, negotiation, actualization and post-actualization). These phases are specified in the ISO/IEC 15944-1

[77] standard. Note, these phases are not related to the "standard" e-commerce transaction phases [122]. The following list provides a summary of these five phases as specified by the standard:

1. **Planning:** In the planning phase, both the buyer and seller are engaged in activities to decide what action to take for acquiring or selling a good, service, and/or right.
2. **Identification:** The identification phase pertains to all those actions or events whereby data is interchanged among potential buyers and sellers in order to establish a one-to-one linkage.
3. **Negotiation:** The negotiation phase pertains to all those actions and events involving the exchange of information following the Identification Phase where a potential buyer and seller have (1) identified the nature of good(s) and/or service(s) to be provided; and, (2) identified each other at a level of certainty. The process of negotiation is directed at achieving an explicit, mutually understood, and agreed upon goal of a business collaboration and associated terms and conditions.
4. **Actualization:** The actualization phase pertains to all activities or events necessary for the execution of the results of the negotiation for an actual business transaction. Normally the seller produces or assembles the goods, starts providing the services, prepares and completes the delivery of good, service, and/or right, etc., to the buyer as agreed according to the terms and conditions agreed upon at the termination of the Negotiation Phase. Likewise, the buyer begins the transfer of acceptable equivalent value, usually in money, to the seller providing the good, service, and/or right.
5. **Post-Actualization:** The post-actualization phase includes all of the activities or events and associated exchanges of information that occur between the buyer and the seller after the agreed upon good, service, and/or right is deemed to have been delivered. These can be activities pertaining to warranty coverage, service after sales, post-sales financing such as monthly payments or other financial arrangements, consumer complaint handling and redress or some general post-actualization relationships between buyer and seller.

UMM business entity states are used in REA

The concepts provided so far only show the meta model from a conceptual point of view - i.e the nomenclature of the stereotypes and the relationships between each other. It does not give any information about the inherited UML artifacts, which is a task of the abstract meta model. However, Figure 5.9 contains information about the super structure of the stereotypes. As we can see in the conceptual meta model, each stereotype is inherited by an abstract superclass. This is denoted by the name of the superclass in italic letters in the upper right corner of each stereotype. The superclass of each class in the conceptual meta model is *bEntity*, which stands for *UMM's business entity*. In UMM, a business entity is defined as a real-world thing having business significance that might be shared among two or more business partners in a collaborative business process [188].

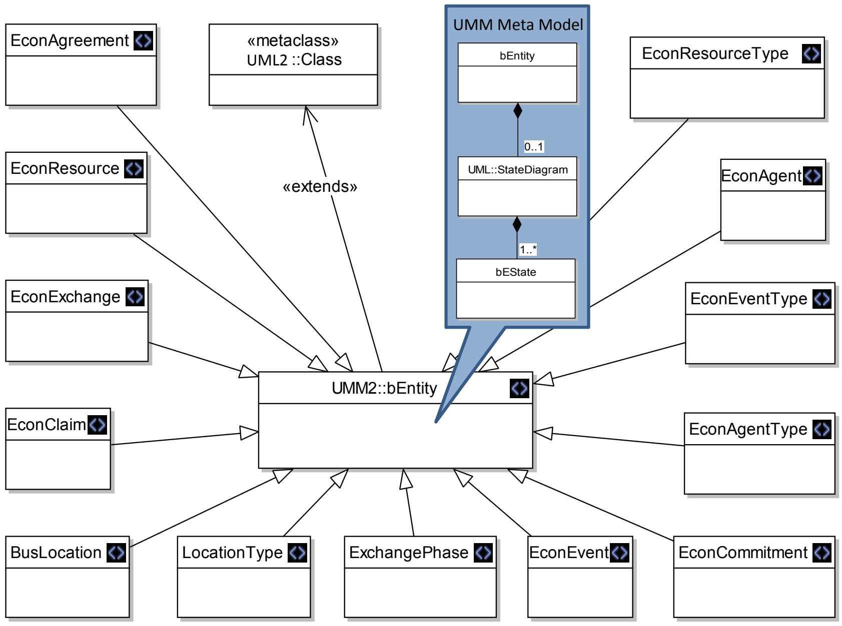


Figure 5.10 *The abstract meta model for the core elements of REA*

Furthermore, business entities are UML classes which have an underlying state machine diagram specifying the life-cycle of the business entity. Figure 5.10 depicts this constellation by combining the abstract meta model of REA with a cut-out of the conceptual meta model of UMM's business entity view. To learn more about UMM's business entity lifecycles we refer to Section 5.3.

The intuition behind the use of business entity lifecycles is similar to UMM's approach of using them by means of state machine diagrams. At the instance level, the business analyst models a state machine diagram for any REA stereotype, which should depict its state changes over the time. By means of this lifecycle he can assign the different states to the according REA concepts. The vantage of this approach is to have different REA constellations modeled according to the state changes of the business entities. Assume the state machine diagram of Figure 5.11 is modeled for depicting the lifecycle of the economic agent *third party vendor*. It defines that the economic agent is first in state *candidate*, then in state *proposed* and finally in state *specified*. According to the Open-edi phase assigned to the *economic exchange phase* the economic agent is in one of those states. For instance, in the *planning phase*, the third party vendor is in state *candidate* and in the *negotiation phase* in state *proposed*. It follows that the modeler has to develop a REA model for each negotiation phase. By using this concept the REA models turn from a rather static perspective into a dynamic perspective by considering the Open-edi phases of a commercial business transaction.

5.2.5 Modeling Multi-Party Collaborations in REA

Improvements on limitation 2

This sub-section deals with limitation 2 regarding the graphical representation of an REA model. The REA ontology assumes that, all multi-party collaborations may be decomposed into a set of corre-

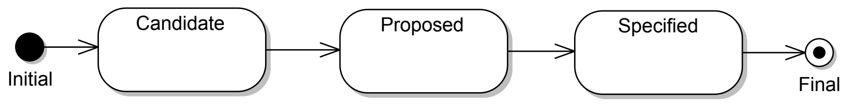


Figure 5.11 The state machine diagram of an economic agent

sponding binary collaborations. The e^{3}value methodology illustrates e-commerce supply chains along similar lines with its graphical modeling tool [47]. In both modeling environments, requited economic exchanges are limited to instances between just two trading partners, even in the cases where buyers and sellers are aided in their e-commerce dealings by third parties like banks, logistics providers, or taxing authorities. The difference is that e^{3}value provides a notation in order to clearly depict the whole e-business network constellation within its graphical modeling tool, whereas an REA model explodes geometrically in complexity as the number of trading partners grows. In order to overcome this limitation we slightly changed the meta model of the REA ontology and introduced new concepts how to decrease the complexity of an REA model in a multi-party collaboration.

Avoiding inconsistencies of multi-party collaborations

The class diagram notation for REA does not satisfy the needs of an appropriate network view as it is provided by e^{3}value . A simple binary transaction in REA exchanging two resources, as portrayed in Figure 5.7, is already quite complex. Extending the sample with another actor, would require at least three more classes. Additionally the corresponding associations between the different agents and events in which they participate have to be modeled as well (four more associations for each event). Since REA has a more IT driven perspective (in regard to the implementation of the IT system with an OO platform or with a relational database) it is necessary to stick to uniquely identifying concepts within an REA model. This uniqueness is not given any longer, if the number of trading partners explodes and the nomenclature for the big amount of *from-* and *toParticipate* associations does not fulfill their distinct purpose. Furthermore, economic events in REA tend to be reused more often within the business network (e.g. Payment). If multiple partners use this event, it is not unique anymore for a specific transaction between two trading partners. Therefore we need, to abstract the economic event in order to foster a reuse of this REA concept. Moreover, the notation of the *duality* relation between two or more economic events results in a high number of connections between these artifacts. If the modeler skips these associations, the only way to identify the give-and-take principle of the REA model is given by transitive implications. For instance, if economic event A has a duality relationship with economic event B and economic event B has a duality relationship with economic event C, then economic event A has a duality relation with economic event C.

The following list summarizes the four main tasks in order to realize multi-party collaborations in REA:

1. Reduce the complexity of REA models regarding its graphical representation if they have more than two participating agents in the business network.
2. Enable the reuse of events for multiple agents.
3. Guarantee that all associations between agents and events can be uniquely identified.
4. Avoid that duality relationships can only be detected by transitive implications.

In [173] we propose three approaches to overcome limitation 2 and provide an unambiguous graphical representation of an REA model. The first one extends the REA meta model by an additional concept, the second one uses an n-ary duality relationship to provide better readability, and the third one uses OCL constraints to overcome the limitations by a formalized language.

Extending the REA meta-model

Event Realizations are used to overcome limitation 2

In order to provide an overall view of the whole B2B model (such as it is possible in e^{3}value) we introduce the concept of *event realizations*. With this additional concept all associations between agents and their corresponding events can be identified exactly. A similar approach is used in UMM where realizations are used to bind a business collaboration to a set of business partners (see Section 5.3).

To implement the concept of economic event realizations into REA we had slightly change the original meta model. As depicted in Figure 5.12 a new element called *EconEventRealization* has been added to the REA meta model. A realization can be seen as a concrete instance of an economic event. The dependency between an event and its realizations is modeled with a *realize* dependency. As a consequence, economic agents are no longer associated with economic events in order to participate in an economic exchange. Instead they have to be connected with economic event realizations via *fromParticipate* and *toParticipate* relationships. The *duality* concept of REA stays the same with the difference that they are now modeled between economic event realizations. This means that in order to fulfill an economic event realization all corresponding realizations have to be fulfilled as well.

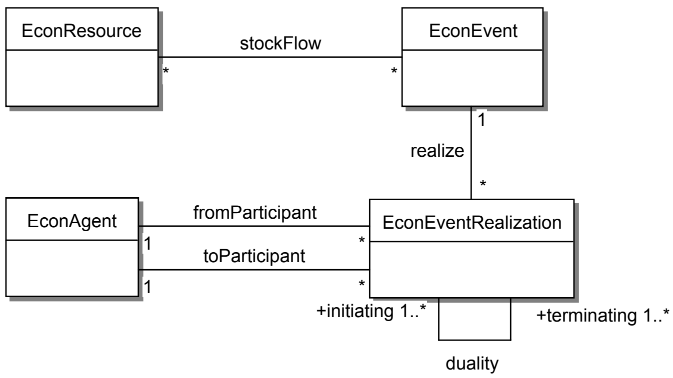


Figure 5.12 *The extended REA meta-model*

Figure 5.13 shows the application of the new REA concepts. We are now able to depict the same model within REA's trading partner perspective as we have shown in the network constellation by e^{3}value . The focus for the explanation of this model should lie on the event realizations. In the model we have prefixed economic event realizations (1) denoting their participating agents in their class names (e.g. *NP_CC_TestSubscriptionDelivery*, *NP_CC_CallServicePayment*). Thereby, NP stands for newspaper publisher and CC for call center. As said before, the participating agents newspaper publisher (2) and call center (3) are connected via *toParticipate* and *fromParticipate* denoting the flow direction of the economic resources. The event realization has the *realize* dependency (4) connected to the economic event test subscription purchase (5). The economic event is now realized by many event realizations and therefore given a more general name compared to the economic event in the REA example of Figure 5.8 - i.e. *test subscription purchase*. The semantics of the economic event concept remains the same. It is connected via the *stockflow* association with the economic resource *test subscription*. However, it is important to underline, that economic events are no longer connected to other economic events denoting the duality concept. Instead, the economic event realizations are connected by duality associations to other economic event realizations (7). As a result, we are able to depict multiple economic agents within one REA diagram.

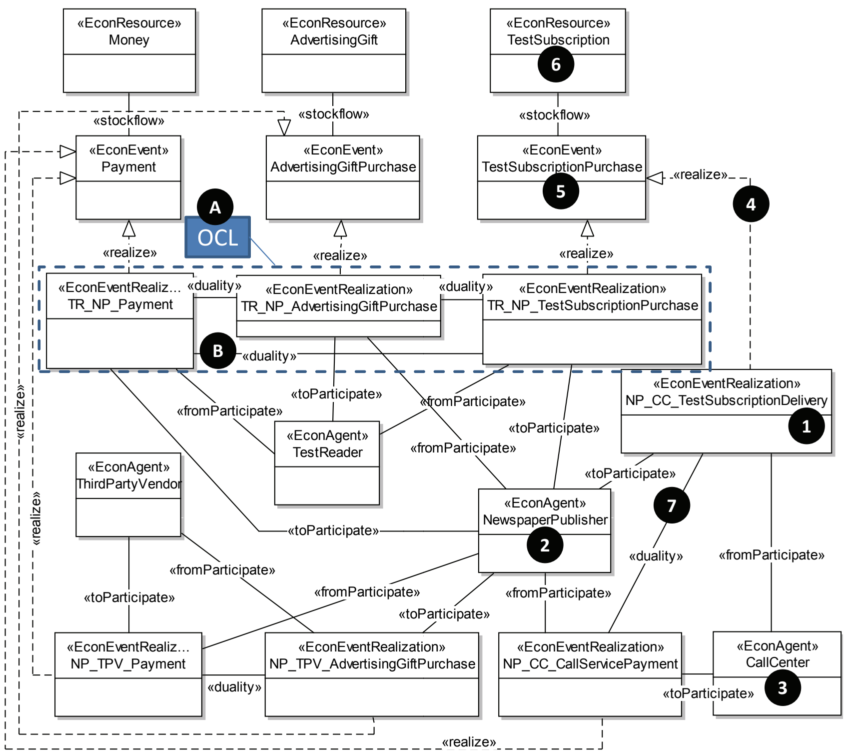


Figure 5.13 *The REA trading partner perspective of the customer acquisition example extended by event realizations*

N-ary duality relationship

In this sub-section we propose a further solution on limitation 2. In more detail, we deal with the problem of the graphical representation of duality relationships in case of three ore more reciprocal economic event realizations. In Figure 5.13 we highlighted such a constellation by a dashed rectangle. Each event realization is connected by a duality relationship denoting that all event realizations must be executed in order to fulfill the economic exchange. If there are more than three event realizations belonging together and representing such an economic behavior, all the elements must be connected by duality relationships. This results in a high number of duality associations.

N-ary duality relationships reduce the complexity of a REA model

In order to reduce the complexity of multiple duality associations we use the concept of an n-ary association. N-ary associations are a well known concepts in UML [135] and are used to model complex relationships between three or more elements. The graphical notation for an n-ary association is a diamond. In REA n-ary associations can be used to model duality dependencies in order to realize the give-and-take principle. Similar concepts have been introduced by Hruby in [68]. The advantage of using an n-ary association is that in fact the number of necessary duality relationships will be decreased by a half. Instead of connecting all economic events with duality associations an n-ary association interlinks all corresponding economic events. Figure 5.14 shows the cutout of the duality constellation of Figure 5.13 (see the dashed rectangle) by using an n-ary association in order to depict REA's duality principle.

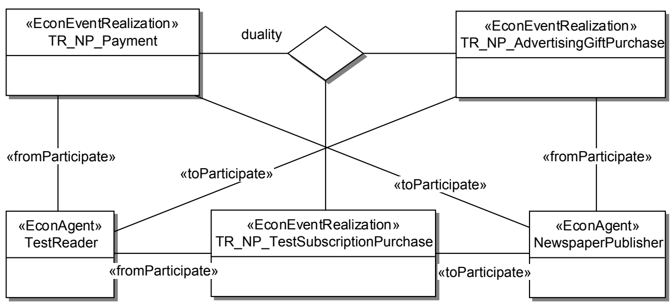


Figure 5.14 *Reduced complexity due to the use of n-ary associations*

OCL denoting duality

Another solution for the complexity problem of REA's graphical representation is the use of the Object Constraint Language (OCL), which is a formal language for the description of rules that apply to UML models [139]. OCL is standardized by the Object Management Group (OMG) and is integrated in the UML standard. As stated in the previous sub-section a duality association relates corresponding economic events. Again, as a consequence of modeling multi-party collaborations this means that the complexity will explode geometrically. Modelers tend to skip transitive duality associations, such as the one annotated by (B) in Figure 5.13 for the sake of complexity reduction. Thus, we only can determine a duality relationship between those connected event realizations via transitive implications.

In fact, deleting this duality association reduces the complexity of the graphical representation but we loose important information for an unambiguous specification of the model.

OCL constraints support unique identifying conclusions

However, the readability of business models for non-technicians is important and the deletion of duality associations (such as the one annotated by B) eases the graphical representation of REA models. For being still uniquely identifiable (e.g. for business analysts toward the derivation of UMM artifacts), the REA model can be constrained with OCL statements to ensure that the event realizations having transitive relationships between each other, belong together. This solution can be considered as an "either-or" solution to the n-ary duality constellation provided in the previous sub-section.

Figure 5.13 shows the note of an OCL constraint (A) attached to the value exchange highlighted by a dashed rectangle. The value exchange consists of the three economic event realizations: *TR_NP_Payment*, *TR_NP_AdvertinsingGiftPurchase*, and *TR_NP_TestSubscriptionPurchase*. The OCL statement in Listing 5.1 specifies that each economic event realization that is part of this economic exchange must be connected by a duality association. Having a deeper look at the OCL code, we determine a simple algorithm for ensuring adherence of the duality rules. The statement in line 76 counts the number of duality associations within the economic exchange. As a matter of fact, this number must be equal to the number of event realizations decreased by 1. Additionally, the code in line 77 ensures that each economic event realization is not associated to itself. If both conditions are fulfilled we can guarantee that all the economic event realizations are connected by duality associations.

```
74 context isEconExchange()
75         let     EconEventRealizationSize:Integer = self.contents.isEconEventRealization()
76         self.contents -> select(a|a.isEconEventRealization()) -> forAll(e|e.associations -> select(d|d.
               isDuality())->size() = EconEventRealizationSize-1
77        AND  e.associations-> select(d.isDuality()) -> forAll (ae|ae.associationEnd<>e))
```

Listing 5.1
OCL constraint for Figure 5.13 A

5.3 UN/CEFACT's Modeling Methodology

UN/CEFACT is known for its work in the area of EDI

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) is an international e-business standardization body known for its work in the area of electronic data interchange (EDI). For analyzing and designing inter-organizational systems, UN/CEFACT has started to work on a development process called UN/CEFACT's modeling methodology (UMM) in 1998. During the course of time this development process has changed considerably. UN/CEFACT's Modeling Methodology (UMM) is an integrated approach for capturing the collaborative space between organizations. The UMM has always been developed according to the business operational view (BOV) of the Open-edi reference model [78], which covers the business aspects such as business information, business conventions, agreements, and rules among organizations.

UMM 1.0 is a UN/CEFACT standard for modeling B2B processes

When UN/CEFACT and OASIS started the ebXML initiative in 1999, UMM just provided guidelines for using the general purpose modeling language UML [135]. It missed a formal customization of the UML meta model. Furthermore, we recognized a step toward service orientation. Being part of the UMM project team, we addressed these challenges by developing a UML profile that integrates service-oriented concepts. A UML profile specifies a set of stereotypes, tagged values and constraints for customizing UML. This means the general-purpose language UML is customized for the specific purpose of inter-organizational systems. Thereby, UMM puts UML in a very strict corset. The resulting artifacts are well defined. Each artifact is restricted to a number of precisely defined modeling elements (stereotypes) and the relationships among them is also fixed. As a consequence, it is easier for software engineers to act upon the resulting artifacts in order to bind their local systems to the public process defined by UMM. We have been the editing team of the resulting specification 'UMM foundation module 1.0' [188] [64] that was finalized in 2006. A survey of different B2B business process modeling languages and standards [39] has shown, that UMM 1.0 is the most complete approach.

There are some shortcomings of UMM 1.0

However, first experiences in applying the UMM in real world projects have shown some shortcomings: First, the current UMM provides rather vague means for modeling business documents. Second, there is a lack of alternative responses in a business transaction. Third, results of a business transaction currently do not propagate changes of business entity states. Fourth, current UMM business choreographies used guards in natural language and, thus, lack information to be machine-processable. Fifth, UMM does not allow to interlink activities of two different business collaborations. Finally, stakeholders have argued against the complex package structure of a UMM 1.0 model.

UMM 2.0 overcomes the limitations

Consequently, we propose new concepts to be adapted by UMM in order to overcome the limitations mentioned in the previous paragraph. We submitted these concepts to UN/CEFACT in order to move the UMM foundation module toward version 2. The goal of this sub-chapter is to demonstrate the adapted UMM 2.0 development process which overcomes the limitations of UMM 1.0. We go step by step through UMM's development process of a rather simple part of the customer acquisition use case scenario - the *order from quote* process. This process is part of the *goody purchase management* and demonstrates the purchase of an advertisement gift between the newspaper publisher and the third party vendor.

UMM consists of three views

In the following, we go through the three main parts of the UMM 2.0: *business requirements view (bRequirementsV)*, *business choreography view (bChoreographyV)* and *business information view (bInformationV)*. The latter view covers the modeling of the business documents that are exchanged. Since the artifacts delivered by the business information view are not considered by the overall approach of this thesis, we only strive this UMM view and just explain the most important parts. To learn more about this view, the interested reader is referred to [98]. Figure 5.15 gives an overview of the UMM 2.0 package structure and the most important artifacts delivered by UMM. The three main views of UMM are highlighted by a rectangle. Note, throughout this sub-chapter the stereotype names are shown in parentheses, which are abbreviated forms of the views' full names. However, in the text we use the full name.

5.3.1 Business Requirements View

The BRV has three sub-views

The business requirements view is the first view to be constructed during the elaboration of a UMM model. The upper part of Figure 5.15 shows the package structure of the *business requirements view* and its three sub-views *business domain view (bDomainV)*, *business entity view (bEntityV)*, and *business partner view (bPartnerV)*. The alphabetically numbered dots associate the example diagrams with the respective packages they belong to, e.g., Figure 5.16 shows the detailed view of A in Figure 5.15.

Business Domain View

Gathering domain knowledge of the domain under consideration

At the beginning of the UMM development process, the business analyst gathers domain knowledge and existing process knowledge of the business domain under consideration. The analyst has to capture the justification of the project and has to determine its scope. He interviews business experts and other stakeholders to get an understanding of the existing business processes in the domain. Thereby, worksheets are a popular mechanism to guide the interview and to capture business know-how. Worksheets are structured forms for the elicitation of specific requirements. It is important that the analyst does not influence the business expert. The interview has to take place in the language of the business domain expert; technical and modeling terms should be avoided. The interviews ensure that all involved parties share a common understanding of the business domain. In this step, the analyst discovers intra- and inter-organizational business processes as existing or desired by individual parties. Similar to the worksheets used in the general approach (Section 4.3.2), UMM worksheets are structured forms for the elicitation of specific requirements [74]. In Chapter 7, those worksheets are explained in more detail and a guide on how to use them for modeling a UMM compliant model is given.

UMM's business areas and process areas

The results of the interviews are transformed into a UML notation. Each worksheet describing a business process results in a *business process uses cases (bProcessUC)*. Business processes are classified according to UN/CEFACT's Catalog of Common Business Processes (CBPC) [192], the Supply Chain Reference Model (SCOR) [184] or Porter's Value Chain (PVC) [160]. Classifying business processes facilitates the understanding of the business domain as well as its scope. A hierarchical composition of business areas and process areas is used to represent the classification as shown in Figure 5.15. In this example we only show one *business area Procurement/Sales* which includes the *process area identification* and *negotiation*. In the full customer acquisition example, the *business domain view* comprises additional *business* and *process areas*.

Business process use cases

The business process use case *purchase goody* is assigned to the process area negotiation within the business area procurement/sales (A in Figure 5.15). The corresponding use case diagram is shown

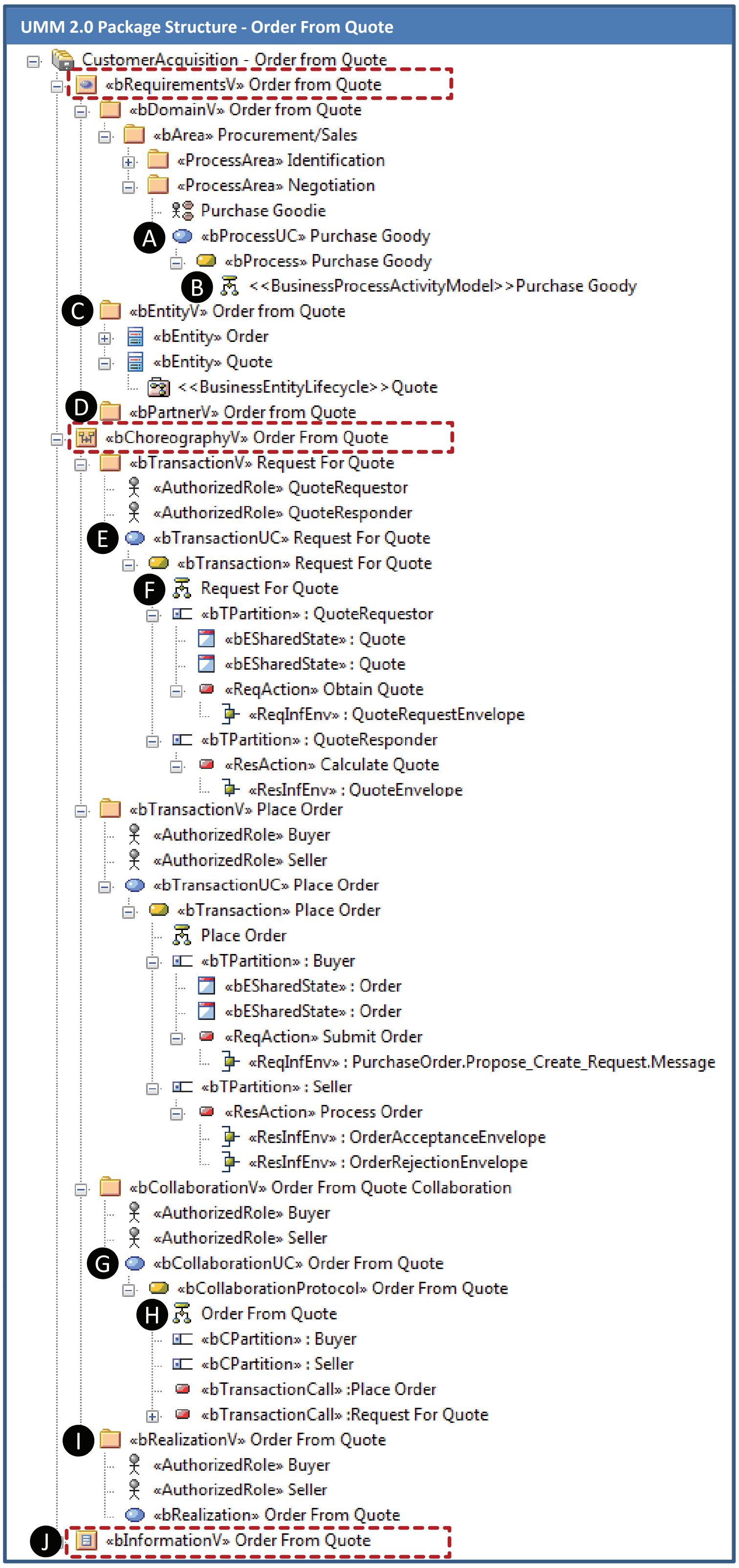


Figure 5.15
Request for quote example - UMM 2.0 Model Structure

in Figure 5.16. In general, business partners participating in the business processes and stakeholders who have an interest in them are associated to the business process use cases. In our example, the *business partners newspaper publisher* and *third party vendor* participate in *purchase goody*.

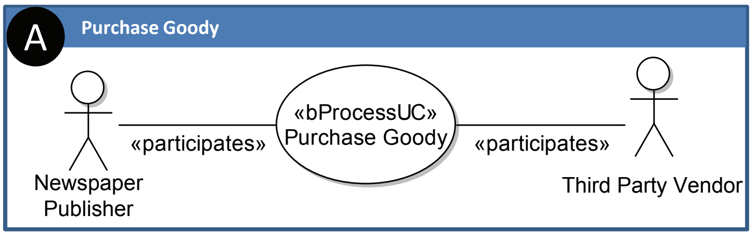


Figure 5.16 *Business Process Use Case with Business Partners*

Business processes are detailed by an activity diagram

Once all business processes are discovered, a review cycle is initiated in order to identify those who in fact have a relevance for the business collaboration to be developed. These business processes are further detailed by an activity diagram according to the requirements specified in the respective worksheet. The activity diagram becomes a child of the *business process use case*. In our example, we show the activity diagram for purchase goody in Figure 7.17. According to Figure 5.15, this activity diagram (B) is a child of the corresponding business process use case (A).

The following business semantics are kept in the activity diagram: the *newspaper publisher* requests a quote from the *third party vendor*. The *third party vendor* processes the quote request and returns the quote information to the newspaper publisher that prepares the order and finally submits the order to the *newspaper publisher*. The same business process happens between the *third party vendor* and the *manufacturer*. For the sake of simplicity this scenario is not depicted in Figure 7.17.

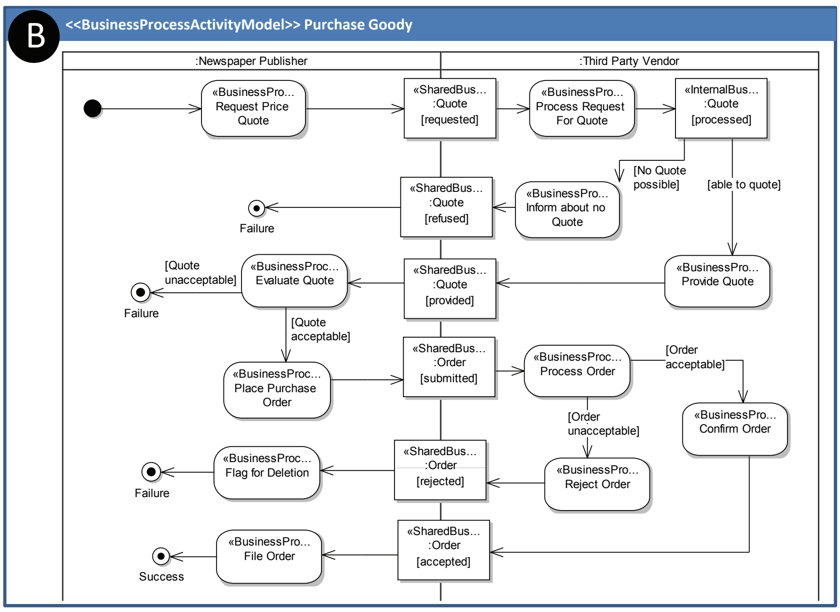


Figure 5.17 *Business Process Activity Model*

The synchronization of business entity states

The exchange of information must always lead to a synchronization of changed *business entity states* at each partner's side. Thus, the object flow between activities is denoted by a *shared business en-*

tity state, which is further discusses below in the sub-section on the business entity view. The concept of *shared business entity states* denotes the need for communication between business partners. Thus, *shared business entity states* are a strong indicator for requiring information exchange in later designed business collaborations.

Business Entity View

Definition of a business entity

A *business entity* is a real-world thing having business significance that is shared between two or more business partners in a collaborative business process (e.g. "order", "account", etc.). In our example, the information exchanged is about the *business entities quote* and *order*.

The business entity lifecycle defines the different states

A *business entity lifecycle* is described by a UML state diagram as part of the business entity view (cf. C in Figure 5.15). It delineates the states a *business entity* may obtain as well as the flow between them. The lifecycle is designed in accordance with the activity diagrams in the *business domain view*. The object flow in the activity diagrams is based on *shared business entity states* (cf. Figure 7.17). Each *shared business entity state* reflects a *business entity state* in the *business entity lifecycle* (cf. Figure 7.19). Thus, the order of changing *business entity states* in the activity diagrams must be kept in the *business entity lifecycle*.

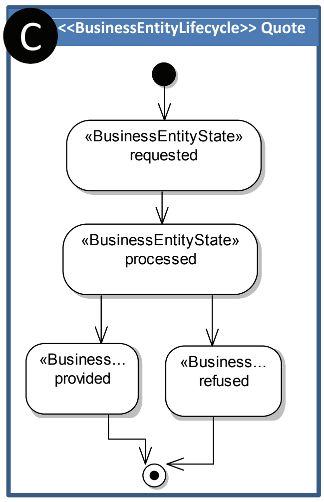


Figure 5.18 *Business Entity Life Cycle: Quote*

The *business entity lifecycle* depicted in Figure 7.19 represents the states of the *business entity quote*. A *quote* object is created with the state *requested*. After it is in state *processed* it is either set to the state *provided* or *refused*. These four *business entity states* are referenced by the *shared business entity states* of the activity diagram in Figure 7.17.

Business partners are captured in the BDV

Business partners identified in the business requirements view are modeled in diagrams that belong to the business domain view. However, for the sake of an easier re-use, business partners and stakeholders are kept in a dedicated container - the *business partner view* (D in Figure 5.15). The business partner view may also be used to analyze relationships between the business partners and/or stakeholders in optional role models, which are not further elaborated here.

5.3.2 Business Choreography View

The BCV consists of three sub-views

In the *business choreography view* the analyst builds upon the previously created artifacts in order to develop models describing a global choreography. According to Figure 5.15, it consists of three sub-views: *business transaction view (bTransactionV)*, *business collaboration view (bCollaborationV)* and *business realization view (bRealizationV)*. The *business transaction view* models the basic-building blocks of a choreography which correspond to a single business document exchange and returning an optional business document as a response. The *business collaboration view* models a global choreography built by these basic building blocks. A *business realization view* is used if the same choreography is realized between different set of business partners.

Business Transaction View

Business transactions are the basic building blocks of the choreography

The basic building blocks of a UMM choreography are *business transactions*. The goal of a business transaction is synchronizing the business entity states between two parties. Synchronization of states is either required in an uni-directional or in a bi-directional way. In the former case, the initiator of the *business transaction* informs the other party about an already irreversible state change the other party has to accept - e.g., the notification that a shipment has arrived. It follows, that responding in such a scenario is neither required nor reasonable. In the latter case, the initiating party sets a *business entity* to an interim state and the responding party decides about its final state - consider a request for a quote that the responder might either provide or refuse.

Business transactions are either uni- or bi-directional

The synchronization takes place by exchanging business information. According to the definitions above, an exchange takes always place between exactly two parties. It is either a uni-directional exchange or a bi-directional exchange including a response. The activity diagrams created in the business domain view (cf. Figure 7.17) already indicate the need for exchanging business information to synchronize business entities by the concept of *shared business entity states*. However, these activity diagrams are not necessarily consolidated between the various parties and are just used for requirements elicitation. The business transaction has to present a consolidated view on the basic building blocks. Thus, it has to identify the commonly agreed *shared business entity states* and, possibly, aggregate two of them in a bi-directional business information exchange.

Authorized roles are involved in the business transactions

This identification and consolidation process leads to a number of *business transaction use cases* and the two *authorized roles* participating in the use case. According to Figure 5.15, each *business transaction use case* (E) and the two participating *authorized roles* are placed in their own *business transaction view*. Figure 5.19 depicts the *business transaction use case request for quote* which involves the participating *authorized roles quote requestor* and *quote* *responder*. Note, we use the abstract concepts of *authorized roles* instead of *business partners*, because *business transactions* and their use cases may be realized between different sets of *business partners*.

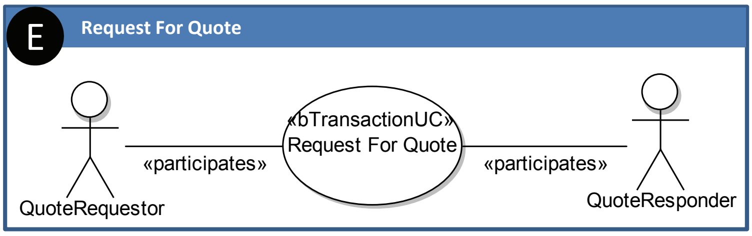


Figure 5.19
Business Transaction Use Case

Business transactions are activity diagrams

The requirements of a *business transaction* are further elaborated using the concept of an activity diagram. For each *business transaction use case* an activity diagram is created and placed as a child underneath the respective use case, e.g., in Figure 5.15 the *business transaction use case request for quote* (E) is refined using the activity diagram (F).

Business transaction patterns

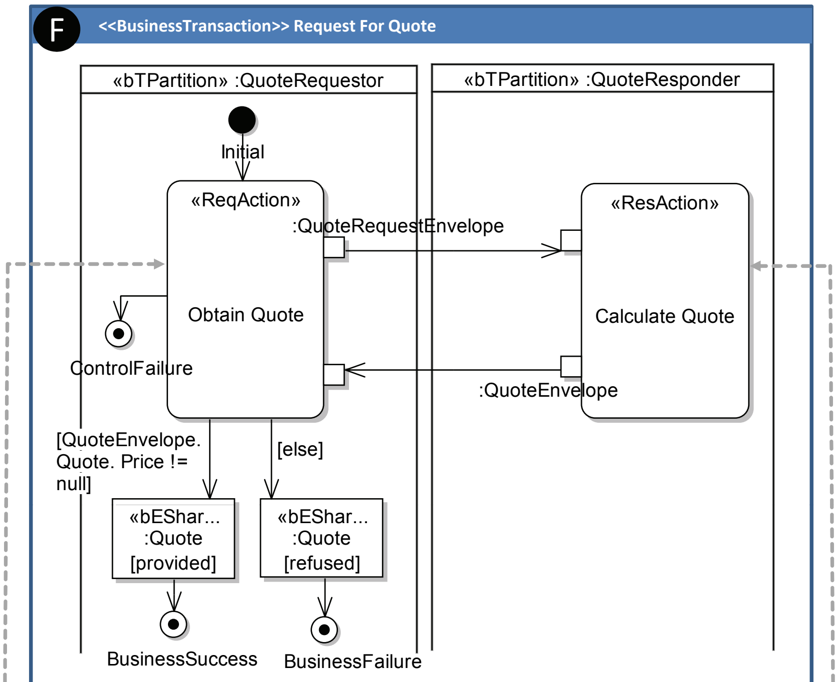
The main purpose of a business transaction activity diagram is to formally describe a UMM business transaction. It is important to notice, that a *business transaction* always follows the same pattern. The business transaction pattern thereby defines the type of a legally binding interaction between two decision making applications as defined in Open-edi [78]. We distinguish between two one-way (information distribution, notification) and four two-way (query/response, request/response, request/confirm, commercial transaction) types of business transactions.

The basic building blocks of a business transaction

The basic building blocks of a business transaction are activity partitions, which are used to denote the authorized roles, participating in the transaction. Furthermore, a business transaction contains exactly two actions - a requesting action and a responding action - one on each business partner's side. Between the different actions the business information exchange is denoted using the concepts of object flows and action pins. There is always exactly one object flow from the requesting action to the responding action. In a one-way business transaction there is no flow in the reverse direction. In case of a two-way business transaction there are one or more object flows in the reverse direction. In case of two or more object flows they are considered as alternatives. The type of the action pins in the business transaction is set using business documents from the business information view.

Figure 5.20 shows the *business transaction request for quote*. On the left hand side the *business transaction partition (bTPartition)* of the requesting role is shown and on the right hand side the one of the responding role. The type of a *business transaction partition* is determined by the *authorized roles* participating in the *business transaction use case*, which the *business transaction* refines. In Figure 5.20 the type of the requesting partition is set by the *authorized role quote requestor* and the type of the *responding partition* is set by the *authorized role quote responder*.

The *requesting partition* contains a so called *requesting action (ReqAction)* and the *responding partition* a *responding action (ResAc-*



Obtain Quote (Activity)	
isAuthorizationRequired	true
isIntelligibleCheckRequired	true
isNonRepudiationReceiptRequired	false
isNonRepudiationRequired	false
retryCount	0
timeToAcknowledgeProcessing	PT0D10H0M0S
timeToAcknowledgeReceipt	PT0D6H0M0S
timeToRepsond	PT0D12H0M0S

Calculate Quote (Activity)	
isAuthorizationRequired	true
isIntelligibleCheckRequired	true
isNonRepudiationReceiptRequired	false
isNonRepudiationRequired	false
timeToAcknowledgeProcessing	PT0D24H0M0S
timeToAcknowledgeReceipt	PT0D14H0M0S

Figure 5.20
Business Transaction Request for Quote

tion). In the example shown in Figure 5.20 the *quote requestor* starts the *business transaction* by sending a *quote request envelope* to the *quote responder*. Since the transaction is bi-directional the *business entity quote* is set to an interim state. Depending on the response of the *quote responder*, the *business entity* is set to its final state.

After the *quote responder* has processed the request from the *quote requestor* he replies with a *quote envelope*. In the quote requestor's partition two *shared business entity states quote* are shown together with guard conditions leading to the *shared business entity states*. Depending on the reply of the *quote responder* the *shared business entity state quote* is either set to the final state *provided* or to *refused*. In case a control failure occurs during the transaction the *business transaction* results in a control failure as shown on the left hand side of Figure 5.20.

The tagged values of a business transaction

At the lower side of Figure 5.20 the tagged values containing the different business signal information of the requesting and the responding action are shown e.g. time to acknowledge receipt indicates the maximum time within the responding party has to confirm a successful/unsuccessful syntax, grammar, and sequence validation. Further tagged values are: is authorization required, is non-repudiation required, time to perform, time to acknowledge receipt, time to acknowledge acceptance, is non-repudiation of receipt required and retry count. These tagged values are considered as self-explanatory and are explained in detail in the UMM specification [188].

As shown in Figure 5.15 the simplified order from quote example consist of exactly two business transactions: *request for quote* (Figure 5.20) and *place order*. The latter one is a two-way transaction as well and is not explained in detail here.

Business Collaboration View

A business collaboration choreographs the execution order of business transactions

After the identification of the different *business transactions* the modeler continues with creating *business collaborations*. A *business collaboration* choreographs the execution order of different *business transactions* and *business collaborations* (since *business collaborations* can be recursively nested).

Each *business collaboration view* contains exactly one *business collaboration use case* and two *authorized roles* participating in the use case (G in Figure 5.15). By definition a *business collaboration* consists of different *business transactions* and/or *business collaborations*. Included *business transactions*/*collaborations* are denoted using the concept of *include* dependencies. Each included *business transaction* is defined in its own *business transaction view* and each included *business collaboration* is defined in its own *business collaboration view*.

The order from quote example includes two business transactions

As shown in Figure 5.21 the *business collaboration use case* order from quote includes two *business transactions*, namely request for quote and place order. Again the abstract concept of *authorized roles* is used instead of *business partners* because *business collaborations* may be realized between different sets of *business partners*.

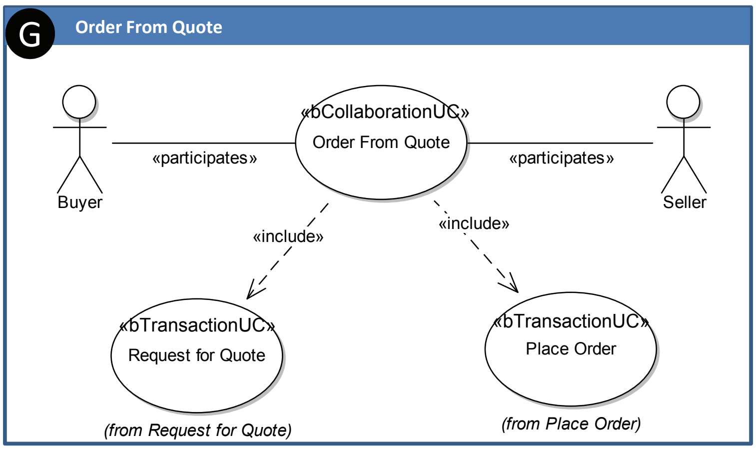


Figure 5.21 Business Collaboration Use Case

The business collaboration protocol specifies the sequence of the business transactions

Similar to the concept of a *business transaction use case* a *business collaboration use case* is further elaborated using the concept of a so called *business collaboration protocol*. For each *business collaboration use case* a *business collaboration protocol* is created and placed as a child under the respective use case e.g. in Figure 5.15 the *business collaboration use case* order from quote (G) is refined using the *business collaboration protocol* (H). Consequently a *business* *collaboration use case* is always the parent of exactly one *business collaboration protocol*.

Business collaboration protocol is built by using business transaction calls and business collaboration calls

The main aim of a *business collaboration protocol* is to describe a *business collaboration* on a formal basis. Thereby, a *business collaboration protocol* is built using *business transaction calls* and *business collaboration calls*. A *business transaction call* calls a *business transaction* and a *business collaboration call* calls a *business collaboration*. In order to depict the *authorized roles* participating in a *business collaboration*, a *business collaboration protocol* uses the concept of partitions. For each *authorized role* exactly one partition is created. In some cases an *authorized role*, during the course of a *business collaboration*, might internally execute another *business collaboration*. In this case the concept of so called *nested business collaboration* is used. *Nested business collaborations* are defined in another *business collaboration view*. In order to denote the execution order of different *business transaction calls* and *business collaboration calls* the concept of *initFlows* and *reFlows* is used. Thereby an *initFlow* can either lead to a partition or - in case a *nested collaboration* is used - to a *nested business collaboration*. The same applies to *reFlows*. Guard conditions attached to the different object flows within the *business collaboration protocol* determine the exact execution sequence.

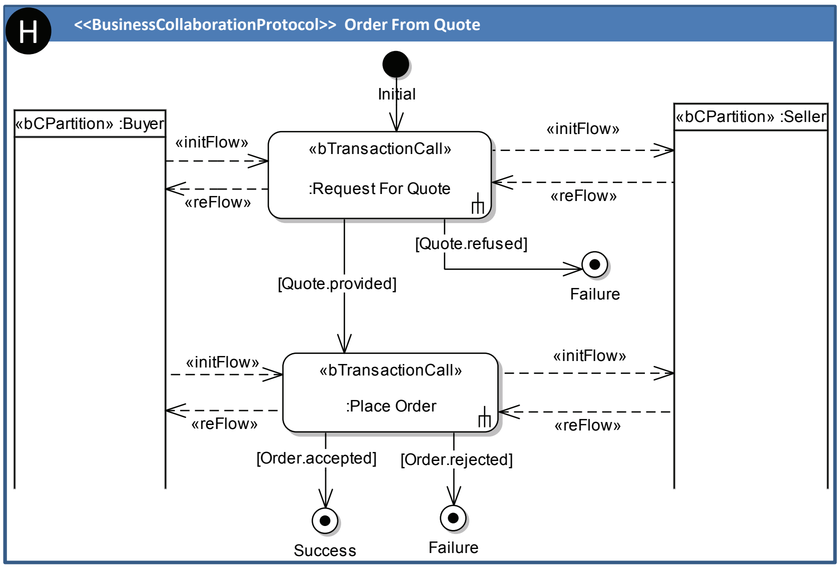


Figure 5.22 *Business Collaboration Protocol Order from Quote*

An example of a business collaboration protocol

The *business collaboration protocol* in Figure 5.22 defines the exact choreography of the *order from quote* collaboration. Using the concept of two *business collaboration partitions (bCPartition)* the two *authorized roles buyer* and *seller* participating in the *business collaboration* are shown. The *business collaboration order from quote* starts with the *business transaction request for quote*. The *initFlow* dependency between the *buyer* and the *business transaction call request for quote* in Figure 5.22 indicates, that the *buyer* initiates the *business transaction*. Since there is a *reFlow* dependency from the *seller* to the *business transaction call* and the *buyer*, the *business* *transaction* is a two-way transaction. If the *seller* refuses the quote, he sends a *quote rejection envelope* to the *buyer*.

If the *business transaction request for quote* fails, because the *seller* refused the quote, the *business collaboration order from quote* also fails. In Figure 5.22 this is indicated by the control flow with the guard condition *Quote.refused* leading from the *business transaction call* to the final state *Failure*. Please note, that the guard conditions of the control flows directly match to the *shared business entity states* of the underlying *business transaction* (see Figure 5.20).

In case the *business transaction request for quote* was successful, the guard condition *Quote.provided* evaluates true and the *business transaction place order* starts. Similar to the previous *business transaction*, the order is either accepted or rejected. Thus, the *business collaboration* finally ends with the *business entity order* being in either in state *accepted* or *rejected*.

Business Realization View

Business realizations foster re-use

We have seen so far, that *business transactions* and *business collaborations* are executed between *authorized roles* instead of specific *business partners*. By using the concept of authorized roles, the same business collaboration/transaction may be re-used between different sets of specific business partners. This enables the standardization of business collaboration models and, in turn, fosters re-use, which is one of the key goals of UN/CEFACT.

In order to bind a *business collaboration* (and implicitly the *business transactions* it consists of) to a set of *business partners*, UMM provides the concept of so called *business realizations*. Figure 5.23 shows a possible *business realization* for the *business collaboration order from quote*.

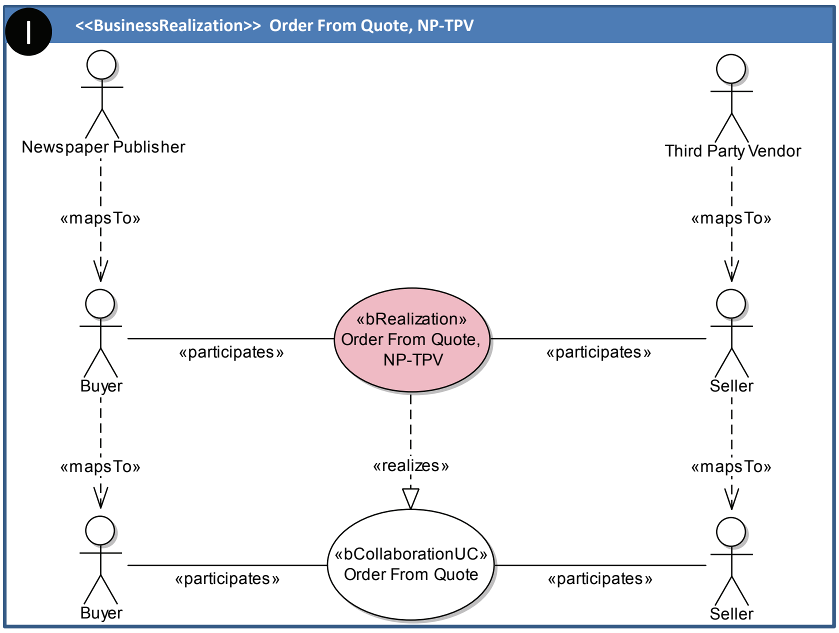


Figure 5.23
Business Realization View

The business realization realizes a business collaboration

On the lower left hand side of Figure 5.23 the *business collabo-*

ration order from quote is shown between the two *authorized roles buyer* and *seller*. A *business realization* is connected to a specific *business collaboration use case* using a *realize* connection. In Figure 5.23 the *business realization order from quote, NP-TPV realizes* the *business collaboration use case order from quote*. The *business realization* again has two *authorized roles buyer* and *seller*. Finally, *business partners* identified in the *business partner view* are bound to *authorized roles* by connecting them via *mapsTo* dependencies.

The business collaboration can also be used between the third party vendor and the manufacturer

The benefit of this concept is easily demonstrated by our example. As we learned in Section 5.3.1, our example business collaboration between the *newspaper publisher* and the *third party vendor* may also be performed between the *third party vendor* and the *manufacturer*. This issue is modeled by introducing an additional business realization, which realizes the *business collaboration use case order from quote*. In this case it is performed between the *third party vendor* and the *manufacturer*. Thus, the concept of *business realizations* evidently contributes to the re-use of modeling artifacts. With the completion of the *business realization view* the business modeler has finished the business process perspective of the UMM.

5.3.3 Business Information View

The business information view is used to model the exchanged business documents

The final view of UMM is the so called *business information view* (see J of Figure 5.15). Within the *business information view* the business documents, which are exchanged in the different *business transactions* of UMM are defined. UMM does not mandate to use a specific business document modeling technique in this view, but leaves it up to the modeler which technology to use. It is, however, strongly suggested to use UN/CEFACT's Core Components [191] for the modeling of the exchanged business documents. Core components are syntax independent, reusable building blocks, standardized by UN/CEFACT for the modeling of business documents. In order to allow for an integration of core components into a UML modeling tool, UN/CEFACT has developed the UML Profile for Core Components (UPCC) [190].

A UML profile for Core Components

With the help of UPCC, *core components* and so called *business information entities* are modeled. Thereby *core components* are context independent and generic building blocks for business documents. If a *core component* is used in a certain business scenario it becomes a so called *business information entity*. A *business information entity* has a certain context and is always derived from the underlying *core component* by restriction. Hence, the generic *core component's* building blocks are tailored to the specific needs of a certain business scenario.

Business document modeling is out of scope for this thesis

As already mentioned, the *business information view* is neither considered in the mapping of our business modeling based approach, nor is it necessary to define worksheets for the artifacts in this view. Due to the fact that document modeling is out of scope for this thesis we refer to [98] for more details on UMM's *business information view*.

In this chapter we have shown the essentials of the interlinked languages for engineering inter-organizational systems. Furthermore, we provided solutions in order to overcome limitations of those methods. The next chapter demonstrates a conceptual mapping between the introduced languages and a formalization of the transformation.

6 Business Modeling based Approach

A mapping from e^{3}value and REA to UMM

In this chapter we propose a mapping between the business modeling layer and the business process modeling layer. In more detail, we provide a conceptual mapping on how to get from the business modeling ontology e^{3}value to the business modeling ontology REA and further on to the business process modeling methodology UMM. Moreover, we formalize the mapping by the use of the model-to-model transformation language ATL (Atlas Transformation Language) [9]. The basic idea of this multi-layered approach, in which we propose to use different ontologies and techniques for the development of maintainable inter-organizational systems has been published in [70] [173] [174].

The remainder of this chapter is structured as follows: in Section 6.1 we give a short overview of the business modeling approach. In Section 6.2 we provide a conceptual mapping from e^{3}value to REA and in Section 6.3 we detail the mapping from REA to UMM. Finally, we define the mapping rules in a formalized way by the use of ATL in Section 6.4.

6.1 The business modeling based approach at a glance

The importance of business models

Figure 6.1 depicts the different methods, their perspectives and their goals. Before the business analyst starts developing a business model for a B2B system, he needs to define a partner network. It is not possible to define a partner constellation without knowing the economic input and output factors of the business partners under consideration. The network perspective of the business model deals with values that are exchanged between different possible business partners. Therefore, we propose to start with e^{3}value for getting a first overview of the economic values exchanged in the network. Furthermore e^{3}value offers the possibility to prove the economic sustainability of the business idea by quantifying the net value flow for each actor in the value web. Whereas e^{3}value concentrates more on the profitability of the IT system, a business model is needed that focus on economic issues that may be relevant for the implementation and alignment of a B2B system. In other words, to ensure that business processes beneath do not violate the domain rules, i.e. to fulfill the basic economic principle for every business transaction - the give-and-take convention, called economic reciprocity. These requirements are manifested by the trading perspective. The REA ontology assumes in its original version that all multi-party collaborations may be decomposed into a set of corresponding binary collaborations. It is the modeler's choice to use this original notation, or to use our proposed improvements for depicting multi-party collaborations with REA (c.f. Section 5.2.5) as it is possible with e^{3}value and its graphical modeling tool [47]. As a matter of fact, e^{3}value and REA are perfect candidates that complement each other for developing the business models of a B2B system. Hence, we introduce conceptual rules for mapping an e^{3}value model to an REA model.

Conceptual mapping rules for interlinking the different models

As soon as the modeler has developed the business model, he is able to break down the requirements from an economical point of view into the process perspective. Talking about the business process perspective, UMM is the most promising methodology that is compliant to REA for integrating the requirements gathered in the trading perspective [189]. UMM is a UML modeling approach to design the business services that each business partner must provide in order to collaborate. It concentrates on the flow of interactions between the collaborating business partners. However, it does not address their private processes. In general, the execution of an inter-organizational business process depends on commitments established between the participating partners. UMM is used to model these procedural and data exchange commitments that must be agreed upon between the business partners at design time of the inter-organizational business process, i.e. before executing the process. REA captures such commitments on the business modeling layer and filters out which economic exchanges are candidates for business processes in UMM's business requirements view. Thus, we provide conceptual mapping rules that help the business analyst to develop a UMM model based on an REA model. Moreover, the goal of this conceptual mapping between REA and UMM is to provide four types of specific guidance to UMM users as they develop their collaboration models: an ontology-based methodology for developing the (1) classes for *business entities*, (2) state machine life cycles for *business entities* within the *business entity view*, (3) business partners of the *business partner view* and (4) the *business process use cases* they are participating. All these UMM artifacts are part of the *business requirements view*. Therefore, we only provide a mapping to the business requirements view and not to artifacts that represent concepts of the business choreography view.

The conceptual mapping rules do not automatically generate fully-fledged modeling artifacts

Using the conceptual mapping rules does not imply, that one is able to generate a whole UMM model out of the trading perspective. However, it is possible to semi-automatically generate a skeleton for a UMM compliant B2B model. In general, this applies for all transitions between the different perspectives. The reason is, that each perspective needs distinct business information that is only relevant for a specific methodology. For example, the value of *customer loyalty* gathered in e^{3}value is not needed anymore in REA, and the actual exchange of money (payment) defined in REA is in most of the cases no candidate for a one-to-one mapping to a *business process use case* in UMM. Hence, it is required that the business analyst has extensive

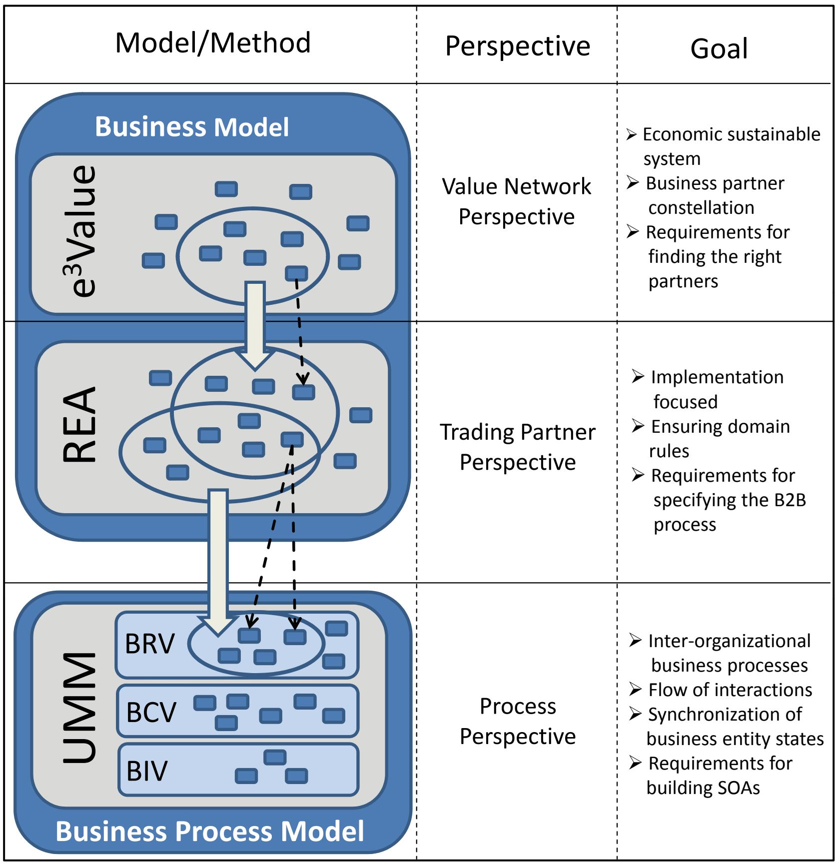


Figure 6.1
Overview of the business modeling approach

knowledge about the semantics of the domain under consideration. The worksheet driven approach described in the next Chapter 7 also helps the modeler to keep track of the gathered requirements of each perspective in a structured way.

Not all concepts of each method are mapped between the different models

Having a deeper look at Figure 6.1, the small rounded rectangles within each modeling method denotes a specific concept or element that is specified by its meta model. For instance, a concept of e^{3}value are the *value object* and the *value port* (c.f. Section 5.1.1). As mentioned in the previous paragraph, by our proposed business modeling based approach we do not map every single concept between the different perspectives. Some of the concepts are simply not needed one layer beneath and some concepts are not mappable at all. The circle within each model in Figure 6.1 depicts schematically that only a sub-set of the concepts of a model will be mapped to the successive model. As examples of the upper layer, e^{3}value concepts that are not mapped to REA are the *scenario path* including all its sub-elements, the *value offering*, or the *value interface*. The same principle applies to REA when their concepts are used to initialize a UMM model. As we can see at the middle layer, REA builds an intersection of the concepts initialized by e^{3}value and additional REA concepts that are significant for the mapping to UMM. Examples of the latter one are REA's *economic commitment* or *economic exchange*. REA concepts that can not be mapped or do not have a necessity for a transformation are the *economic agreement*, the *location type*, the *business location*, the *economic exchange phase*, the *economic claim* as well as the *economic resource type*, the *economic event type*, and the *economic agent type*. Regarding the business process model developed by means of UMM, we only propose a mapping to the business requirements view. Thereby, we are able to initialize almost each concept (stereotype) within this view. Concepts that cannot be addressed within the business requirements view are the *business process activity model* and the *business entity life cycle*. Finally, the dashed arrows denote, that sometimes one concept is mapped to exactly one other concept and sometimes mapped to two concepts. The exact mappings between such concepts are discussed in the next section by describing the conceptual mapping rules.

REA improvements do not directly influence the conceptual mapping rules

As one can see in Figure 6.1, REA serves as the intermediate method to develop the process perspective from the network/value perspective. Note, the notational improvements of REA's complexity problem for multi-party collaborations does not influence the conceptual mapping between these layers. If the modeler wants to demonstrate the REA constellation in a "big picture" (similar to e^{3}value), we recommend using our proposed REA improvements - e.g., the n-ary notation introduced in Section 5.2.5, or the OCL solution presented in Section 5.2.5. Since we explicitly propose a mapping of *concepts* and not on notational constraints, it is also possible to use our mapping rules and break down each REA constellation that consists of multiple business partners or economic exchanges into binary collaborations. However, in order to keep the example simple, we only use binary collaborations in the following to explain the conceptual mapping rules between the different perspectives.

6.2 Conceptual mapping from e^{3}value to REA

In this section we describe the conceptual mapping from e^{3}value artifacts to REA artifacts. The description of our mapping is twofold: first, we list the mapped concepts of both perspective from a meta-level perspective to provide a compact overview about the source (e^{3}value) and target (REA) artifacts. Second, we apply these rules to the *customer acquisition by call center* scenario and discuss the backgrounds of the mapping rules.

Some rules can be applied automatically and some of them semi-automatically

As listed in Table 6.1 we identified five rules. All of them can be applied to a mapping from e^{3}value to REA. Each rule is indicated by a number (first column), which is also important for further descriptions and for demonstrating the formalization of the model transformation. The second column defines the e^{3}value artifacts that are mapped to REA concepts, which are presented in the third column. The last column indicates whether the rule can be applied fully automated (•) or whether the transformation needs additional semantic information (-). The additional domain knowledge should be provided by the business analyst and the domain expert. In case of a non-automated transformation it is not possible to do a straight forward mapping between the source and the target artifact since both concepts share similar but not equal semantics. For instance, rule R2 can not always be applied automatically. Since e^{3}value deals with the exchange of values between business partners, the exchanged objects are not always valuable in terms of money as required by REA (e.g. customer loyalty, popularity, etc.).

Rule	e^{3}value concept	REA concept	Auto
R1	Actor	EconAgent	•
R2	Value Object	EconResource	-
R3	Valu Transfer	EconEvent	•
R4	Value Port	from-/toParticipate	•
R5	Value Transaction	duality (EconExchange)	•

Table 6.1 *Mapping table for e^{3}value to REA artifacts*

In the following we will transform the e^{3}value notation into an REA-stereotyped UML class diagram. The top of Figure 6.2 is a cutout of Figure 5.3 and shows the value exchange between the newspaper publisher and the call center. The figure is separated into two parts indicated by a dashed line to distinguish between the e^{3}value view and the REA view. The upper part shows the e^{3}value web and the lower part the corresponding REA class diagram. Based on the following rules we demonstrate translating concepts used in e^{3}value to those used in REA. Below we list the necessary conceptual transformation rules. The numbers beside the description follow the annotation in Figure 6.2.

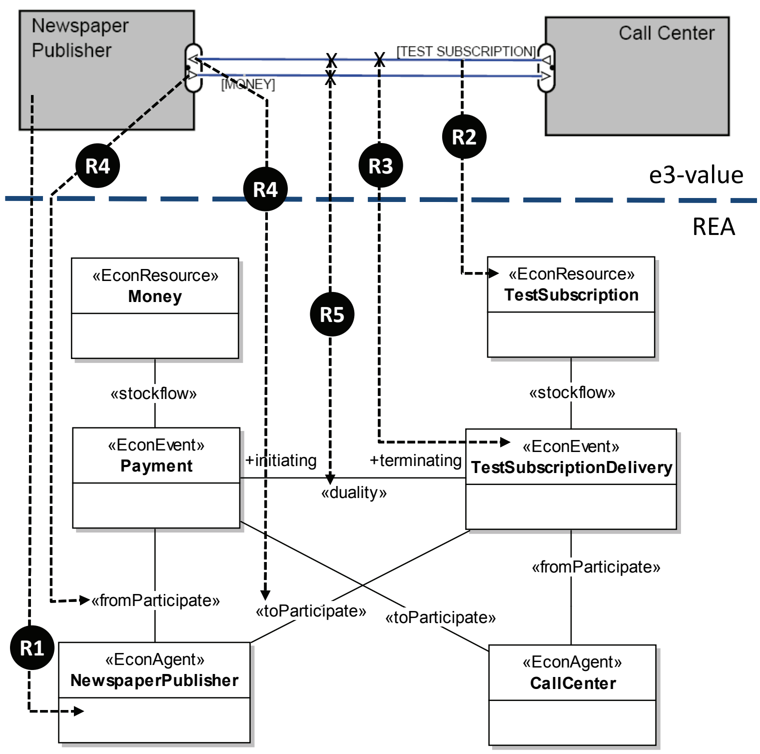


Figure 6.2 *Mapping from e^{3}value to REA*

R1 : An actor is an economic agent and is, therefore, mapped to an *EconAgent* in the REA class diagram. The mapping is straight forward since both concepts share the same semantical background: An actor in e^{3}value and an economic agent in REA are both independent participants in a business transaction with the intention to gain profit.

R2 : A value object in e^{3}value is similar to an economic resource in REA which is stereotyped as *EconResource*. In most cases the mapping between value objects and economic resources can be done easily. However, some conceptual differences have to be admitted at this point. Since REA has an accounting background, an economic resource has to be measurable in some way. In contrast, a value object does not always need to be measurable. Value objects such as *coverage* in Figure 5.3 or *customer loyalty* are of important interest when modeling an e^{3}value model in order to indicate *why* an actor participates in a value transaction from an economic perspective. However, from a REA point of view this kind of value object is not measurable in terms of money and therefore can not be mapped directly to the concept of *economic resources*.

R3 : A value transfer in e^{3}value is mapped to REA's economic event stereotyped as *EconEvent*. A value transfer in e^{3}value is depicted as the line between an in-going and an out-going value port. The e^{3}value toolset allows to name such value transfers by specifying properties, but they are not visible in the model itself - e.g. *TestSubscriptionDelivery*. Since a value transfer and an economic event are used to transfer resources between participating business partners, we map a value transfer to an *economic event*.

R4 : Value transfers in e3-value are always modeled between exactly two e^{3}value actors. This principle corresponds to the original REA notation of a ternary control relationship with two economic agents and an economic event [118]. In our proposed mapping, this relationship is decomposed into its ontological form with binary participate (from, to) relationships shown as *fromParticipate* and *toParticipate* associations.

R5 : A value transaction in e^{3}value is a bundled set of value transfers. In other words, a value interface bundles value ports that are responsible for a value transfer. In REA, an economic exchange contains a set of REA constellations, whereby their economic events are interlinked by *duality* relationships. Therefore, we map a value transaction to an REA duality association. A value transaction always follows the rule that if a value transaction takes place, all grouped value transfers happen, or none of them. The same principle is valid for economic events which are connected by a *duality* association. The semantic behind REA's duality concept is, that if one event will be executed, all of its corresponding events will be executed. e^{3}value interfaces (denoting a value transaction) share the same semantic.

6.3 Conceptual mapping from REA to UMM

In this section we detail the conceptual mapping rules for a transformation from a REA model to a UMM model. The approach is analogous to the previous mapping: first, we provide a mapping table which lists the mapped artifacts, and second, we demonstrate the mapping rules by means of real models on the instance level.

As listed in Table 6.2 we identify seven rules. The numbers of the rules follow a continuous numeration from Table 6.1. Again, the last column indicates whether the mapping rules can be applied automatically or require additional domain knowledge. For instance, this semi-automatic mapping applies to rule R8, R11, and R12. The backgrounds of the mappings are detailed further on by a description of each rule. In general, rather "dynamical" REA concepts, such as events and commitments are mapped to process related UMM artifacts and rather "statical" REA concepts, such as agents and resources are mapped to entities that are involved in business processes.

Rule	REA	UMM Stereotype	Auto
R6	EconExchange	≪bProcessUC≫	•
R7	EconAgent	≪bPartner≫	•
R8	EconRessource	≪bEntity≫	-
R9	EconEvent	≪bProcessUC≫	•
R10	EconCommitment	≪bProcess≫	•
R11	Duality(initiating)	≪bProcess≫	-
R12	Stockflow	≪bESharedState≫	-

Table 6.2 *Mapping table for REA to UMM artifacts*

In the following we describe the mapping rules in detail. Figure 6.3 is separated into two parts indicated by a dashed line to distinguish between REA (upper part) and UMM (lower part). Based on the mapping rules listed in Table 6.2 we demonstrate how to translate REA concepts to UMM concepts that are used in the business requirements view. The identifiers beside the description follows the annotation in Figure 6.3 and corresponds to the rule numbers in Table 6.2.

R6 : An *economic exchange* in REA is mapped to UMM's *business process use case*. In UMM, a business process use case is a set of related activities that together create value for a business partner. Both concepts contain the business transactions, data exchanges, events, and business partners that are necessary to fulfill the business goal.

R7 : An *economic agent* is similar to a *business partner* and is, therefore, mapped to the stereotype *bPartner* in UMM. The mapping is straight forward since both concepts share the same semantics. An economic agent in REA and a business partner in UMM are both independent participants with the intention to join a business collaboration.

R8 : *Economic resources* in REA are the actual subjects of exchange. Business entities in UMM have a similar meaning. Both concepts are used to transfer something of interest between participating business partners. However, there is a significant distinction between both stereotypes. In REA, an economic resource is a good, service, or right with economic value that is being exchanged between business partners. In UMM, such business entities are used to model different states in order to synchronize the interfaces between the business partners (e.g. the *AddressDataSet* is in state *validated*, thus it is ready for an upload). In Figure 6.3 the business entity (denoted by R3 in the *bEntityV*) is a UML class. Its state changes are further on modeled as a state machine diagram. Those states are then used as objects in the business process (see R12). Furthermore, a straight forward mapping between both concepts is often not possible, due to the very generic nature of the resource concept in REA. To map a resource to a corresponding business entity we need a detailed domain knowledge. For example, the economic resource *money* can not be mapped 1:1 to a business entity *money*. Since it is rather likely that a business entity *money* does not change its states during a business process, we propose to use the term *payment*, which is commonly used for electronic data interchange.

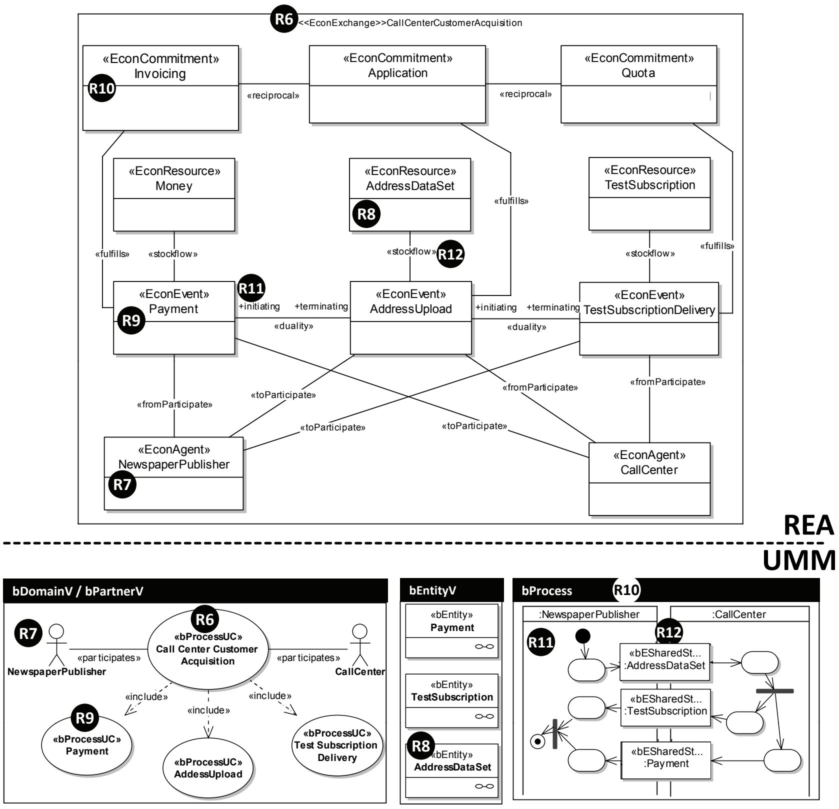


Figure 6.3
Mapping from REA to UMM

R9 : An *economic event*, similar to an *economic exchange*, is mapped to the concept of a *business process use case*. Since an economic event is always nested within an economic exchange the mapped business process use case has an *include* relationship to the *business process use case* created by rule R6.

R10 : An *economic commitment* details an *economic event*. Therefore, we propose to map an *economic commitment* to the concept of a *business process*. A business process in UMM is modeled as a UML activity diagram. An economic commitment comprises the agreements made between business partners involved in an inter-organizational business process. Those agreements are reflected in UMM's business processes as well.

R11 : The initiating role of the duality association denotes that the event holding this role initiates an economic exchange. It follows that the involved *economic agent* associated via a *from/toParticipate* association plays the initiating role in the business process. Note, it does not necessarily mean, that the terminating role is hold by the opposite business partner. In our use case scenario the initiator is the newspaper publisher since he starts the business process with an upload of the address data set.

R12 : This rule is related to R8. The *stockflow* association denotes the flow of resources triggered by an economic event. The business entity has already been created by R8. Although we do not know anything about the underlying business entity life cycle, we can already include the objects that are classified by the business entities into the corresponding business process. In UMM, these objects are stereotyped as *shared business entity states*. As soon as the modeler has specified the life cycle state of the business entity (e.g. first, the address data set is in state *validated*, then in state *uploaded*, then *processed*, etc.) he can assign these states manually to the generated *bESharedStates*.

6.4 Formalization of the transformation

Conceptual rules for a transformation provide an appropriate guideline for modelers on how to transform an input model to an output model manually. However, to be more constraint on the transformation and to support a semi-automatic mapping, a Model Driven Engineering (MDE) approach is needed. By means of this approach business analysts and software engineers are supported in performing this semi-automatic task by model-to-model transformation tools and techniques [81].

Formalization of the conceptual mapping rules by ATL

In this section we formalize the conceptual mapping rules described in the previous sections by using the Atlas Transformation Language (ATL)[9]. ATL provides a model-to-model transformation engine which is able to transform any given source model to a specific target model. In our case, we perform a transformation from the e^{3}value methodology to the REA ontology and further on we detail the rules which are necessary for a transformation from a REA model to a UMM compliant business process model. Before the engine can perform such a transformation, a proper ATL program specifies the transformation rules that are necessary for a mapping. Thereby, ATL provides a special syntax for defining those rules. The syntax of ATL will not be detailed within this thesis, since its core concepts originate from basic procedural programming languages and are easily interpretable. However, we give a short excursus on the main features of ATL. For a complete description of the abstract syntax of ATL and its execution semantics we refer to the ATL user manual [8]. In ATL, the basic construct to express the transformation logic are called transformation rules. Furthermore, ATL distinguishes between *declarative rules* and *imperative rules*. Declarative rules are also called *matched rules*. They are composed of a source pattern and a target pattern. An imperative rule has a more complex structure since it provides sophisticated features, such as native operation calls, or the call of other rules. For our business modeling approach, we only use matched rules. In order to demonstrate the transformation by these matched rules we use the Eclipse Modeling Framework (EMF) [182].

Transformation rules for a mapping from e^{3}value to REA

First, we start with the transformation rules R1 - R5, which deal with the mapping from an e^{3}value model to an REA model. Figure 6.4 shows the model-to-model transformation pattern for our customer acquisition example. At the top we define two EMF meta models (Ecore) on the MOF M2 layer [134] - the e^{3}value meta model (A) and the REA meta model (B). The first one serves as the input for the transformation and the second one for the output. In order to understand the principles of the different MOF layers, we shortly explain the *Meta-Object Facility* architecture. MOF is a four layered approach that helps to understand the metamodeling architecture of different modeling languages. It originates from UML and is standardized by the Object Management Group. The top layer M3 covers the meta-meta model which is used to develop a meta model at the M2 layer. Models that follow the constraints and definitions of such M2 meta models are called M1 models. In terms of the ATL pattern depicted in Figure 6.4, we specify the e^{3}value source model (D) at the M1 level and use it to initialize the REA target model (E) at the M1 level. The fourth layer of the MOF architecture is the M0 layer which is used to describe real-world objects. This layer is not used for our mapping and is therefore not further explained. In order to map concepts from the source model to the target model, the transformation definition (C) specifies the mapping rules in ATL and refers to both meta models. For performing the transformation, the transformation engine reads the e^{3}value source model and generates the stub for an REA compliant model. A supporting standard of MOF is XMI (XML Metamodel Interchange) [141], which defines an XML-based exchange format for models on the M3-, M2-, or M1-Layer. This means, that each model must be specified by XMI to be processable by the transformation engine. In our transformation pattern, the *e3Value2REA.atl* file executed by the transformation engine captures the transformation rules that are responsible for transforming the XMI code of a source model to a target model.

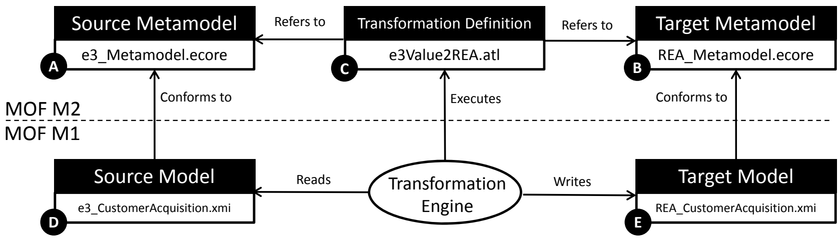


Figure 6.4
The transformation pattern for mapping an e^{3}value model to an REA model

The meta models need to be specified to perform the transformation

Before we can develop and execute the transformation rules we have to specify the meta models of the input model (A) and the output model (B). Both, the e^{3}value meta model and the REA meta model are defined in the Eclipse Modeling Framework as *Ecore* diagrams, which are basically the graphical XMI representation within the Eclipse tool. Figure 6.5 depicts a screenshot of the e^{3}value meta model integrated into the tool environment and ready for the model transformation. It defines the modeling elements of the e^{3}value methodology and the relationships between each other. For instance, the *actor* element contains one to many *value interfaces*. Each value interface must contain exactly two value offerings. In e^{3}value, a value offering is defined as a set of equally directed *value ports*. Thus, a value offering has one to many *in-going value ports* and one to many *out-going value ports*. Those value ports are connected to a value transfer, which is part of a value transaction. Finally, a *value object* is *transferred* via a value transfer. This information is necessary to perform a model-to-model transformation. The meta model of the REA ontology is modeled as well in the Eclipse Modeling Framework but not shown in this section, since it was already discussed in Chapter 5 by means of Figure 5.6.

Once the meta models for the input and output methodology have been specified, the transformation definition must be developed (C in Figure 6.4). Listing 6.1 presents the transformation rules which are necessary to perform the transformation from an e^{3}value model to an REA model. In general, transformation definitions in ATL start with a header section, in which the parameters for the transformations are defined. The most important parameters are the definition of the source and the target model (line 82). The keyword *OUT* indicates the target model and the keyword *IN* refers to the source model. The remainder of this source code represents the mapping rules for the actual transformation of the modeling artifacts.

The transformation of an e^{3}value actor to an economic agent

The first transformation rule *R1* covers the mapping from an e^{3}value actor to an economic agent of REA. In comparison to the other transformation rules of the provided source code it represents a rather simple rule, which is shown between line 85 and line 93. The name of the rule is *R1_Actor2EconAgent* and is provided after the keyword *rule*. The source pattern of the rule specifies a variable of

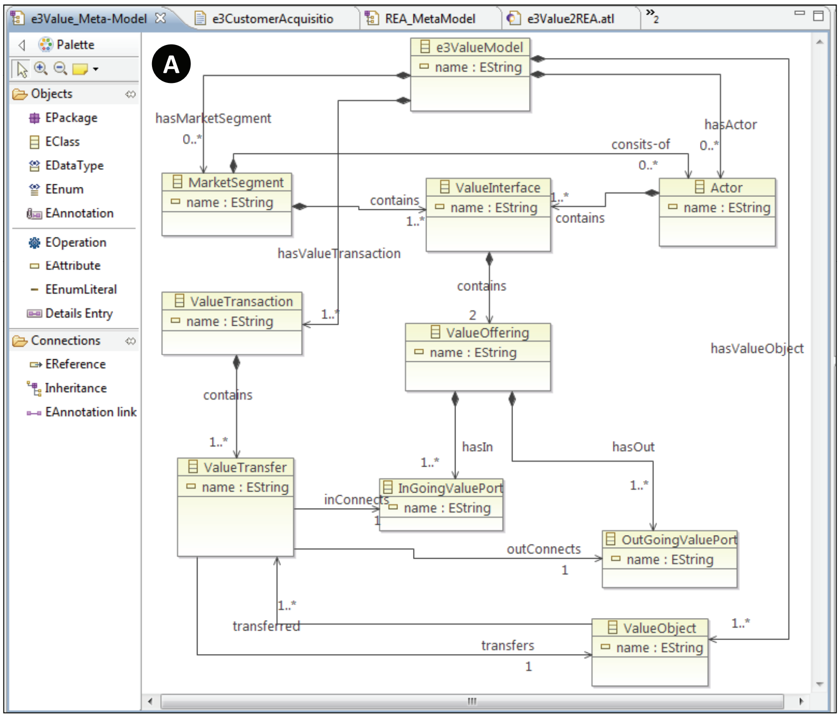


Figure 6.5
The e^{3}value meta model developed within the Eclipse Modeling Framework

type *Actor* which is part of the e^{3}value meta model (line 87). In line 89, the target pattern defines one element of type *EconAgent* which is part of the REA meta model. This element specifies two bindings (line 90 and line 91). The first one is used to express the initialization of the *name* attribute and the second one is used to define the *stereotype*. In general, the symbol '<-' is used to express the feature to be initialized (left-hand side) and the feature that represents the initialization expression (right-hand-side).

The transformation rules R2 and R3

The transformation rule *R2* in line 96 is used to express the mapping from a value object to an economic resource of an REA model. The rule is similar to R1, since the source and target pattern share similar semantics. The only distinction is represented by line 103. It initializes the *stockflow* association, which connects in REA an economic resource and an economic event. From the e^{3}value meta model we know, that a *transferred* relation connects the value object and the value transfer. Since a value transfer is transformed to an economic event by rule *R3* in line 108, we can directly initialize the *stockflow* association by this connection in line 103. Note, as already demonstrated in Table 6.1, the transformation rule *R2* can not always be performed fully automatic. If the e^{3}value model contains value objects in their value transfers that are not needed for a further refinement by REA, it is not recommended to apply this transformation rule. From a practical point of view, this means that, after the transformation, the modeler needs to assess whether a generated economic resource either remains in the target model, or needs to be deleted, or must be renamed to comply with the semantics of the target model. In [117], we discuss such value objects, that are necessary in the value perspective but have no benefit within REA's trading partner perspective.

Listing 6.1
ATL Transformation rules for a mapping from e^3value to REA

```
78 -- @path e3=/ATL_REA2UMM/metamodels/e3Value_Meta-Model.ecore
79 -- @path REA=/ATL_REA2UMM/metamodels/REA_MetaModel.ecore
80
81 module e3Value2REA;
82 create OUT : REA from IN : e3;
83
84 --Transformation of an e3-Value Actor to an REA Economic Agent
85 rule R1_Actor2EconAgent{
86         from
87                 e : e3!Actor
88         to
89                 r : REA!EconAgent (
90                         name <- e.name,
91                         stereotype <- 'EconAgent'
92                 )
93 }
94
95 --Transformation of an e3-Value Object to an REA Economic Resource
96 rule R2_ValueObject2EconResource{
97         from
98                 e : e3!ValueObject
99         to
100                r : REA!EconResource (
101                        name <- e.name,
102                        stereotype <- 'EconResource',
103                        stockflow <- e.transferred
104                )
105 }
106
107 --Transformation of an e3-Value Transfer to an REA Economic Event
108 rule R3_ValueTransfer2EconEvent{
109        from
110                e : e3!ValueTransfer
111        to
112                r : REA!EconEvent (
113                        name <- e.name,
114                        stereotype <- 'EconEvent',
115                        stockflow <- e.transfers
116                )
117 }
118
119 --Transformation of an e3-Value Transaction to an REA duality relationship
120 --Transformation of an e3-Value In-going Port to a toParticipate relationship
121 --Transformation of an e3-Value Out-going Port to a fromParticipate relationship
122 rule R4_R5_AssociationMapping{
123        from
124                vtrans : e3!ValueTransaction,
125                vt1 : e3!ValueTransfer,
126                vt2 : e3!ValueTransfer (vtrans.contains.includes(vt1) and vtrans.contains.includes(vt2))
127        to
128                event1 : REA!EconEvent(
129                        duality <- event2,
130                        toParticipate <- e3!Actor.allInstances() -> select(a|a.contains.contains.hasIn.
                                includes(vt1.inConnects)).first(),
131                        fromParticipate <- e3!Actor.allInstances() -> select(a|a.contains.contains.hasOut
                                .includes(vt1.outConnects)).first()
132                ),
133                event2: REA!EconEvent(
134                        duality <- event1,
135                        toParticipate <- e3!Actor.allInstances() -> select(a|a.contains.contains.hasIn.
                                includes(vt2.inConnects)).first(),
136                        fromParticipate <- e3!Actor.allInstances() -> select(a|a.contains.contains.hasOut
                                .includes(vt2.outConnects)).first()
137                )
138 }
139 [...]
```

Transformation rule for mapping the value port to the from-/toParticipate association

In summary, *R1* to *R3* initializes the core stereotypes of the REA model - the economic resource, the economic event and the economic agent. These rules deal with rather straight-forward mappings and are easily understandable, since they are implementing a mapping between classes. Following the conceptual transformation Table 6.1, *R4* and *R5* describe the transformation of e^3value concepts to associations within REA. Whereas in *R4* a value port is mapped either to a *from-* or *toParticipate* association according to its direction, a value transaction is mapped to a *duality* association in *R5*. Both actions can be performed in one transformation rule. Thus, a more complex transformation rule is expressed by *R3_R4* in line 122. The input pattern of this rule specifies a variable which is of type value transaction in line 124. This variable is further on used to express the properties of the value transaction that should be mapped to a duality association. In line 125 and 126, two variables of type value transfer are defined. The second one is constrained by an OCL statement. It specifies, that only those value transfers are considered in the transformation that are connected to a value transaction. Since REA specifies multi-party business collaborations by breaking them down to multiple bilateral transactions (having exactly two value transfers) that are connected via duality associations, we can also specify exactly two value transfers in the declarative part of the transformation rule. Although the e^3value meta model allows one to many value transfers within one value transaction, the rule fires for each paired value transfer as given by the OCL constraint in line 126. The target pattern as specified between line 128 and 136 is responsible for the initialization of the output model. Line 129 and 134 denote, that the two economic events that resulted from the transformation of a value transfer are connected via a duality association as specified in the REA meta model.

Initializing REA's duality association

An even more complex statement is given in line 130 dealing with the initialization of the *toParticipate* associations. In REA, each economic actor is connected with an economic event via this association denoting the direction of the *economic stockflow* and the commitment of participating in the economic exchange. At the right hand side of this statement we use an OCL constraint to navigate through the input model. The task is to get all in-going value port instances of an e^3value actor that are connected to the value transfer specified in the declarative part by the variable *vt1*. Once the corresponding ingoing vale port is retrieved via this OCL constraint, a *toParticipate* association (see left-hand side) between the corresponding economic actor and the economic event is initialized. According to the generation of the *toParticipate* association, the generation of the *fromParticipate* association is defined similarly in line 131. Finally, the associations assigned to the reciprocal economic event are specified between line 133 and 136. This statement is side-inverted to the statement between line 128 and 131 as detailed in this paragraph and, hence, does not require further explanations.

Transformation rules for a mapping from REA to UMM

The aim of this sub-section is to present the matched rules necessary for generating UMM artifacts out of a business model based on REA. Again, the transformation does not automatically initialize an entire UMM compliant model, but it generates a stub for UMM's business requirements view. The ATL transformation pattern for this mapping is depicted in Figure 6.6. An REA meta model (A) serves as the meta model for the REA source model (D) and a UMM meta model (B) serves as the meta model for the UMM target model (C). The transformation engine executes the model-to-model transformation definition (C) in order to perform the mapping.

The target meta model needs to be specified to perform the transformation

In the following, we present the meta model of the target model. Due to space limitations we only present a cutout of the UMM meta model developed by means of the Eclipse Modeling Framework in Figure 6.7. However, since our proposed business modeling approach performs a mapping from REA artifacts to UMM artifacts delivered by the business requirements view, we only show the most impor-

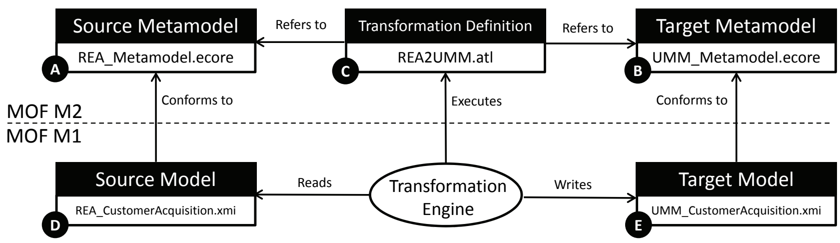


Figure 6.6 The transformation pattern for mapping an REA model to a UMM model

tant parts that are involved in this transformation. The interested reader is referred to the UMM 2.0 specification to learn more about the meta model of UMM [193]. Note, the meta model as depicted in Figure 6.7 differs slightly from the one in the standard specification. This results from restrictions made by the Eclipse Modeling Framework in regard to the naming rules of the relationships between the stereotypes. Hence, we tailored the UMM meta model for our model transformation purposes. Nevertheless, it reflects the same semantics as defined in the specification.

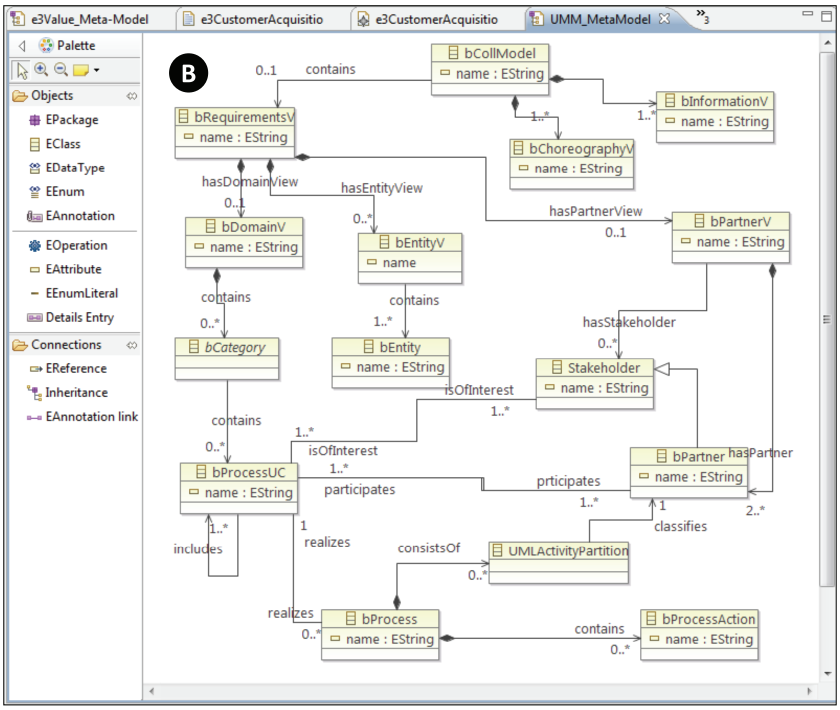


Figure 6.7 The UMM meta model developed within the Eclipse Modeling Framework

The transformation rules for the mapping between REA and UMM are presented in Listing 6.2. According to the transformation overview of Table 6.2, not all the transformation rules are foreseen to be performed automatically (e.g. *R11*, *R12*). Hence, we only detail the transformation rules *R6* to *R10* by discussing the essentials of these code expressions.

The transformation rules R6 and R7

Similar to the first part of the previous transformation, where we transformed e^{3}value to the main REA concepts, the matched rules *R6* to *R8* are straight forward, too. Again, we only transform concepts that are independent from other modeling elements and therefore do not require a complex code structure. For instance, the transformation rule *R6* in line 147 is responsible for the mapping of an economic exchange to a business process use case. REA does neither contain any information about the location of this business process use case in the UMM model, nor any information about its relationships in the target model. Thus, we are only able to map the plain concepts between the two methodologies. Slightly different from this transformation is the transformation rule *R7* in line 158. This rule performs the transformation from an economic agent to UMM's business partner. By performing this rule, we can already make conclusions about the relationships of the target element. In REA, the economic agent is connected via *from-* or *toParticipate* to an economic event. Line 165 and 166 use this information to initialize the *participates* associations and to attach the involved business partners to the business process use case.

The transformation rule R8 and R9

The transformation rule *R8* in line 171 is similar to *R6* and deals with the mapping between an economic resource and a business entity. A more complex statement is presented by rule *R9* in line 182. This rule is responsible for the mapping of an economic event to an included business process use case. In order to get the necessary information to initialize all *includes* relationships between the business process use cases, we need to navigate through the instances of an economic exchange. The task is to find the economic events attached to the economic exchange, since their relationships are mapped to the *includes* relations. This is done by the OCL statement in line 189.

Transforming an economic commitment to a business process

Finally, mapping rule *R10* in line 194 deals with the transformation of an economic commitment to a business process. The task of this transformation rule is to initialize the business process by the name of the commitment and to generate the UML activity partitions. This first initializing step is fulfilled in line 199. The second step is specified in line 202 and 205 and requires an OCL statement to perform the transformation. UMM's business processes comprise UML activity partition denoting the participating business partners. These partitions are assigned to a certain business partner by classifying the instance of a business partner object. Thus, we need to calculate which business partner is connected to the transformed economic commitment. The OCL statement is used to navigate through the source model.

Listing 6.2 *ATL Transformation rules for a mapping from REA to UMM*

```
140 -- @path REA=/ATL_REA2UMM/metamodels/REA_MetaModel.ecore
141 -- @path UMM=/ATL_REA2UMM/metamodels/UMM_MetaModel.ecore
142
143 module REA2UMMTransformation;
144 create OUT : UMM from IN : REA;
145
146 --Transformation of an economic exchange to a business process use case
147 rule R6_EconExchange2bProcessUC{
148         from
149                 r : REA!EconExchange
150         to
151                 u : UMM!bProcessUC(
152                         name <- r.name,
153                         stereotype <- 'bProcessUC'
154                 )
155 }
156
157 --Transformation of an economic agent to a business partner
158 rule R7_EconAgent2BusinessPartner{
159         from
160                 r : REA!EconAgent
161         to
```

```
		u : UMM!bPartner (
			name <- r.name,
			stereotype <- 'bPartner',
			participates <- r.fromParticipate,
			participates <- r.toParticipate
		)
}

--Transformation of an economic resource to a business entity
rule R8_EconResource2BusinessEntity{
	from
		r : REA!EconResource
	to
		u : UMM!bEntity (
			name <- r.name,
			stereotype <- 'bEntity'
		)
}

--Transformation of an economic event to an included business process use case
rule R9_EconEvent2IncludedBPUC{
	from
		event : REA!EconEvent
	to
		u : UMM!bProcessUC (
			name <- event.name,
			stereotype <- 'bProcessUC',
			includes <- REA!EconExchange.allInstances() -> select(a|a.contains.includes(event
				)).flatten()
		)
}

--Transformation of an economic commitment to a business process
rule R10_EconCommitment2bProcess{
	from
		commitment : REA!EconCommitment
	to
		u : UMM!bProcess (
			name <- commitment.name
		),
		part1 : UMM!UMLActivityPartition (
			classifies <- REA!EconAgent.allInstances() -> select(a|a.fromParticipate.fulfills
				.includes(commitment)).first()
		),
		part2 : UMM!UMLActivityPartition (
			classifies <- REA!EconAgent.allInstances() -> select(a|a.toParticipate.fulfills.
				includes(commitment)).first()
		)
}
[...]
```

Executing the transformation rules

Once the transformation rules have been defined in ATL, we can perform the transformation by executing the ATL code. This task is done by the ATL Execution engine, which is integrated in the ATL IDE on top of Eclipse. It includes an *ATL Compiler* to transform ATL programs into byte-code and an *ATL Virtual Machine* to execute the byte-code generated by the compiler. Furthermore, a *Model Repository* stores the meta models as well as the models as XML files serialized according to the XMI standard [141].

Steps for transforming an e^{3}value model to an REA model and further on to a UMM model

Before we demonstrate the transformation by executing the transformation rules of Listing 6.1, we provide an overall step-by-step guide on how to apply these formalized rules to the specific models on each layer. Figure 6.8 shows the workflow on how to get from an e^{3}value model to an REA model and further on to a UMM model. In step 1 the modeler designs the e^{3}value model to depict the value network perspective. In order to use the benefits of the formalized transformation rules the e^{3}value model must be exported to XMI (transformation A). However, the e^{3}value toolset [47] is not applicable for an automatic export to XMI. Thus, the modeler has two possibilities. Either, he uses the graphical tree-view feature of the Eclipse Modeling Framework that is able to provide a user-friendly XMI interface (c.f. Figure 6.9), or he uses a UML tool that is able to export to XMI. If the modeler decides to follow the latter one he needs to be aware of the XMI interoperability issues between the dif-

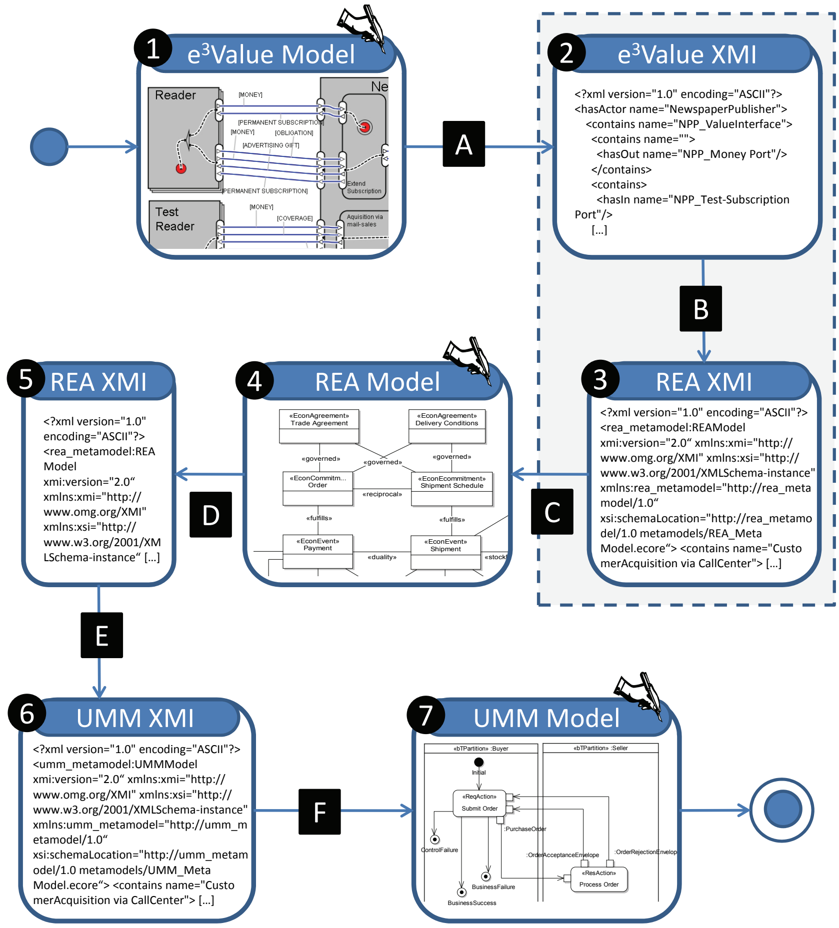


Figure 6.8 *The workflow for getting from an e³value model to a UMM model by using the transformation rules*

ferent tool vendors. Once the e³value model is represented by XMI it serves as the source model for the first transformation (B). Thereby, the transformation rules of Listing 6.1 initialize the target model presented in step 3. The generated REA model provides a skeleton for further refinements that can be made by the use of any UML modeling tool that has an XMI import functionality (step 4). Since REA is based on UML the export of the enhanced REA model to XMI can be performed directly by the UML tool (transformation D). Similar to transformation B, the XMI code of the REA model (step 5) is used as a source model that is transformed (E) to a UMM target model (6) by executing the transformation rules of Listing 6.2. As a final step, the generated XMI code must be imported (F) to the UML tool to make further refinements on the UMM model manually (7).

Source and target model are represented as XMI code

As a proof-of-concept and to give an overview on how to define a source model by using the XMI standard and the Eclipse Modeling Framework, we demonstrate the transformation by executing the transformation rules of Listing 6.1. The code shows the transformation of an e³value model to an REA model. In the step-by-step guide of Figure 6.8 this transformation is highlighted by a dashed rectangle (step 2 and step 3). We use a simple e³value example from the customer acquisition use case to show how the execution is performed.

Figure 6.9 demonstrates this transformation by means of a serialized e^{3}value model (A) and a serialized REA model (B).

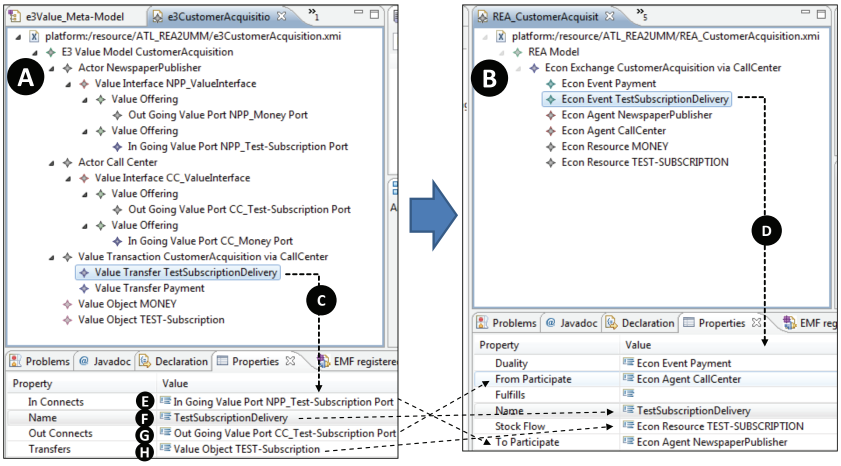


Figure 6.9
Source and target model specified by XMI

The Eclipse IDE allows an enhanced graphical representation of the XMI code by a tree-view

According to the step-by-step guide of Figure 6.8 and in order to perform the transformation, both models are no longer represented by a graphical notation. Instead, they are defined by a serialized XMI construct which has been prepared to a tree-view by the IDE in order to improve the readability of both models. Having specified the source meta model (see Figure 6.5) the Eclipse IDE allows for a semi-automatic generation of a meta model compliant stub. We use this feature to generate the input model as shown on the upper left-hand side of Figure 6.9. The elements shown in the tree-view represent the simplified e^{3}value model on top of Figure 6.2. Each element is further specified by properties as defined in the meta model. For instance, the value transfer element *TestSubscriptionDelivery* highlighted by (A) is specified by different parameters according to its attributes and relationships within the meta model (C). All these properties are used to perform the transformation and to generate the target model (B). As we learned from the conceptual mapping between an e^{3}value model and an REA model, a value transfer is transformed to an economic event, which is highlighted in (B). Analogous, the properties refined by (D) represent the attributes and relationships to other modeling elements. Having a deeper look at the properties refined by (C), we can see the one-to-one transformation of the *value transfer* as defined by the matched rule *R3*, *R4*, and *R5* of Listing 6.1. The in-going value port of the e^{3}value model, which is attached to the newspaper publisher via the *inConnects* association is mapped to the *fromParticipate* association in REA (E). Accordingly, the out-going value port maps to the *toParticipate* association as depicted by (G). The transformation of the name of the main concept (value transfer to economic event) is highlighted by the arrow annotated by (F). Finally, the value object is mapped to an economic resource, which is connected with the economic event via a *stockflow* association (H).

The generated REA target model is now ready for an import to any graphical modeling tool which provides XMI interoperability. Since e^{3}value does not capture all the information necessary in the trading partner perspective, it is now the modeler's task to add the missing artifacts manually which were not initialized by the transformation. Once this task is fulfilled, the REA model must be exported to XMI as well to serve as the input model for generating the skeleton of UMM's business requirements view. However, the execution of this transformation is not presented anymore, since it follows the same approach as depicted in Figure 6.9.

6.5 Final assessment

In this chapter we demonstrated the business modeling based approach by describing the conceptual mapping between three major modeling methodologies in the filed of B2B - e^3value, REA, and UMM. We motivated this approach by the need of business models to elicit the economic requirements before developing the business processes. e^3value depicts the business partner network and evaluates whether the model is economic sustainable or not. REA follows a similar approach, but strongly focuses on observing the give-and-take principles of the B2B system to be developed. Both methodologies, which are located on the business modeling perspective, complement each other, whereby especially REA takes advantage of the information gathered by the e^3value model. On the business process perspective, UMM needs to gather business domain knowledge for its business requirements view in an early design phase. REA delivers the requirements from an economic point of view and provides the input for modeling a UMM compliant business process model. In other words, our mapping rules should help the modeler setting up a UMM compliant model without disregarding the give-and-take principles by REA and to ensure economic sustainability by e^3value. We demonstrated the mapping i) by conceptual mapping rules specified in natural language, which should guide the modeler in interlinking the different models and ii) as formalized transformation rules by means of ATL supporting a model-to-model transformation in a semi-automated manner.

It is not guaranteed to generate a complete business process model from an e^3value and REA model by using our mapping rules. However, the modeler can (semi-)automatically generate a stub of a UMM model, which needs to be finalized during further modeling steps. The lessons learned in this chapter are valuable for understanding the following chapter, which describes one of the key contributions of this thesis - the worksheet-driven approach on how to get from business models to business process models during the requirements elicitation phase by the help of structured forms.

7 Worksheet Driven Approach

In the previous chapter, we introduced the business modeling approach for developing B2B processes. This approach spans over three different methodologies - e^3value, REA, and finally UMM. It included conceptual mappings between the different methods and a formalization of the mapping rules. In order to apply this approach to the B2B model, it is crucial that business experts are able to express and evaluate value exchanges, agreements and commitments between the partners and that the software engineers get all necessary information to bind the private process interfaces to the public ones. However, business experts - who usually have a very limited understanding of conceptual modeling - prefer expressing their thoughts and evaluating the results by plain text descriptions. Thus, we need a method for describing rationales behind the inter-organizational system.

Worksheets have already been introduced in the general approach

In this chapter we describe an approach that complements and integrates the business modeling based approach of Chapter 6 by using text-based templates called worksheets. We already know the application of worksheets from the general requirements engineering approach introduced in Chapter 4. Worksheets should help the modeler to gather requirements in natural language in order to develop a B2B model [74]. The information kept in those worksheets is gathered by the business analyst through interviews with the business expert. Communication with business experts is often based on plain text descriptions and less on formal models. Today, a business analyst already uses some predefined templates in order to gather information from the business experts. Thereby, UN/CEFACT proposes a set of documents and forms for UMM. However, these documents are loosely connected to the B2B model. Accordingly, the business analyst has to connect the dots - by combining information spread all over the documents - in order to create a B2B model. Furthermore, those documents are limited to a set of UMM artifacts. They do not consider the artifacts delivered by the business modeling techniques e^3value and REA. By our worksheet driven approach we suggest a better alignment of those documents in order to facilitate the development process of a B2B model. Furthermore, we present implementation guidelines on how to integrate the worksheets into a modeling tool.

The remainder of this chapter is structured as followed: in Section 7.1 we motivate the use of a worksheet driven approach in more detail. Section 7.2 gives a short overview of the features and the rationals of the approach. Section 7.3 demonstrates an example and shows how to apply worksheets to the customer acquisition example. Furthermore, we provide a step-by-step guide, which guides the business analyst during the different modeling phases. Finally, in Section 7.4 we discuss the technical aspects of our approach as well as the implementation of a tool called worksheet-editor, which is part of the UMM Add-In [100] [63] [102]. Section 7.5 concludes the chapter with a final assessment.

7.1 The motivation for a worksheet-driven approach

Worksheets are used to capture business domain knowledge

As already said, worksheets are structured forms for the elicitation of specific requirements. The information kept in those worksheets is gathered by the business analyst through interviews with the business expert. Thus, the business analyst should not influence the business expert. The language of choice should be a non-technical one, i.e. technical and modeling terms should be avoided. The interviews help the business analyst to understand the business domain and to ensure that all involved parties share the same understanding.

Besides, worksheets are more than documenting the collaboration space. The information collected in worksheets is strongly interrelated between each other. This helps the business analyst to have a guidance during the modeling process and to identify the connections between the different modeling artifacts.

Worksheets must be aligned with the delivered models

Today, the requirements elicitation is mostly done on paper or by means of a word processor. It is obvious that maintaining the text-based worksheets independent of the information stored in a model results in a duplication of efforts and a high danger of inconsistencies. For example, the meta model of UMM foresees tagged values that are mandatory to specify in order to get a UMM compliant model. Those tagged values capture almost the same information as we provide in our worksheets. Hence, it does not make sense to store this information twice. Furthermore, if the worksheets are loosely connected to the model, it is not possible to generate artifacts out of the captured requirements.

Using the same vocabulary avoids inconsistencies

The strong alignment of worksheets by structured and pre-defined forms allows an integration into any modeling tool that supports e^{3}value, REA or UMM. It forces the modelers to use the same vocabulary, which helps to avoid inconsistencies between requirements. However, different domains require different information and thus, different worksheet templates. For this reason, we show how a specially designed XML-based worksheet definition language allows customization to special needs of certain business domains.

In our approach, we propose worksheets for all modeling steps - beginning from e^{3}value over REA to UMM. Especially UMM worksheets require specific attention during the requirements elicitation phase. The reason is twofold: First, UMM is quiet complex compared to e^{3}value and REA in regard to the delivered artifacts and the size of the model. Second, UMM requires the specification of tagged values which are important for the implementation of the B2B system further on. Since these tagged values are often represented one-to-one in UMM worksheets, it is rather important to go into more details of UMM worksheets than into worksheets of e^{3}value and REA.

As a summary of the motivation for a worksheet-driven approach we provide four features that integrate the different views and fosters the development process of a B2B model:

1. Providing a single access point
2. Flexibility in customizing worksheets
3. Automatic generation of model and code artifacts
4. Documenting the collaborative space

7.2 The worksheet-driven approach at a glance

The complementation of the business modeling based approach by using worksheets brings a set of advantages to the modeler which will be outlined in this section. However, those advantages does not always cover all of the three modeling languages to the same extent. For example, the feature of using a single access point brings more benefits to UMM compared to e^{3}value. The reason is that UMM stores a lot of information in tagged values. e^{3}value does not contain any tagged values and, hence, does not implement this feature. In contrary, the flexibility in customizing worksheets is used by all of the three modeling languages - e^{3}value, REA, and UMM.

Providing a single access point

As already mentioned, UMM is defined as a UML profile. Using stereotypes, tagged values, and OCL constraints the profile tailors the UML meta model to the specific needs of inter-organizational business process modeling. In particular relevant for the process based requirements engineering are tagged values.

One major flaw of current process requirements engineering approaches is the gap between the business process models and functional/non-functional requirements such as time to respond, pre-conditions for a business process, authorization, and non-repudiation properties etc. These requirements are often stored separately from the business process model using text documents, hence causing problems in terms of maintainability, traceability, and alignment. What is needed, is a single point of access for both, the business process models and the functional/non-functional requirements belonging to the models.

The repository approach brings advantages to the development of a UMM model

Instead of choosing a repository approach where both, the model and the requirements information are stored separately, but tied together using identifiers, we suggest to store the requirements information directly in the business process model. This provides the advantage, that requirements information is always present and accessible for vertical model transformations finally leading to executable code. In order to store the requirements information directly in the UMM model, an extension mechanism and methods for storage and retrieval of the requirements information is necessary. Since UMM is defined as a UML profile we use the concept of tagged values as the underlying storage for process requirements information.

Figure 7.1 shows three classes from the UMM meta model depicting the abstract type *BusinessAction* and the two stereotypes *requesting action* and *responding action*. These stereotypes inherit the tagged values from the super class. The stereotype *requesting action* defines additional tagged values. It follows that actions stereotyped as *requesting action* or *responding action* are not just regular UML actions but carry also well defined functional/non-functional properties for the very specific purpose within UMM.

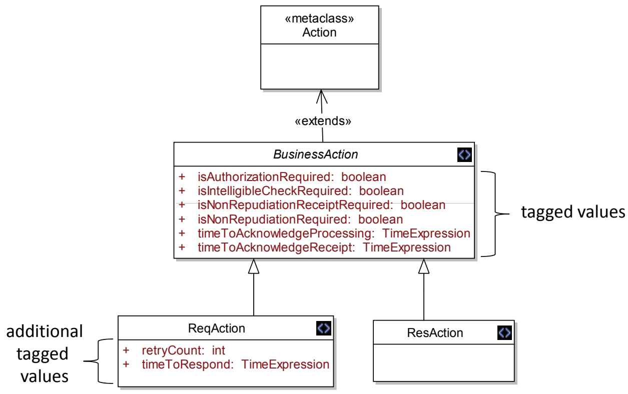


Figure 7.1 *Cutout from the UMM 1.0 meta model*

In Figure 7.2 we exemplify how the stereotypes *requesting action* and *responding action* are instantiated in a UMM example showing a business transaction. A requesting action (denoted by A) calls a responding action (D). In UMM a business transaction is the core concept of a business information exchange between two organizations. In the given example the *buyer* submits the order to the *seller* by sending a *purchase order* (C). The *seller* processes the purchase order and replies either with an *order acceptance envelope* or an *order rejection envelope* (B).

Tagged values are maintained by worksheets

At the bottom of the graphical representation of the business transaction in Figure 7.2 we present the tagged values maintained in a UML tool capturing the requirements of the requesting action (E) and the responding action (F). Amongst other things, tagged values denote whether an authorization is required prior to the execution of the transaction (isAuthorizationRequired), how often the newspaper publisher may restart the transaction in case of a control failure (retryCount) etc. The instantiation of these tagged values is driven by information gathered from the business experts during the requirements elicitation phase. It is quite common that such information is stored separately to the business process model using word processors. UN/CEFACT also follows such an approach. The UMM comes with a set of worksheets that are predefined text forms.

A prototypical implementation of an interactive worksheet editor for UMM

In order to allow for a convenient handling of the different worksheets we have developed a tool called Worksheet Editor. The Worksheet Editor is an integral part of the UMM Add-In [100] - a prototypical customization of the UML modeling tool Enterprise Architect supporting UMM. By integrating a set of worksheets into our

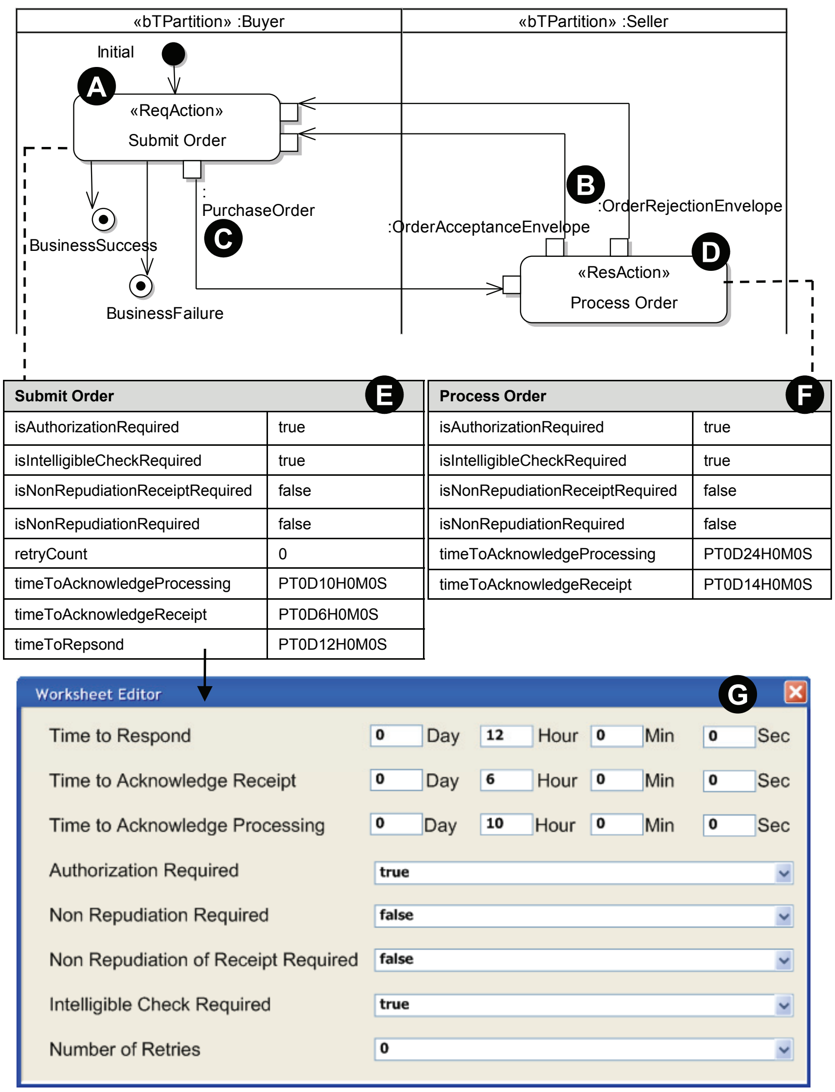


Figure 7.2
UMM model as single repository

tool we are able to maintain the functional/non-functional requirements gathered from the business experts within the business process model. Figure 7.2 shows a screenshot of the worksheet editor (G). The worksheet editor provides an easy to use interface for the business experts in order to express and check his needs during the requirements elicitation phase.

Flexibility in customizing worksheets

As outlined before, worksheets provide effective means to persist existing business domain knowledge and business process knowledge. UN/CEFACT developed its current set of worksheets according to typical questions asked by a business analyst. As a result these worksheets are not perfectly aligned to the UMM meta model or the business modeling methodologies. In our approach we focus on a better alignment with UMM's UML profile and the special needs of using them for e^{3}value and REA. It is important that all stereotype and tagged values in case of UMM and all modeling elements in case of e^{3}value and REA are reflected in the worksheets - but not each stereotype or modeling element has its own worksheet. This means that some stereotypes and their tagged values are described in the worksheet assigned to another stereotype - the master stereotype. Analogous, this concept also applies to e^{3}value and REA in terms of modeling elements.

The Worksheet Definition Language (WDL) provides a flexible layout of worksheets

Different business domains are likely to have different requirement needs. Worksheets from the chemical industry are different from worksheets from the tourism industry. In order to cope with changing requirements we developed a flexible solution - the Worksheet Definition Language (WDL). The WDL is defined as an XML schema. A concrete WDL instance is used to define the content and the layout of a worksheet. Furthermore it specifies the dependencies between worksheet elements and the tagged values for storage within the process model. The UML tool dynamically loads the WDL information and renders the Worksheet Editor's layout at runtime. Using this generic approach the WDL is not specific to UMM, e^{3}value, or REA, but may also be used with other meta models. For instance, WDL can be applied to a Domain Specific Language (DSL). First experiments with such an integration to a DSL has been made within the context the BSopt project [1]. Thereby, we integrate worksheets into a self-developed modeling tool that is based on DSLs to connect the different layers (cf. Figure 2.2). Within this tool the worksheets serve as a kind of "wizard" for the modeler to get from one layer to the other one. WDL has been used to import the different worksheets into the tool and to interlink the requirements information that is captured among the different layers.

As shown in Figure 7.3, we create a WDL instance representing an appropriate set of worksheets for e^{3}value and REA concepts, and especially for the UMM meta model. Thereby we started off from the UMM meta model. We grouped the semantically related stereotypes and their tagged values into worksheets and defined their layout using WDL. The WDL instance we created builds the basis valid for every UMM model. This basis may be extended to the specific needs of certain industries. In order to capture additional domain specific functional/non-functional requirements, the modeler may add new elements to existing worksheets or create new worksheets. Furthermore the modeler is able to bind this new information to new tagged values in the business process model. Both, the meta model specific information and the user extensions in the WDL are rendered at runtime to create the layout of the Worksheet Editor. An elaboration of the WDL is given in Section 7.4 "Technical implementation".

Automatic generation of model and code artifacts

Automatic model generation enhances productivity and reuse

The concept of worksheets does not only facilitate the requirements elicitation, but also allows for a semi-automatic creation of business model and business process artifacts. During the creation of a B2B model the modeler gradually fills the worksheets with functional and non-functional requirements. Requirements information gathered in previous steps may be used in later modeling steps to semi-automatically

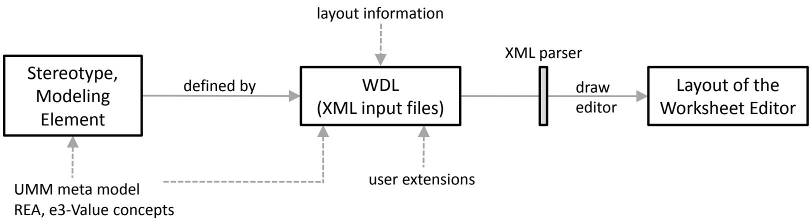


Figure 7.3 Flexible worksheet design

generate artifacts. For example, a UMM business transaction as shown on top of Figure 7.2 depends on information gathered in previous steps of the UMM. Using the Worksheet Editor the modeler selects certain parameters and the business transaction is generated automatically using the existing information. Hence, the user does not have to enter all the required information manually nor does the user have to draw the business transaction. Thus, the productivity and the level of reuse are greatly enhanced. A detailed description of the generation mechanism is given in Section 7.4.2.

UMM provides an approach for capturing the requirements of B2B processes. In order to allow for an integration into B2B systems a derivation mechanism for deployable artifacts has to be defined. We developed a vertical model transformation approach leading from business requirements to process models and eventually to executable code. However, the derivation of executable artifacts for a service oriented architecture is not part of this thesis. In [74], we show, how worksheets support the IT experts to generate deployment artifacts by example.

Documenting the collaborative space

Exporting worksheets to Microsoft Word, HTML, or WDL

A good documentation is crucial in any software development project. A report on a B2B model should serve the needs of all stakeholders - from the business experts to the software engineers - with all their different skills and needs. Thus, it is not appropriate to limit a report only to modeling artifacts. By having integrated the worksheets into the B2B model we are able to generate a documentation, suitable for the evaluation by the business expert. The worksheet editor is able to generate an appropriate documentation in Microsoft Word, HTML or WDL based on the captured business domain knowledge. The usage of the WDL file format allows re-importing the data again into the worksheet editor for recovering purposes. In general, the documentation contains a textual, table-like description of the business knowledge. Different kinds of simple modeling artifacts may be added to these descriptions on demand.

7.3 Worksheets by example

In this section we go step by step through the development process of a B2B model by using our worksheet-driven approach. The approach requires worksheets within each modeling perspective, i.e. specific worksheets for e^{3}value, REA, and UMM. We start off with a "road-map" to provide a guide for the business analyst about the sequential order of the different worksheets used during the modeling phase. In the following, we show the different worksheets for each modeling methodology. For demonstration purposes we only present the most important and significant ones.

7.3.1 A guide for the business analyst

Guiding the business analyst through the development process of a model compliant to our methodology

The reference guide for the worksheet-driven approach is coarse-grained outlined in Figure 7.4. In this figure, we distinguish the different main steps to create the B2B model. The main steps contain the different methodologies - e^{3}value (A), REA (B) and UMM (C). Within each methodology, the business analyst must create a specific set of worksheets (e.g. A.Wx). The "x" denotes that, in the following, there is a unique identifying number assigned to each worksheet. For instance, A.W1 identifies the *consumer need worksheet*. Worksheets having an ID that starts with "A" are assigned to e^{3}value, the ones starting with "B" are assigned to REA, and the worksheets starting with "C" are used for the UMM perspective. The workflow, which is rudimentarily shown within each perspective, depicts the sequential order of the requirements elicitation steps (e.g. A1 to A3 in e^{3}value). Due to space limitations we can not depict the whole workflow of e^{3}value, REA, and UMM in one figure. Hence, each perspective is described by its own workflow. The figure should help the reader to get the big picture of the worksheet-driven approach and to demonstrate the semantics of the annotation of the step-by-step guide.

The steps marked by a pencil, denote that the business analyst has to fill a worksheet manually in order to get to the next step (e.g. A1). The results are worksheets delivered by each step for e^{3}value, REA, and UMM. Those are annotated by a document icon. As already mentioned in the previous sub-section, the worksheet-driven approach is able to create all kinds of modeling artifacts. As a matter of fact, the completed worksheets generate modeling artifacts of the corresponding modeling perspective (e.g. A2). This is denoted in Figure 7.4 by annotating these steps as *automatic*. In contrary, *manual* means that the business modeler needs to model a specific artifact of the B2B model by hand (e.g. A3). Those steps which cannot be automated relate to the design of diagrams such as process flows (capturing a graph in a table-based worksheet may evidently become a complex undertaking). It should be noted that Figure 7.4 shows only one iteration of the B2B development process. In fact, the development process of a B2B model is iterative and the business analyst may even decide to return to any preceding step in a single iteration if required.

worksheets of each modeling perspective capture different requirements

A traditional - not worksheet-driven - B2B development process would not involve the steps marked by a pencil, and all the artifacts have to be created manually. A worksheet-driven approach must start with the selection of appropriate WDL definitions (Figure 7.4 D). The business analyst may either use standard worksheets, as for example provided by our approach, or may create customized work-

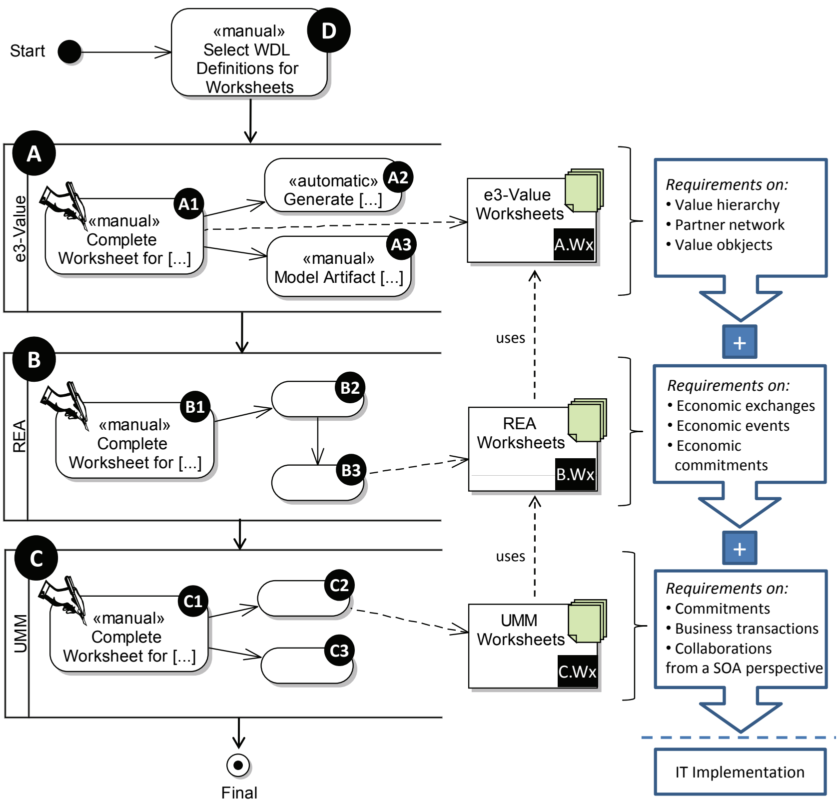


Figure 7.4
The "big picture" of the step-by-step guide

sheets using the WDL to reflect the specific requirements of the business domain under consideration. At the right hand side of Figure 7.4 we illustrate that the worksheets of each modeling perspective capture different requirements. The overall requirements specification of the whole B2B model will be enriched by completing one worksheet after the other. This fact is schematically visualized by the plus symbol. In the following we zoom into Figure 7.4 and describe each workflow delivering the required artifacts of each perspective.

Workflow of the network perspective (e^{3}value)

The e^{3}value perspective requires three different types of worksheets

Figure 7.5 shows the step-by-step guide for developing the network perspective by e^{3}value. First the business analyst has to complete the *consumer need* worksheet (A.W1) in step A1. Within this step, the business analyst defines the value objects that are needed in order to satisfy the consumer need. The information captured in this worksheet serves as input for modeling the value hierarchy (A2). In Section 7.3.2, the use and purpose of a value hierarchy is explained in detail. After having specified the consumer need and the value hierarchy, the actors need to be described from an economic point of view in A3. This results in the *e^{3}value actor worksheet*. From the information gathered in this step, the business partner network is generated automatically in A4 - i.e. all actors involved in this B2B model are created. The next step (A5) covers the main principles of e^{3}value - the value exchanges. Worksheet A.W3 contains all necessary information to document these value exchanges. By doing this, it is necessary to know, which business partners are already foreseen in the partner network. This information is captured in worksheet A.W2. Thus, the *value exchange worksheet* A.W3 references to the related information in A.W2. This scenario is denoted by the *use* arrows. From the information gathered in the *value exchange worksheet* we can automatically generate all value exchanges between the actors (A6). The only manual modeling step in this workflow is the modeling of the e^{3}value scenario path and its occurences (A7). Since the output of e^{3}value is a profitability sheet, the valuation part of worksheet A.W3 captures all necessary information for the automatic creation of the profitability sheet (A8).

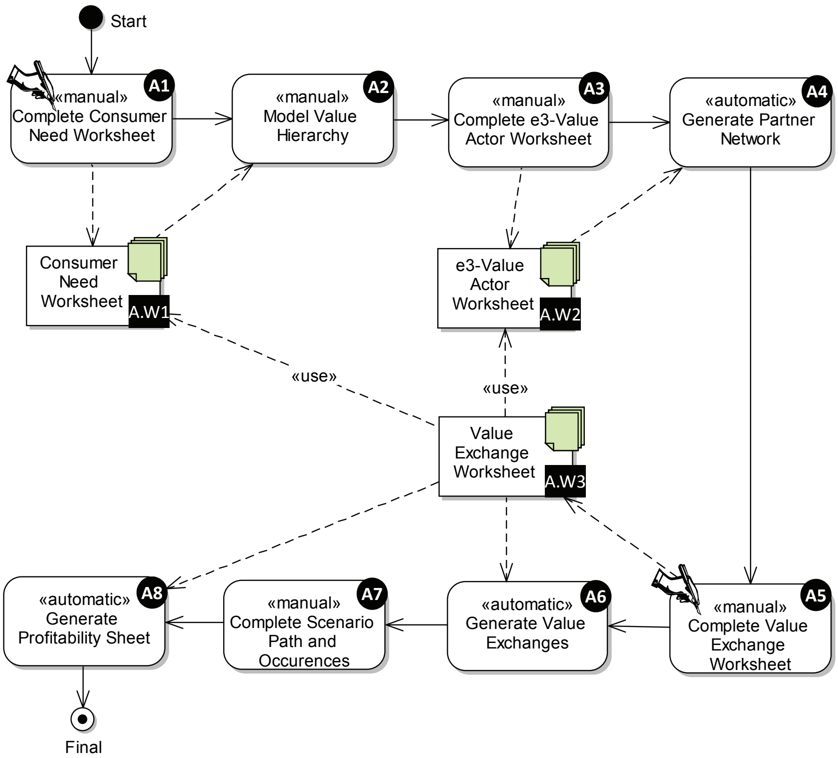


Figure 7.5 *Workflow for developing the network perspective (e^{3}value)*

Workflow of the trading partner perspective (REA)

REA uses worksheet information already gathered by e^{3}value

The workflow for the REA methodology is depicted in 7.6. This perspective contains only two types of worksheets. The first one is called *REA core worksheet set* (B.W1) and the second one is called *REA concepts worksheet* (B.W2). The *REA core worksheet set* consists of four different sub-worksheets covering the core concepts of REA: economic exchanges, economic agents, economic events, and economic resources. It is important to notice, that the *REA core worksheet set* uses some information gathered in the requirements elicitation phase of e^{3}value. This applies for the *e^{3}value actor worksheet* (A.W2) and the *value exchange worksheet* (A.W3). In B1, the business analyst completes the worksheet for the economic exchanges. In B2, the economic agents are specified. This worksheet inherits information from the e^3*value actor worksheet*. In step B3 and B4, the *event worksheet* and the *resource worksheet* is completed. The result of the requirements elicitation by the *REA core worksheet set* is the automatic generation of the REA constellations (B5). Since, REA covers additional concepts, such as economic commitments, economic phases, etc., those concepts are captured in REA concept worksheets B.W2 (B6). Finally, those concepts must be added manually (B7) to the REA constellation already created by B5.

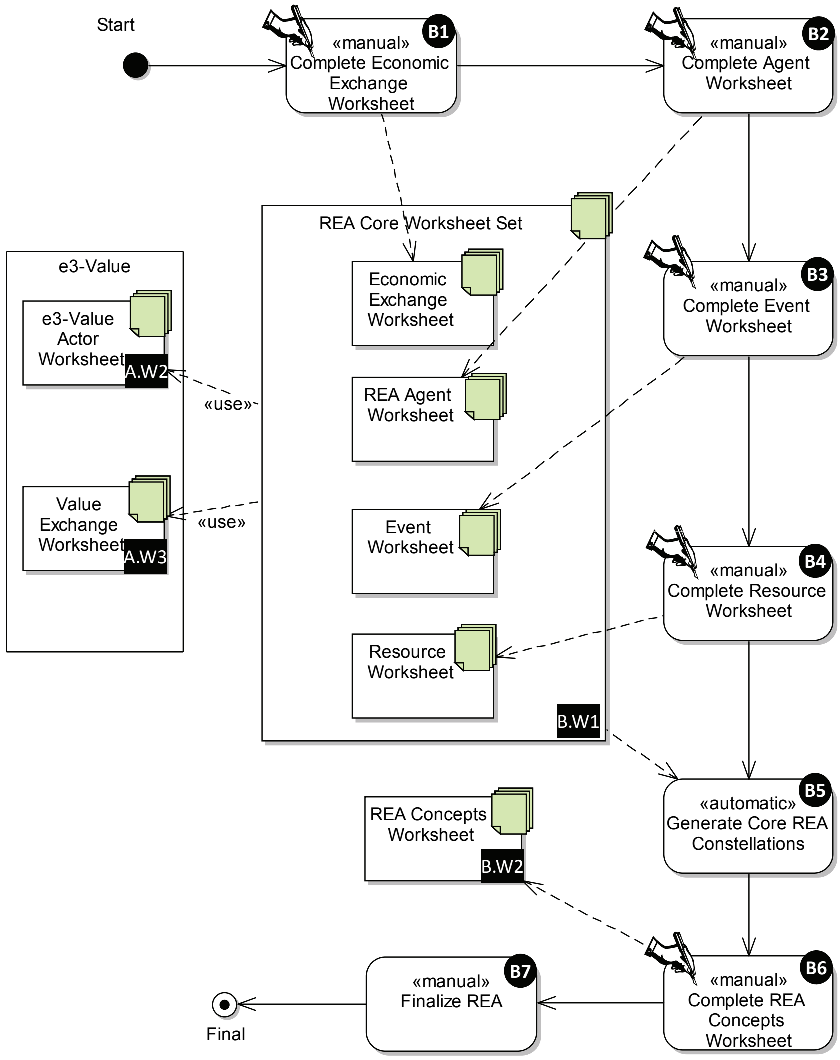


Figure 7.6 *Workflow for developing the trading partner perspective (REA)*

Workflow of the process perspective (UMM)

UMM requires different types of worksheets on its way to a business collaboration protocol

The workflow for developing a UMM compliant B2B model by the use of worksheets is depicted in Figure 7.7. First, the business analyst has to complete the *business process worksheet* (C.W1). Similar to the REA approach, UMM worksheets also require information gathered in previous phases. In case of the *business process worksheet*, it links to the *REA core worksheet set* to integrate the requirements captured in REA into UMM's business requirements view. As a result of this core worksheet set, the business processes is created (C3). As already said, that it is not possible to automatically generate a sequential flow of different activities, the business process activity model needs to be modeled by hand (C4). In step C5, the worksheets for the business entities (C.W2) need to be completed. The information gathered in these worksheets is significant enough to generate the business entities and the business entity lifecycle (C6 and C7). This step concludes the business requirements view of UMM.

For the business choreography view, the modeler must start with the completion of the *business transaction use case worksheet* (C.W3) in order to generate the business transaction use cases (C9). A business transaction use case captures the requirements for the actual business transaction. Thus, a *business transaction worksheet* (C.W4) uses the information gathered in the previous worksheet for the business transaction use case (C.W3). Since a business transaction follows always the same pattern, it is easy to semi-automatically generate the activity graph for a business transaction in C11. The worksheet C.W5 captures the requirements for a business collaboration use case, which is completed in step C12. The outcome of this worksheet is the generation of the business collaboration use cases (C13). Finally, the business collaboration protocol must be modeled by hand (C14).

Within this sub-section we have demonstrated the sequence of the steps that needs to be processed for our worksheet-driven approach. In the following, we explain the approach by an example and provide the reader with instances of worksheets from the customer acquisition use case.

7.3.2 e^{3}value worksheets

Defining the consumer need for modeling an e^{3}value model

Before developing an economic sustainable e-business partner network, the stakeholders must be clear about the consumer need. It shows why a consumer is interested in potentially obtaining products or services from the value web [51]. If the consumer need can not be determined explicitly, it does not make sense to invest into the development of a new IT system. The specification of the consumer need also helps to find the right business partners for your partner network. Once, the consumer need is defined explicitly, it is the business analyst's task to find the right instruments for satisfying the consumer need. In e^{3}value, those instruments are called value objects. The authors of [87] introduce the so-called value hierarchy, which depicts a prioritized ordering of the value objects that are used to satisfy the consumer need. In general, value hierarchies are used to articulate the increasing value that can be derived by the application of information technology within competitive organizations [194]. e^{3}value defines three elementary constructs for modeling a value hierarchy:

1. a **consumer need**, which states what a consumer desires at most
2. **value objects** capturing how a need is satisfied and

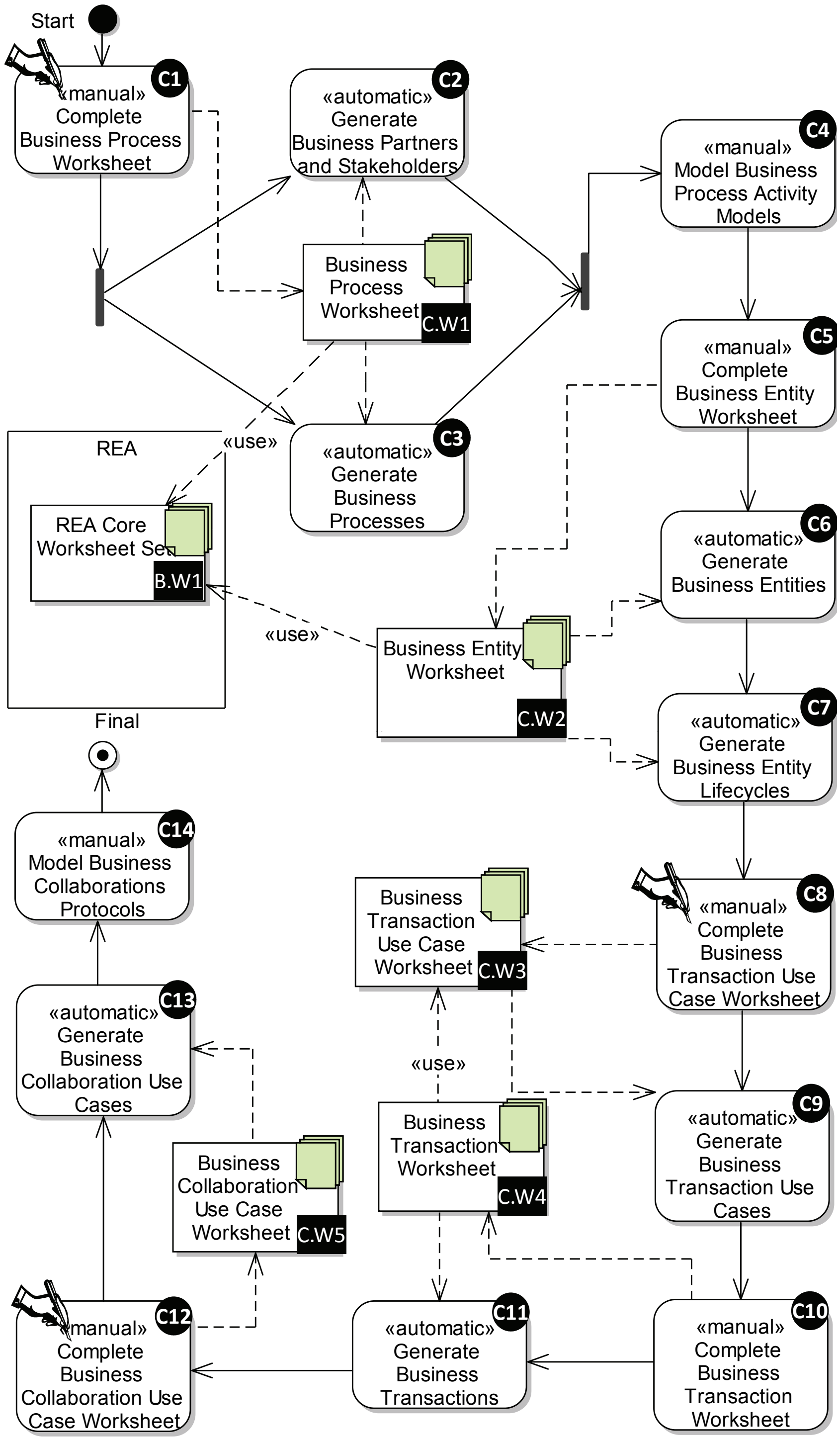


Figure 7.7
Workflow for developing the process perspective (UMM)

3. a **dependency path** between the value objects to depict relations

The e^{3}value *consumer need worksheet* consists of the first two constructs as mentioned above - the consumer need and the description of the value objects. Figure 7.8 depicts an example of this worksheet by means of our costumer acquisition use case. First, the business analyst has to investigate, who are the actual consumers in the network. In our case, the reader and the test-reader are definitely the end-users of the system. Their main consumer need is the delivery of a daily newspaper, which we call *newspaper service* in our model. After having specified the main consumer need, it is necessary to gather the value objects satisfying the consumer need. In the example, we distinguish between a long-term newspaper service (VO1) and a short term newspaper service (VO2). The long-term newspaper service is only given by a permanent subscription (VO3). Thus, the VO3 value object in the worksheet links to the dependent value object VO1. The same applies to the other value objects. In case of the short term newspaper service, an additional value object is required in order to satisfy the overall need - the advertising gift (VO3). The requirements elicitation by means of the e^{3}value *consumer need worksheet* is depicted as step A1 in Figure 7.5.

Form: *ConsumerNeed*	
Consumer Information	
Name	**Test-Reader, Reader**
Description	The reader and the test-reader are customer groups of the newspaper publisher. Both, the reader and the test-reader consume the newspaper service, which is the delivery of a daily newspaper. The reader has a permanent subscription and the test reader has a test-subscription for a shorter period of time.
Consumer Need	
Name	**Newspaper Service**
Definition	The delivery of a daily newspaper.
Description	Each customer gets a daily newspaper. The newspaper service consists of the delivery of the newspaper, as well as the additional customer services (monthly sport magazine, TV programs, online articles, etc...)
Value Object #1	
ID	VO1
Name	**Daily Newspaper (long-term)**
Definition	The delivery of the newspaper for a period of minimum 1 year.
Dependent Value Object	-
Value Object #2	
ID	VO2
Name	**Daily Newspaper (short-term)**
Definition	The delivery of newspaper for a period of 1 - 3 month.
Dependent Value Object	-
Value Object #3	
ID	VO3
Name	**Permanent Subscription**
Definition	Ensures the delivery of a newspaper for at least one year and increases the coverage for the newspaper publisher in long term.
Dependent Value Object	VO1
Value Object #4	
ID	VO4
Name	**Test Subscription**
Definition	Ensures the delivery of a newspaper for a period of 1 – 3 month and increases the coverage for the newspaper publisher in short term.
Dependent Value Object	VO2
Value Object #5	
ID	VO5
Name	**Advertisement Gift (Goody)**
Definition	A goody is used to pitch test-subscription to a test-reader.
Dependent Value Object	VO2

Figure 7.8
e^{3}value consumer need worksheet

As mentioned before, the *consumer need worksheet* only covers the consumer need itself, the value objects, as well as their dependencies. It does not contain the relations between the value objects, which is realized by the dependency path. The dependency path has already been introduced in the e^{3}value section of Chapter 5. Regarding value hierarchies, a dependency path (also named scenario path in e^{3}value models) connects value objects in order to show whether a value objects obtains another value object to satisfy the customer need. As we already know, *AND* and *OR* elements are used as elements by the dependency path of e^{3}value. The *AND* element combines value objects denoting that *all of them* are required to obtain the value object one level above. For instance, a daily newspaper service requires a test subscription AND an advertisement gift (cf. Figure 7.9). In contrary, an *OR* element is used to show, that in order to satisfy a need or to obtain a value object, one value object out of a set of alternative value objects is required. The modeling of the dependency path must be done manually in step A2 of Figure 7.5. Figure 7.9 depicts an example of the value hierarchy for satisfying the reader and test-reader. Note, triangles between the value objects denote *OR* elements, the rectangles represent *AND* elements.

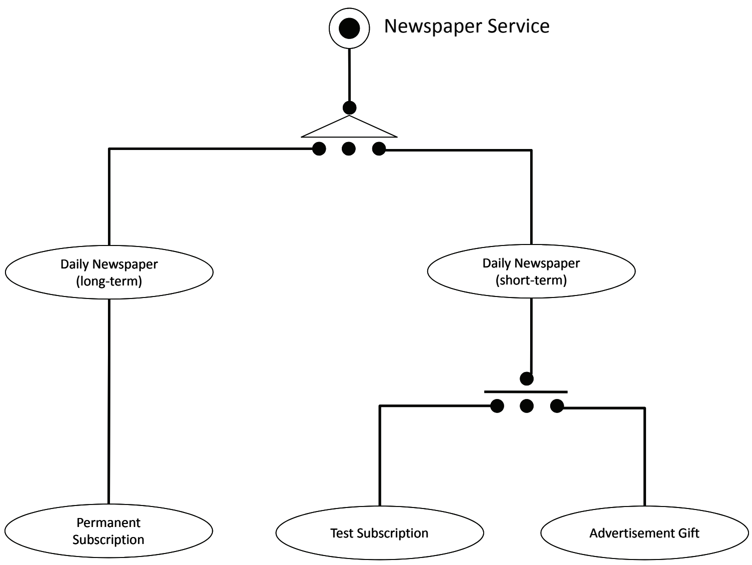


Figure 7.9 *The value hierarchy for satisfying the consumer need*

Capturing the requirements for modeling the actors within e^{3}value

Within the next step of the e^{3}value guide in Figure 7.5, the business analyst has to describe the actors used within the business partner network (A3). This is done by the e^{3}value *actor worksheet* (W.W2). An example of this worksheet is depicted in Figure 7.10. Due to space limitations, the worksheet is not shown in its full length. The description has been limited to a certain amount of characteristics in order to provide best readability. Having a deeper look at the worksheet, we see the requirements of each actor captured in a list. Besides the name, a unique identifier and a description, we also propose to capture candidates for the actor element in an e^{3}value model. For instance, the test reader could be a customer segment such as students or Austrian citizens. Another interesting element in the actor worksheet is the *type* property. In e^{3}value we distinguish between two types of actors: the first one is a regular actor as described in Chapter 5, the second one is called market segment. Following the definition of Gordijn [49], an e^{3}value market segment is defined as a concept that breaks a market (consisting of actors) into segments that share common properties. In terms of e^{3}value this means, that a market segment composes a number of actors who exchange objects having assigned the equal economic value. The number of actors within a market segment is denoted by the *count* variable in the actor worksheet. Finally, it is most likely that the actors, who are no consumers, have a fixed amount of expenses. This value is most important for the calculation of the profitability sheet.

Form: *Actors*	
Actor #1	
Name	**Test-Reader**
ID	A1
Description	A test-reader has a test-subscription and gets a daily newspaper for a […]
Candidates	Students, citizens of the eastern part of Austria, etc.
Type	Market Segment
Count	200.000
Expenses	-
Actor #2	
Name	**Third Party Vendor**
ID	A2
Description	The third party vendor sells a goody to the newspaper publisher. He […]
Candidates	Apple, Microsoft, Acer, Event-Ticket, Tefal, etc.
Type	Market Segment
Count	22
Expenses	€ 40.000.000,-
Actor #3	
Name	**Address Registry Provider**
ID	A3
Description	The address registry provider validates a set of addresses against a […]
Candidates	Herold
Type	Actor
Count	1
Expenses	€ 2.000.000,-
Actor […]	

Figure 7.10
The e^{3}value actor worksheet

The last worksheet of the e^{3}value methodology is the *value exchange worksheet* as depicted in Figure 7.11. A value exchange is set of value transfers that is bundled within one value interface. For instance, Figure 7.12 zooms into the e^{3}value example from Chapter 5 and depicts a value exchange. The magnifying glass highlights the value exchange between the third party vendor and the newspaper publisher. The actual objects of exchange are *money* against the *advertising gift* denoting the fundamental value exchange from the *order from quote* process. The focus within this worksheet lies on the description of the value objects. As we know, a value exchange is specified by value interfaces which contain the in- and out-going value ports. It follows, that from a requirements engineering point of view, it is interesting to capture the value objects and their direction of exchange. For example, the in-going value object for the newspaper publisher in the value exchange as depicted in Figure 7.12 is the *advertising gift*. The out-going value object from the newspaper publisher's point of view is *money*.

The valuation is used for generating the profitability sheet

An important issue of the description of the different value object is the valuation in terms of money. Each value object is specified by a certain monetary value. However, as you can see in the worksheet of Figure 7.11, the same value object (*advertising gift*) is valuated differently by each actor. This is due to the fact, that actors consume or produce the same value objects, but the monetary value is different for each individual. For example, the advertising gift's value for the newspaper publisher is EUR 25,- but only EUR 5,- for the third party vendor covering the production costs. An exception is the valuation of *money* as a value object, which assumes that it has the same value for both actors.

Form: *ValueExchange*		
Value Exchange		
Name	**Order from Quote**	
Description	The order from quote exchange covers the purchase order of a goody. The newspaper publisher requests a quote from the third party vendor and […]	
Involved Actors	Newspaper Publisher, Third Party Vendor	
Actor #1		
Name	**Newspaper Publisher**	
ID	A4	
In-going Value Objects	Value Object #1	
	Name	Advertisement Gift
	ID	VO5
	Valuation	€ 25,-
Out-going Value Objects	Value Object #1	
	Name	Money
	ID	VO6
	Valuation	€ 20,-
Actor #2		
Name	**Third Party Vendor**	
ID	A2	
In-going Value Objects	Value Object #1	
	Name	Money
	ID	VO6
	Valuation	€ 20,-
Out-going Value Objects	Value Object #1	
	Name	Advertisement Gift
	ID	VO5
	Valuation	€ 5,-

Figure 7.11 *The value exchange worksheet*

As mentioned in the e^{3}value guide for the business analyst in Figure 7.5, the *value exchange worksheet* retrieves other worksheets for information already captured. In this case, the information about the actors has already been specified in the e^{3}value *actor worksheet* (A.W2). Thus, the relevant information about the actors of the *value exchange worksheet* is referred to the data in the *actor worksheet*. The same applies to the value objects already captured in the e^{3}value *consumer need worksheet* (A.W1). Since we have specified the most important value objects in there, it is necessary to use this information within the *value exchange worksheet* as well.

7.3.3 REA worksheets

The REA core worksheet set comprises different worksheets

In comparison to e^{3}value and UMM, REA comprises a smaller set of worksheets. In Figure 7.6 we refer to the use of two different worksheets. The first one is the *REA core worksheet set* (B.W1), which contains the REA requirements of different sub-worksheets. The second one is used to gather information about the additional concepts of REA - such as economic commitments or business process phases. In Figure 7.13 we show an example of the *REA core worksheet set*. The overall worksheet contains four different types of sub-worksheets covering the basic concepts of REA:

1. Economic exchange worksheet
2. REA agent worksheet
3. Event worksheet
4. Resource worksheet

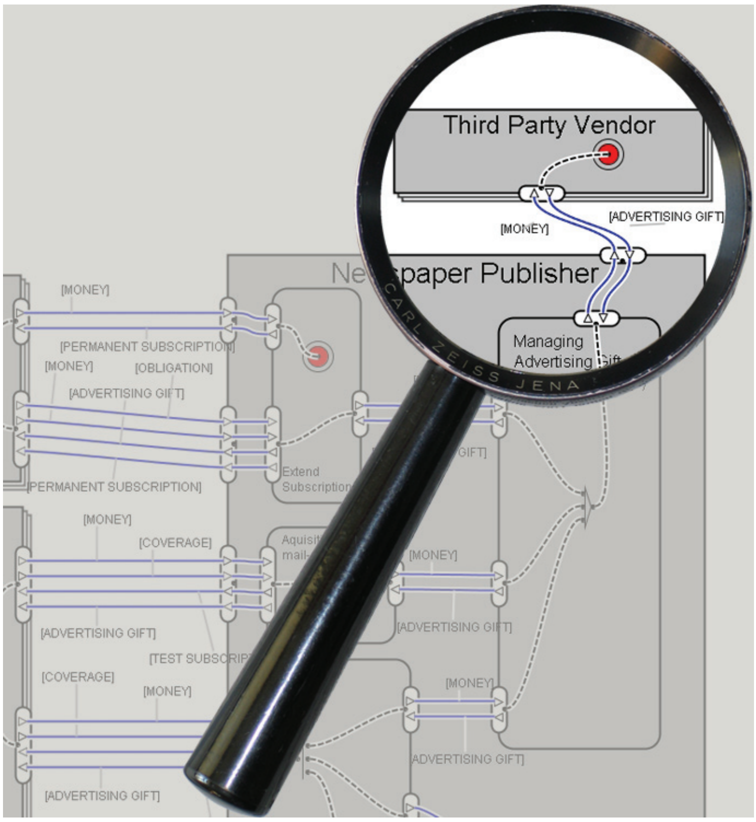


Figure 7.12 Zoom into the value exchange order from quote

A worksheet that captures the REA pattern

The *economic exchange worksheet* captures the information of each exchange from a trading partner perspective. Following McCarthy's definition of an economic exchange [118], the worksheet must contain the information about the REA pattern, which is based on the involved agents, the exchanged resources and the events triggering the exchange. In case of the *order from quote* exchange, the involved economic agents are the newspaper publisher and the third party vendor.

An important issue of the economic exchange is the description of the economic events. An economic event is an occurrence in time wherein ownership of an economic resource is transferred from one economic agent to another economic agent. Two or more economic events where one is the legal or economic consideration for the others are referred to as *duality events* in the worksheet. In our example the duality events are payment and shipment. Furthermore, we have to distinguish, which event is the initiating and which event is the terminating one. In the accompanying example the newspaper publisher only gets the advertising gift, if he pays for these goods. Thus, the initiating event is the *payment* and the terminating event is the *shipment* of the advertising gifts.

The economic agents and resources are also captured by worksheets

The remainder of the core worksheet set is a rather flat structured description of the REA artifacts. In the *REA agent worksheet*, the involved agents are described. Again, the worksheet uses information from previous requirements engineering steps. According to the mapping rules described in Chapter 6, e^{3}value actors are mapped to REA agents. Therefore, we use unique identifiers to reference to

Form: ***REACoreWorksheetSet***	
Economic Exchange Worksheet	
Economic Exchange #1	
Name	Order from Quote
Involved Agents	Newspaper Publisher, Third Party Vendor
Description	The economic exchange of the order from quote transaction deals with the exchange of goodies and their shipment. The involved agents […]
Duality Events	Payment, Shipment
Initiating Event	Payment
Terminating Event	Shipment
Resources	Money, Advertising Gift
Involved Commitments	Order, Shipment Schedule
Economic Exchange #2	
Name	Call Center Customer Acquisition
[…]	[…]
REA Agent Worksheet	
Agent #1	
Name	Newspaper Publisher
ID	A2
Event #2	
Name	Third Party Vendor
ID	A4
[…]	
Resource Worksheet	
Resource #1	
Name	Advertising Gift
ID	VO5
Description	[…]
Resource #2	
Name	Money
ID	R2
Description	[…]
[…]	
Event Worksheet	
Event #1	
Name	Payment
ID	E1
Description	The payment is an event where money flows from one agent to another agent. The precondition of this event is the invoicing event, which […]
Pre-Condition	There must be an invoice according to the *Order Commitment*
Post-Condition	The shipment of the goodies
Event #2	
Name	Shipment
ID	E2
Description	The shipment event deals with the delivery of the ordered goods. […]
[…]	

Figure 7.13 *The REA core worksheet set*

the actors captured in the e^3*value actor worksheet* (A.W2). The same applies to the *resource worksheet*, where some of the economic resources are linked to the value objects captured in the e^3value *consumer need worksheet* (A.W1). Finally, the *event worksheet* summarizes all economic events by a description, as well as by pre- and post-conditions.

Additional REA concepts

As we learned in Chapter 5, a REA model is not only defined by the REA pattern (Resource-Event-Agent), but it also has some additional concepts, such as economic commitments, economic agreements, business locations, and economic claims. Most of these concepts are documents which capture legal aspects of the economic exchange within the trading partner perspective. The according worksheet capturing the requirements for these additional REA concepts is depicted in Figure 7.14.

The worksheet starts withe the captured information about the different economic commitments. In REA, economic commitments play a major role for ensuring the adherence of a contract between business partners. Following the ISO/IEC 15944-4 specification [79], economic commitments are fulfilled by economic events. Furthermore, the economic commitments are the promised analogs of economic events which are connected by duality relationships. Thus, commitments also occur in reciprocal pairs where the promise of one business partner is requited by the promise of the other one. In our example worksheet, the requirements of the economic commitment *order* is depicted in more detail. It captures the information about the involved agents, the reciprocal commitment and the document reference, which represents the formalization of the commitment. Finally, the governed economic agreement is specified. In REA, economic commitments are bundled in economic agreements (similar to a contract) between business partners where, for example, a commitment to deliver advertising gifts is reciprocated by a commitment to pay for them (trade agreement, delivery conditions).

Form: ***REAConceptsWorksheet***	
Economic Commitments	
Economic Commitment #1	
Name	Order
Involved Agents	Newspaper Publisher, Third Party Vendor
Description	The order is the formal commitment to pay the delivered goods in advance before the shipment takes place. It serves as a confirmation[..]
Reciprocal Commitment	Shipment Schedule
Governed Economic Agreement	Trade Agreement, Delivery Conditions
Document Reference	DOC-Order2010
Economic Commitment #2	
Name	Shipment Schedule
[…]	[…]
Economic Agreements	
Economic Agreement #1	
Name	Trade Agreement
Description	The trade agreement contains information about the quality conditions of the goods, the order, the type of payment, the type of […]
Document Reference	DOC-TradeAgree2010
Economic Agreement #2	
Name	Delivery Conditions
Description	Within the document for the delivery conditions, it has been agreed upon the packaging of the goods, the transportation, the […]
Document Reference	DOC-DelCond2010
[…]	
Business Locations	
Business Location #1	
Name	Vienna – Newspaper Publisher Warehouse
Document Reference	DOC-BusLoc2010
[…]	
Economic Claims	
Economic Claim #1	
Name	Shipment Warranty
Description	If the shipment of the goods is delayed the third party vendor has to[…]
Materialized Event	Shipment
Settlement Event	Payment
Document Reference	DOC-EcoClaim2010
[…]	

Figure 7.14 *REA's additional concepts are captured in this worksheet*

REA also suggests to include the business location in the model. Adding business locations to the REA model is an optional part of the later analysis. The location is the place where the economic event takes place (i.e., its jurisdictional domain for accounting, customs, tax, etc.). In general, the location artifact has for example a certain dock capacity or safety requirements as its properties. Those properties are specified in the worksheet. In our case, the business location for the shipment of the advertising gifts is the newspaper publisher's warehouse.

Finally, the economic claims are described in the worksheet as well. The ISO/IEC 15944-4 specification [79] defines an economic claim as "..an expectation of one business partner to receive a future inflow of economic resources to another business partner because of an economic exchange, which is currently incomplete". Furthermore, it specifies that an economic claim is an optional materialization of a temporal imbalance in a duality relationship where an economic event has occurred without its requited correspondence to another economic event. An initial economic event materializes the claim, while the requiting economic event settles it. In case of our example worksheet the economic claim *shipment warranty* has an materialized event *shipment* and a settlement event *payment*. This means that if the shipment is not fulfilled according to the economic commitment and agreement, the payment event has to settle the economic imbalances.

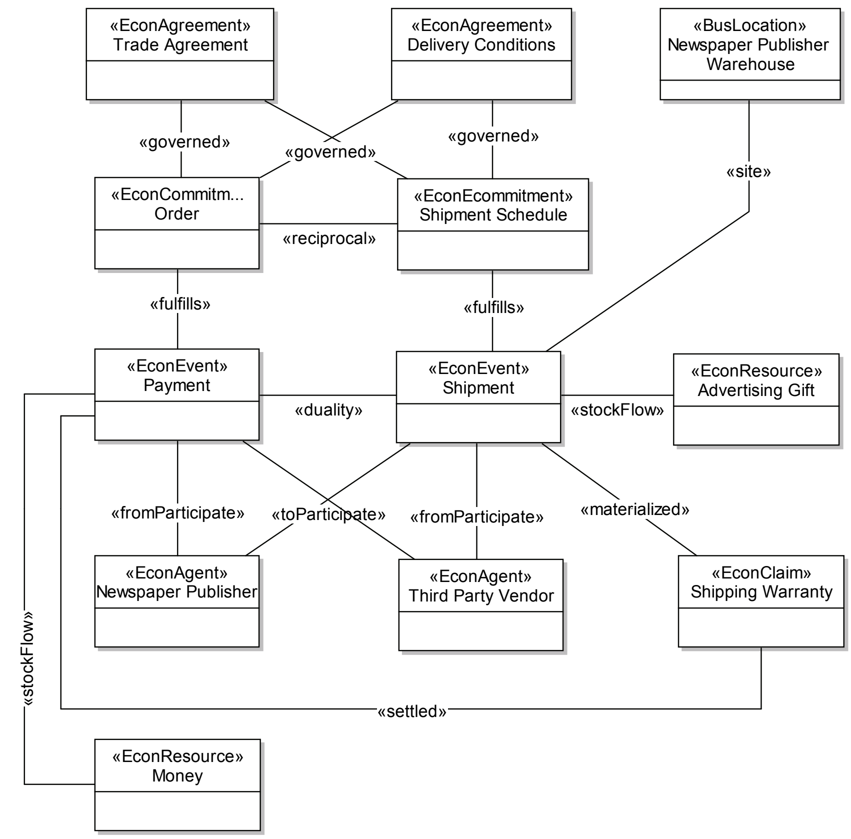


Figure 7.15 *The resulting REA model of the economic exchange order from quote*

Finalizing the REA model by hand

Following the REA step-by-step guide in Figure 7.6, the modeler is now able to finalize the REA model (step B7). The output of the requirements gathered in our example worksheets within this sub-section is shown in Figure 7.15 by means of a REA model. It shows an economic exchange of the *order from quote* scenario covering the basic REA pattern (Resource-Event-Agent) and the additional concepts (e.g. commitments, agreements, claims, etc). Again, the requirements gathered for the REA model, are an essential part for developing the whole IT environment for the B2B system. It considers the most important domain rules of an economic exchange by means of the give-and-take principles called *duality* and *reciprocity*. Moreover, an REA model captures the legal aspects that has to be fulfilled and the policy rules that has to be observed in order to execute business transactions between different business partners. The REA model as well as the REA worksheets serve as an essential input for developing the business process model and its corresponding worksheets by means of UN/CEFACT's Modeling Methodology in the next step.

7.3.4 UMM worksheets

We split this sub-section into the *business requirements view* and the *business choreography view*. The third view of UMM, the *business information view*, does not require any worksheets and is therefore not detailed in this sub-section. To demonstrate the worksheets used in UMM, we limit again the customer acquisition example to the order from quote scenario, where a newspaper publisher purchases a goody from the third party vendor.

Business Requirements View

Capturing existing process knowledge

At the beginning of the UMM development process, the business analyst gathers domain knowledge and existing process knowledge of the business domain under consideration. The analyst is required to understand the justification of the project and to determine its scope. He interviews business experts and other stakeholders to get an understanding of the existing business processes in the domain. The interviews ensure that all parties that are concerned with the to-be-designed business collaborations share a common understanding of the business domain. In this step, the analyst only discovers existing intra- and inter-organizational business processes, no new business collaborations are constructed. However, a business collaboration that is introduced in later steps of the UMM must respect the characteristics of the existing business processes and must not be in conflict with them.

The results of the interviews are *business processes* that are of interest for the domain under consideration. Those are captured in use case diagrams (Steps C1 to C3 in Figure 7.7), which may be generated from the information stored in worksheet C.W1. The business processes might be classified according to UN/CEFACT's Catalog of Common Business Processes (CBPC) [192], the Supply Chain Reference Model (SCOR) [184] or Porter's Value Chain (PVC) [160]. Classifying business processes facilitates the understanding of the business domain as well as of its scope. A hierarchical composition of business areas and process areas may be used to represent these or any other classification schemes.

UMM worksheets refer to requirements captured by REA worksheets

If necessary, a *business process* can be further detailed by using a *business process activity model* as depicted in Figure 7.17 (step C4). Before the business modeler starts to create the *business process ac-*

tivity model the requirements of the process are captured in the *business process worksheet* as shown in Figure 7.16. During the interview with the business domain expert the business analyst captures the necessary information of the business process and enters it into the worksheet. The information includes a definition and a description of the process as well as the participating parties and stakeholders. As mentioned in the step-by-step guide in Figure 7.7, the UMM worksheets link to requirements elicited during the REA modeling phase. In case of the *business process worksheet*, the data for the participating business partners, stakeholders as well as actions are adopted from the REA core worksheets set. The transfer rules of those values are in accordance with the mapping rules described in Chapter 7. Furthermore pre- and post-conditions and start/end characteristics are stored in the worksheet. After the worksheet is filled out completely, the modeler starts to create the *business process activity model* according to the requirements elaborated before.

Form: *BusinessProcess*	
General	
Business Process Name	Purchase Goody
Definition	A purchase goody taking place between a newspaper publisher and third party vendor.
Description	Subject of the business process is the purchase of goodies. The newspaper publisher requests a quote from the third party vendor. The third party vendor processes the quote and refuses or provides a quote to the newspaper publisher. Upon successful receipt of the quote the newspaper publishers places an order. The third party vendor either rejects or accepts the order.
Participants	Newspaper Publisher, Third Party Vendor
Stakeholder	none
Reference	Order from Quote
Start/End Characteristics	
Pre-condition	The type of goody is clear and the newspaper publisher is willing to order a certain amount of goodies.
Post-condition	- The goody is ordered. - No goodie purchase took place.
Begins When	Newspaper publisher knows the type of goody and the desired volume of the goodies.
Ends When	The newspaper publisher has placed his order.
Actions	- Request for quote - Place order
Exceptions	-
Relationships	
Included Business Processes	none
Affected Business Entities	Quote, Order

Figure 7.16 *Business Process Worksheet*

The activity diagram in Figure 7.17 depicts the *order from quote* scenario from the *purchase goody* part. The newspaper publisher requests a quote from the third party vendor. The third party vendor processes the quote and refuses or provides a quote to the newspaper publisher. Upon successful receipt of the quote the newspaper publishers places an order. The third party vendor either rejects or accepts the order.

The information exchanged between the business partners in the business process is about the *business entities quote* and *order*. A *business entity quote* is created with state *requested*. The pending state *processed* is either set to *provided* or *refused* by the responding *third party vendor*. In case of a positive quote the *business entity* order is set to *submitted* and further on either to *rejected* or *accepted*. These so-called *shared business entity states* must be in accordance with the *business entity lifecycle* of *quote* and *order*. As an example,

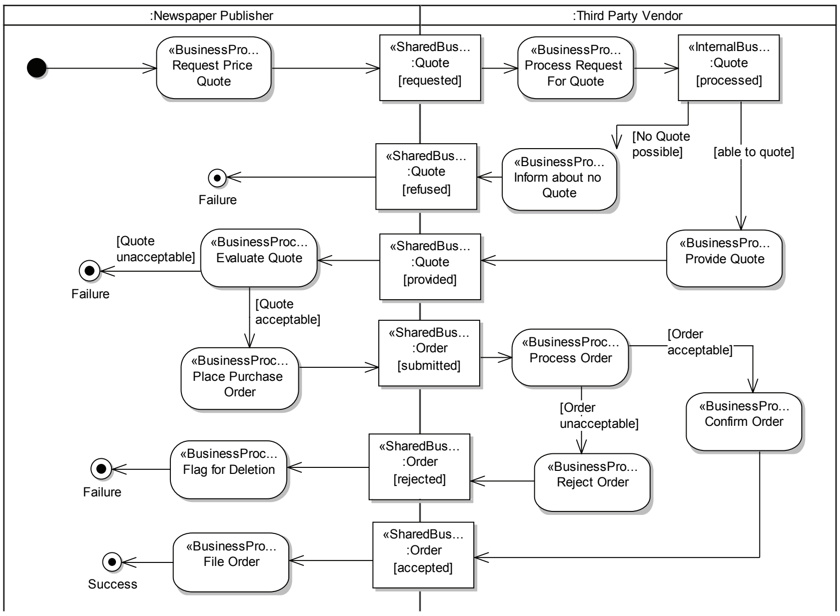


Figure 7.17
Business Process Activity Model

we depicted the business entity lifecycle of the business entity *quote* in Figure 7.19.

Form: ***BusinessEntity***	
General	
Business Entity Name	Quote
Definition	The quote business entity is the list of states a quote can have.
Description	A quote request is taking place between a newspaper publisher and a third party vendor.
BusinessEntityLifecycle	
Pre-condition	The newspaper publisher is willing to order a certain amount of goodies.
Post-condition	The quote has been refused or provided.
Begins When	A quote is initiated.
Ends When	The quote has been successfully provided or has been rejected by the third party vendor.
Exceptions	-
BusinessEntityState #1	
Name	requested
Definition	A quote is in state "requested" if the third party vendor has received the request from the newspaper publisher
Description	Before the third party vendor reports back to the notifier whether the quote is accepted or rejected, the quote is in state "requested".
Predecessing State	-
BusinessEntityState #2	
Name	processed
Definition	A quote is in state "processed" if the third party vendor has processed the request.
Description	Before the newspaper publisher is informed by the third party vendor about the successful or unsuccessful execution of the quote, the waste transport is in state "processed".
Predecessing State	requested
BusinessEntityState #3	
Name	refused
Definition	A quote is in state "refused", if the third party vendor negatively responds to the quote request of the newspaper publisher.
Description	If the quote is declined, the quote is in state "refused".
Predecessing State	processed
BusinessEntityState #4	
Name	provided
Definition	A quote is in state "provided" if the quote request was successfully processed.
Description	If the quote has been accepted by the third party vendor and the quote request was executed successfully, the quote is in state "provided".
Predecessing State	processed

Figure 7.18
Business Entity Worksheet

Both, the *business entity life cycle* and the *business process activity model* are strongly interlinked. Figure 7.18 depicts a worksheet capturing the requirements for a *business entity lifecycle* gathered during an interview between the business domain expert and the business analyst. First, a definition and description of the business entity lifecycle is stored as well as its pre- and post-conditions and its begin/end characteristics. In the next step definitions and descriptions for every business entity state are gathered. The generation of the *business entity life cycle* as depicted in Figure 7.19 is done automatically (steps C6 and C7 in Figure 7.7) by the worksheet editor using the information of the *business entity worksheet* (C.W2) shown in Figure 7.18. The entity graph is constructed by analyzing every life cycle state and its predecessing states. Furthermore, the description and definition for every life cycle state as stored in the worksheet are transferred into tagged values of the model artifacts.

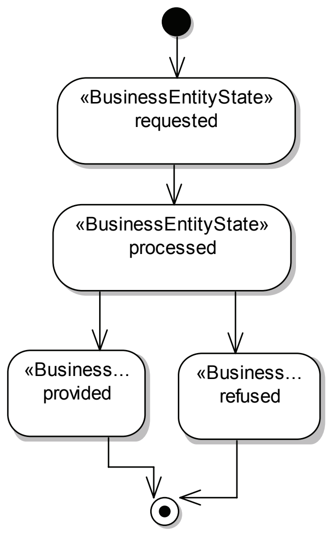


Figure 7.19 *Business Entity Life Cycle*

The worksheet for the business entity lifecycle is designed according to the information stored in the UMM meta model. Figure 7.20 shows the business entity part of the UMM meta model. For our tool support we took the information from the meta model (attributes and associations) and transformed it into corresponding WDL constructs. This WDL file is then used to populate the correct worksheet. In Section 7.4, we explain the technical implementation of the WDL files in more detail.

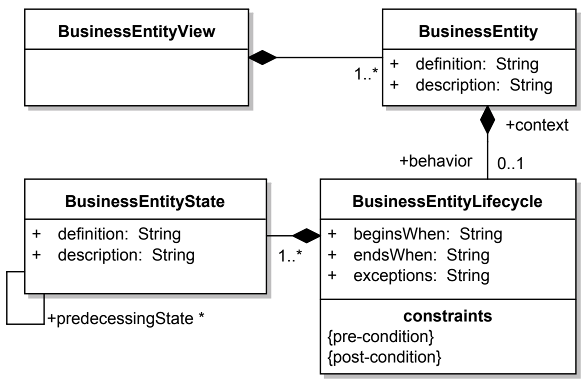


Figure 7.20 *UMM Meta Model for Business Entities*

Business Choreography View

In the BRV the business analyst identified needs for communication in a collaborative business process by means of *business process activity models* and state changes of *business entities*. In the *business choreography view* the business analyst builds upon these artifacts in order to develop formal models of *business collaboration protocols*. In UMM, a *business collaboration protocol* is a choreography of *business transactions*.

The concept of a *business transaction* specifies the information exchange between exactly two business partners. The need for an information exchange is identified by the means of the *shared business entity states* in the BRV. It is the task of a business transaction to align the business information systems of the collaborating business partners. In other words, the business transaction is responsible for keeping all relevant business entities in the same state in both information systems.

Synchronization of states is either realized in an uni-directional or in a bi-directional way. In the former case, the initiator of the *business transaction* informs the responding partner about an already irreversible state change the responder has to accept - e.g., the notification that the shipment of a goody has arrived. It follows, that responding in such a scenario is neither required nor reasonable. In the latter case, the initiating partner sets a *business entity* to an interim state and the responding partner decides about its final state - consider a request for quote that the responder might either accept or refuse.

In our *order from quote* example of the business process in Figure 7.17, we have identified six *shared business entity states*, whereby three of them are part of the business entity lifecycle *quote* and three of them are part of the business entity lifecycle *order*. This time, we consider the business entity *order*, which involves the states *submitted*, *rejected* and *accepted*. *Submitted* is a communicated interim state from the *newspaper publisher* that requires either the acceptance or the rejection of the quote. Thus, *provided* and *refused* represent final states. Given the business intention, the interim state *submitted* and the two final states *provided* and *refused* result in one *business transaction*. We call this business transaction *place order*. Analogous to the business transaction for the *place order* we need a business transaction for the *request for quote* scenario, which deals with the first three states of the business process in Figure 7.17. For demonstration purposes, we only detail the *place order* business transaction. Note, the use of business transactions is discussed in Chapter 5 and the full set of all delivered UMM artifacts is provided there.

As shown in Figure 7.21 the *order from quote business collaboration* consists of two distinctive *business transactions* namely *request for quote* and *place order*.

In the remainder of this sub-section, we will explain the semi-automatic generation of a *business transaction* using our worksheet editor by means of the example *business transaction place order*.

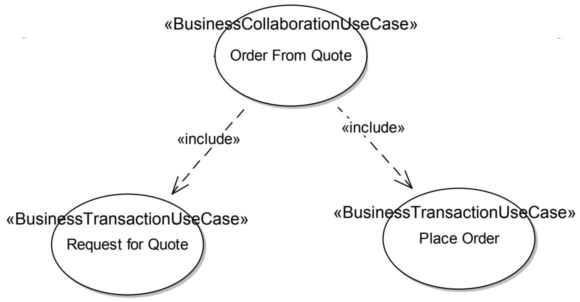


Figure 7.21 Business Collaboration Requirements

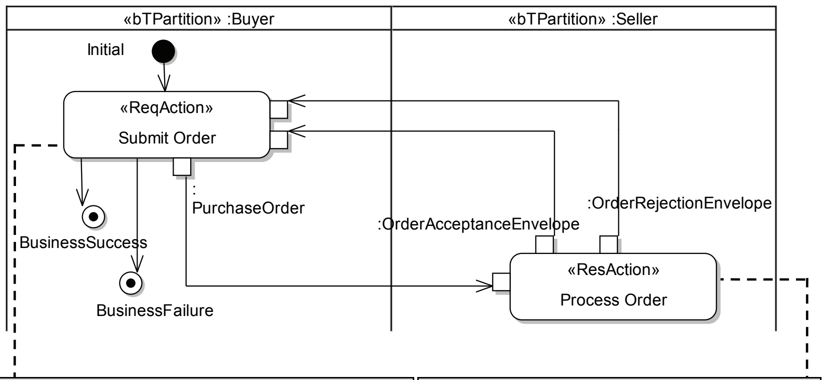
Business Transactions. Developing a *business transaction* starts with defining a *business transaction use case*. The *business transaction use case* captures the formal requirements of the to-be-developed business transaction. Again, the business analyst applies worksheets to gather the know-how that is required to specify the aspects of a business transaction. The corresponding worksheet (C.W3) is given in Figure 7.22. Beside the common information (name, definition and purpose) the worksheet gathers information such as the affected *business entities* - in our case *order* - as well as the participating roles (*newspaper publisher* and *third party vendor*) and their actions (*submit order* and *process order*). Based on the *business transaction use case* worksheet (C.W3) the worksheet editor creates the *business transaction use case place order* in step C9.

Furthermore, the worksheet C.W3 contains enough information to create the basic pattern of the *business transaction*. A *business transaction* is based on a UML activity diagram and follows always the same pattern: Two UML partitions are used to represent the two business partners. Each of them performs exactly one business action that sends and/or receives business information. The initiator's action outputs information that is input to the responder's action. The information flow in the reverse direction is optional.

Form: *BusinessTransactionUseCase*	
General	
Business Transaction Name	Place Order
Definition	"Place order" is used to provide the details of an order that is made by the newspaper publisher.
Description	"Place order" deals with the placement of an actual order. The newspaper publisher shall send signed copies of the completed order document to the third party vendor at least three working days before the shipment should start. The third party vendor either accepts the order or rejects it.
Requesting Role	Newspaper Publisher (Buyer)
Responding Role	Third Party Vendor (Seller)
Requesting Action	Submit Order
Responding Action	Process Order
Start/End Characteristics	
Affected Business Entities	Order
Pre-condition	A quote was successfully provided by the third party vendor.
Post-condition	The newspaper publisher is informed about an accepted or a rejected order.
Begins When	The newspaper publisher has received an accepted quote.
Actions	-
Ends When	The newspaper publisher receives an acceptance or rejection from the third party vendor.
Exceptions	The time frame of the place order is already exceeded.

Figure 7.22 Business Transaction Use Case Worksheet

In case of *place order* we collected the information shown in Figure 7.22 using the appropriate worksheet. *Place order* is performed between the *newspaper publisher* - playing the requesting role *buyer* - and an *third party vendor* - taking up the responding role *seller*. The *buyer* performs the *requesting action submit order* and the *seller* executes the *responding action* named *process order*. We already know that the execution of *place order* implies setting the *business entity order* to an interim state (*submitted*) before it is set to its final state (*accepted* or *rejected*). Thus, it contains a bi-directional flow of business information. Based on this information the worksheet editor generates the two partitions, assigns *buyer* and *seller* as responsible for the respective partition and creates the *requesting action submit order* in the partition of the *buyer* and the *responding action process order* in the *seller's* partition. This basic stub for a *business transaction* is completely created from the corresponding *business transaction use case* worksheet (cf. Figure 7.23). In order to complete the *place order* business transaction we require the following types of information: business transaction pattern, exchanged business information and quality of service parameters.



Submit Order (Activity)	
isAuthorizationRequired	true
isIntelligibleCheckRequired	true
isNonRepudiationReceiptRequired	false
isNonRepudiationRequired	false
retryCount	0
timeToAcknowledgeProcessing	PT0D10H0M0S
timeToAcknowledgeReceipt	PT0D6H0M0S
timeToRepsond	PT0D12H0M0S

Process Order (Activity)	
isAuthorizationRequired	true
isIntelligibleCheckRequired	true
isNonRepudiationReceiptRequired	false
isNonRepudiationRequired	false
timeToAcknowledgeProcessing	PT0D24H0M0S
timeToAcknowledgeReceipt	PT0D14H0M0S

Figure 7.23
Order from quote example: business transaction place order

Business transaction patterns differ in the default values of the tagged values

The business transaction pattern defines the type of a legally binding interaction between two decision making applications as defined in Open-edi [78]. We distinguish between two one-way (information distribution, notification) and four two-way (query/response, request/response, request/confirm, commercial transaction) types of business transactions. The patterns differ in the default values of the tagged values characterizing a *requesting/responding action*: *is authorization required*, *is non-repudiation required*, *time to perform*, *time to acknowledge receipt*, *time to acknowledge acceptance*, *is non-repudiation of receipt required* and *retry count*. These tagged values are considered as self-explanatory. We refer the interested reader to the specification [193] for more information concerning these parameters.

Form: *BusinessTransaction*		
General		
Business Transaction Name		Place Order
Definition		"Place order" is used to provide the details of an order that is made by the newspaper publisher.
Description		"Place order" deals with the placement of an actual order. The newspaper publisher shall send signed copies of the completed order document to the third party vendor at least three working days before the shipment should start. The third party vendor either accepts the order or rejects it.
Select Business Transaction Pattern		Request/Confirm
Secure Transport		true
Requestor's Side		
Requesting Role		Newspaper Publisher (Buyer)
Requesting Business Action Name		Submit Order
Time to Respond		PT0D12H0M0S
Time to Acknowledge Receipt		PT0D6H0M0S
Time to Acknowledge Processing		PT0D10H0M0S
Authorization Required		true
Non Repudiation Required		false
Non Repudiation of Receipt Required		false
Intelligible Check Required		true
Number of Retries		0
Responder's Side		
Responding Role		Third Party Vendor (Seller)
Responding Business Action Name		Process Order
Time to Acknowledge Receipt		PT0D14H0M0S
Time to Acknowledge Processing		PT0D24H0M0S
Authorization Required		true
Non Repudiation Required		false
Non Repudiation of Receipt Required		false
Intelligible Check Required		true
Business Information Envelopes		
Information Envelope from Requesting Business Activity		
	Information Name	PurchaseOrder
	Information State	
	Are Contents Confidential?	True
	Is the Envelope Tamperproof?	False
	Authentication Required?	False
Information Envelope from Responding Business Activity		
	Information Name	OrderAcceptanceEnvelope, OrderRejectionEnvelope
	Information State	
	Are Contents Confidential?	True
	Is the Envelope Tamperproof?	False
	Authentication Required?	False

Figure 7.24 *Business transaction worksheet*

Figure 7.24 shows the worksheet describing our example business transaction *place order* (C.W4 in Figure 7.7). It builds upon the information already gathered in worksheet C.W3. The *business transaction* worksheet C.W4 defines the respective quality of service parameters and the business transaction pattern. It is a request/-confirm *business transaction*, because the *buyer* requests the goody by submitting an order, which is either confirmed or refused by the *seller* playing the reacting part. The pattern of the *business transaction* dictates whether it is a one-way or a two-way message exchange. In case of a one-way message exchange, the worksheet editor permits only the definition of the request message. Otherwise, in case of a two-way business transaction the user is required to specify the type of the requesting business document as well as the responding business document types. Depending on the business intention of the business transaction, different response document types might be chosen to reflect a positive or a negative response to a request.

In order to define the exchanged business document types, the worksheet editor asks the business analyst to choose from a set of business document types. Modeling business documents is part of the *business information view* in the UMM. As already mentioned, we do not elaborate on the modeling of business information in this thesis, but assume that business information is already present in our model. Considering our example worksheet in Figure 7.24, the requesting business document sent by the *buyer* is of type *purchase order*. The *seller* may either accept or reject the order. Thus, we use an *order acceptance envelope* to represent a positive response. A rejection of the order is signaled by an *order rejection envelope*.

By having collected the information about the business transaction pattern, the exchanged business document types, and the quality of service parameters (or accepting their default values), we are ready to generate the entire *business transaction* from the worksheet information (step C11). In case of our example, the worksheet as given in Figure 7.24 results in the *business transaction* shown in Figure 7.23.

In a nutshell, the worksheet approach together with the worksheet editor supports the user in modeling business transactions efficiently. The benefit of our approach is two-fold: Firstly, the business analyst is guided through the various steps needed to particularize the exchange of business information between business partners. This guidance prevents the business analyst from producing non-compliant artifacts. Secondly, the worksheet editor relieves the modeler from recurrent modeling tasks. User input is elicited in a wizard-like manner and the resulting business transaction model is generated.

Business Collaborations. A *business transaction* defines only the exchange of one message and its optional reply between business partners in a B2B process. We already learned that UMM uses the concept of a *business collaboration protocol* to represent complex collaborative processes. According to our order from quote example, we have another *business transaction* in place dealing with the request for quote scenario. The resulting *business collaboration protocol* - we call it *order from quote* - is a sequence of, firstly, the *request for quote* business transaction and, secondly, our example transaction *place order*. The worksheets for business collaborations are modeled in step C12 and C13. It is not necessary to detail the worksheet for the *business collaboration use case* and *protocol*, since there is no automatic generation of UMM artifacts. However, the worksheet structure of those artifacts are given in [193]. As a final step, the business analyst needs to model the *business collaboration protocol* by hand (step C14).

7.4 Technical Implementation

The technical implementation of the worksheet editor is presented by means of UMM. Since UMM contains most of the worksheets and its tagged values play a major role in our approach, it is necessary to detail and explain the technical implementation by means of the process perspective. However, the concepts described in this section, such as the use of the flexible WDL language for customizing worksheets, can also be applied to the worksheets of e^{3}value and REA as shown in Section 7.3.

7.4.1 Customizing worksheets

The goals of the worksheets-driven approach have already been outlined in Section 7.1. As described there, we provide a pre-defined set of worksheets. However, it is not mandatory to use them exactly in the way they are defined. Different business domains usually have different business requirements. It is not possible to foresee the required domain-specific extensions in the worksheets. As a consequence the business analyst must be able to customize the structure of the worksheets according to his needs. Thus, static worksheets, hard-coding its content and layout, would be sub-optimal. Instead, the structure of the worksheets is maintained in XML files. Modeling tools, such as the UMM Add-In [63] dynamically load these XML files and render the worksheet according to its layout definition. The XML based language is called Worksheet Definition Language (WDL). Each worksheet is based on a WDL file specifying the nomenclature of the requirements entries, the clustering of entries into named categories, the design of the input boxes, the corresponding stereotypes, and their tagged values. The meaning of the different WDL elements is further on explained by Listing 7.2. For demonstration purposes, we provide worksheet definition files in WDL format for UMM. However, a business analyst may create new worksheets as well as extend or restrict existing worksheets according to special business needs.

Once the worksheets have been defined in WDL they are ready to be used within our UMM Add-In or any other modeling tool of choice. The business analyst either creates a new modeling element or selects an existing modeling element in order to bind it to a new worksheet entry. The binding is realized by activating a new worksheet when having selected the corresponding modeling element - the activation is a feature of the context-sensitive menu of the modeling element.

Once a worksheet is activated, the worksheet editor window pops up. The UMM Add-In offers two alternatives for selecting an associated worksheet (Figure 7.25): The first one opens a default worksheet associated with the stereotype of the selected model element. The second option allows the modeler to select any worksheet by specifying the URI of its WDL file.

A 1:n relationship between worksheets and stereotypes

A default worksheet is usually tailored to the needs of the corresponding stereotype as defined in the standard, not-customized meta model. This means that all tagged values of this stereotype are considered in the definition file. A worksheet may comprise information about more than one stereotype, i.e. the "master" stereotype and semantically related stereotypes. To better illustrate this fact, think of our example business transaction. A worksheet is associated to the "master" stereotype *business transaction*, but also covers, amongst other things, information of the related stereotypes *requesting/responding business action*. Since each WDL file is bound to a specific

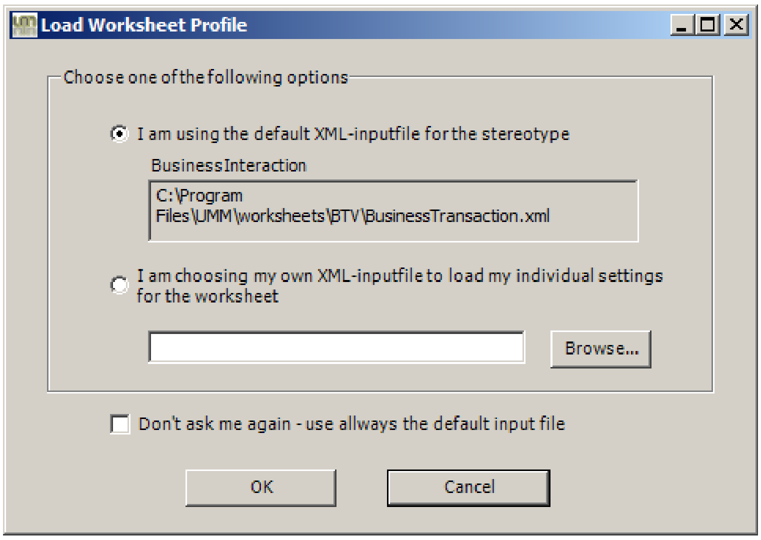


Figure 7.25 *Worksheet editor: choosing the WDL - template*

stereotype, there exists a 1:n-relationship between worksheets and stereotypes.

WDL files define the structure of a worksheet

In order to activate a default worksheet, it is necessary to specify the location of the underlying WDL files for the relevant stereotypes. This dependency of stereotypes and corresponding WDL files is defined in a property file. The XML based property file in Listing 7.1 shows the dependency between UMM stereotypes and default WDL files. UN/CEFACT defines a set of 9 different worksheets. Accordingly, Listing 7.1 specifies 9 dependencies, each specifying the stereotype and the corresponding URI of the default WDL file. For example, the entry in line 218 defines the dependency between the stereotype *BusinessTransaction* and the synonymously named WDL file. This WDL file defines the structure of the *business transaction worksheet* which is described further below and renders to the presentation depicted in Figure 7.26. All the default worksheets have to be placed in the same folder. Consequently, the *FOLDER* element in line 210 completes the URI of the default WDL file by adding a specific path.

If a modeler wants to change/extend/restrict the given set of default worksheets, he simple has to edit the property file. This is recommended if he wants to make a general change to the worksheets. Another option - recommended for more ad-hoc changes - is to use the second alternative as displayed in Figure 7.25. In this case he can bind a WDL file in an ad-hoc mode once the worksheet editor is activated for a selected modeling element. Nevertheless, the dynamically specified WDL file must be well-formed and valid. In order to avoid runtime errors, the worksheet editor checks well-formedness and validity before rendering the corresponding worksheet.

```
209 <!--Defining the Path-->
210 <FOLDER path="C:/UNCEFACT/UMM/Default-Worksheets"/>
211 <!--Defining the default WDL files-->
212 <STEREOTYPE name="BusinessDomainView" file="BusinessDomainView.wdl"/>
213 <STEREOTYPE name="BusinessArea" file="BusinessArea.wdl"/>
214 <STEREOTYPE name="ProcessArea" file="ProcessArea.wdl"/>
215 <STEREOTYPE name="BusinessProcess" file="BusinessProcess.wdl"/>
216 <STEREOTYPE name="BusinessEntity" file="BusinessEntity.wdl"/>
217 <STEREOTYPE name="BusinessTransactionUseCase" file="BTUseCase.wdl"/>
218 <STEREOTYPE name="BusinessTransaction" file="BusinessTransaction.wdl"/>
219 <STEREOTYPE name="BusinessCollaborationUseCase" file="BCUseCase.wdl"/>
220 <STEREOTYPE name="BusinessCollaborationProtocol" file="BusinessCollaborationProtocol.wdl"/>
```

Listing 7.1 *Mapping of default WDL files*

Both, default and ad-hoc worksheets have to follow the WDL schema. Listing 7.2 shows the XML Schema definition for the *ENTRY* element, which is the most important element within any worksheet. The definition of the elements for e.g. categorizing worksheets or structuring worksheets into subsections has been skipped.

The ENTRY element is responsible for the layout of each text field

The element *ENTRY* in line 222 is always an instance of the complex type *entryType* defined in line 228. This complex type specifies four child elements: (i) the mandatory and self-explaining *NAME* element, (ii) an optional *DEFAULT* element for specifying pre-instantiated values, (iii) a mandatory *TOOLTIP* element providing help on the nature of the element, and (iv) an optional *CHOICEBOX* element providing the items in a drop-down list. These elements are discussed in more detail during the explanation of the WDL example in Listing 7.3.

In addition, a set of mandatory attributes characterize an *ENTRY* element. The attribute *lines* (in line 235) specifies vertical size, i.e. the number of lines, of an input field. The *boolean* attribute *protected* (line 236) is *true* if the input field is in a write protected state and cannot be edited within the modeling tool. Figure 7.26 shows an example of a write-protected worksheet entry (denoted by A). This is due to the fact that the values for *requesting role* and *requesting business action name* must not be changed, because they are carried forward from another worksheet.

The attribute "type" is used to customize the entry fields

The attribute *type* (line 253) is used to distinguish the three major types of input fields: text fields, choice boxes and time values. These different options are predefined in the *formType* (line 255). If the attribute is set to the value *text*, the type of the entry value is a *string*. In this case, the input form for this worksheet entry is a simple text box (Figure 7.26 - D). In case of selecting *choice*, the input form is represented as a drop-down box (Figure 7.26 - C). The selection of *time* leads to a user friendly representation of time constraints. In UMM the representation of days, hours, minutes and seconds is combined in a single string, e.g. *PT4D3H2M1S* means 4 days, 3 hours, 2 minutes and 1 second. In the worksheet this string is split into four input fields, one for each time unit (Figure 7.26 - B).

The attributes *taggedValueName* (line 238) and *taggedValueType* (line 239) define the dependency between an input field of a worksheet and a tagged value of a stereotype. This means, the input in this field will be stored in the tagged value specified by *taggedValueName* of the stereotype given in *taggedValueType*. This approach also allows a very simple customization of the UMM meta model. The values for *taggedValueName* and *taggedValueType* may point to a tagged value of a stereotype that does not exist yet. In this case, the tagged value is created within the UMM Add-In on the fly. This approach allows for an easy extension of the underlying meta model not requiring any changes of its profile or code.

```
221 <!-- === Element definitions === -->
222 <xs:element name="ENTRY" type="entryType"/>
223 <xs:element name="CHOICEBOX" type="choiceboxType"/>
224 <xs:element name="ITEM" type="xs:string"/>
225 <xs:element name="NAME" type="xs:string"/>
226 <xs:element name="TOOLTIP" type="xs:string"/>
227 <!-- === Complex Types === -->
228 <xs:complexType name="entryType">
229         <xs:sequence>
230                 <xs:element ref="NAME"/>
```

Listing 7.2
W3C schema for WDL

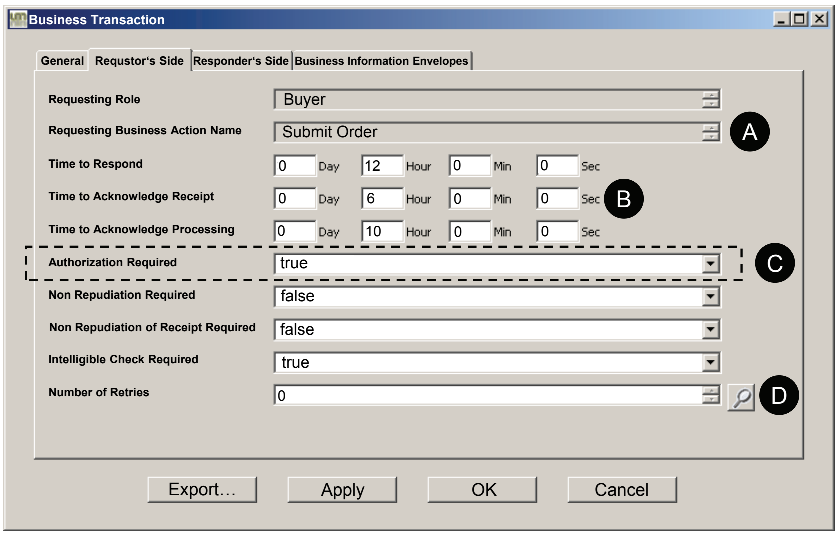


Figure 7.26
Different types of worksheet entries in the worksheet editor

```
                <xs:element ref="DEFAULT" minOccurs="0"/>
                <xs:element ref="TOOLTIP"/>
                <xs:element ref="CHOICEBOX" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute ref="lines" use="required"/>
        <xs:attribute ref="protected" use="required"/>
        <xs:attribute ref="type" use="required"/>
        <xs:attribute ref="taggedValueName" use="required"/>
        <xs:attribute ref="taggedValueType" use="required"/>
</xs:complexType>
<xs:complexType name="choiceboxType">
        <xs:sequence>
                <xs:element ref="ITEM" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute ref="selected" use="optional"/>
</xs:complexType>
<!-- === Attribute definitions === -->
<xs:attribute name="lines" type="xs:integer"/>
<xs:attribute name="protected" type="xs:boolean"/>
<xs:attribute name="selected" type="xs:boolean"/>
<xs:attribute name="taggedValueName" type="xs:string"/>
<xs:attribute name="taggedValueType" type="tvType"/>
<xs:attribute name="type" type="formType"/>
<!-- === Simple Types === -->
<xs:simpleType name="formType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="text"/>
                <xs:enumeration value="choice"/>
                <xs:enumeration value="time"/>
        </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tvType">
        <xs:restriction base="xs:string">
                <xs:enumeration value="standard"/>
                <xs:enumeration value="BusinessTransactionUseCase"/>
                <!-- === [...] enumeration of all UMM Stereotypes === -->
                <xs:enumeration value="RequestingBusinessActivity"/>
        </xs:restriction>
</xs:simpleType>
<!-- [...] -->
```

```
<ENTRY type="choice" protected="false"
        taggedValueName="isAuthorizationRequired"
        taggedValueType="RequestingBusinessActivity">
        <NAME>Authorization Required</NAME>
        <TOOLTIP>If the business transaction requires authorization
        set this worksheet entry to 'true'</TOOLTIP>
        <DEFAULT></DEFAULT>
        <CHOICEBOX>
                        <ITEM selected="true">true</ITEM>
                        <ITEM>false</ITEM>
        </CHOICEBOX>
</ENTRY>
```

Listing 7.3
WDL example

The code in Listing 7.3 is an extraction of the worksheet definition file of a business transaction. It shows the relevant tags for displaying the worksheet entry called *authorization required*. This entry is high-lighted by a dotted line in Figure 7.26. The entry is of type *choice* with the boolean values defined by the *ITEM* elements between the *CHOICEBOX* tags. The *tagged value name* attribute in line 273 specifies that the entry is stored in the tagged value of *is authorization required*. This tagged value is not part of the master stereotype *business transaction*, but defined within the related stereotype *requesting business action*. Consequently, this stereotype is referenced in the attribute *tagged value type* in line 274. The *NAME* element in line 275 defines the heading of the entry. The *TOOLTIP* element in line 276 captures instructions about the meaning, the content, and distinctive characteristics of the input field. During runtime, this note appears in form of a small tool tip, as soon as the business analyst drags the mouse pointer over the input field.

7.4.2 Generation of UMM modeling artifacts

Relieving the business analyst from the burden of manually modeling routine tasks

The integration of the worksheets into a UMM modeling tool is also used to drive the production of UMM artifacts. The creation of UMM artifacts usually requires know-how in UML modeling and the specifics of the UMM profile. The artifacts are created on the modeling canvas of a modeling tool. In our approach the worksheets drive the creation of some modeling artifacts. Since worksheets keep all the information about stereotypes and their tagged values, it is possible to (semi-)automatically create those artifacts that follow a predefined pattern. In this case no special know-how in UML modeling is needed to create the artifacts. This feature relieves the business analyst from the burden of manually modeling routine tasks. We demonstrated this feature in Section 7.3 by the example of a *business entity state lifecycle* and by the example of a *business transaction*.

Business transactions can be easily generated

We already know that a business transaction always follows the same pattern. This means it is always built by exactly the same kind of stereotypes. However, the participants in the business transaction, the names of the activities, and the names of the information flows differ from business transaction to business transaction. All this information is kept in the worksheet of a business transaction.

Figure 7.27 depicts an example for a semi-automatic generation of a business transaction. The generation is supported by the worksheet of a business transaction. When activating the worksheet of a business transaction, some input fields are pre-instantiated. This is due to the fact, that the WDL file references some stereotypes that have previously been created by other worksheets. Accordingly, the information already stored in the corresponding modeling elements (denoted by 1 in Figure 7.27) is transformed into the corresponding input fields of the worksheet (transformation A).

The worksheet editor generates the business transaction

The second part of Figure 7.27 presents a tab of the worksheet editor for business transactions. Each input box marked by a black arrow is pre-instantiated by previously gathered information. All other input fields require additional information to complete the definition of a business transaction. The pseudo code of transformation A in Listing 7.4 specifies how the worksheet editor retrieves information of the UMM model in order to pre-instantiate input fields. Between line 288 and 295 information from business transaction use cases and information envelopes is extracted. The pseudo code between line 297 and 308 calculates the two authorized roles partic-

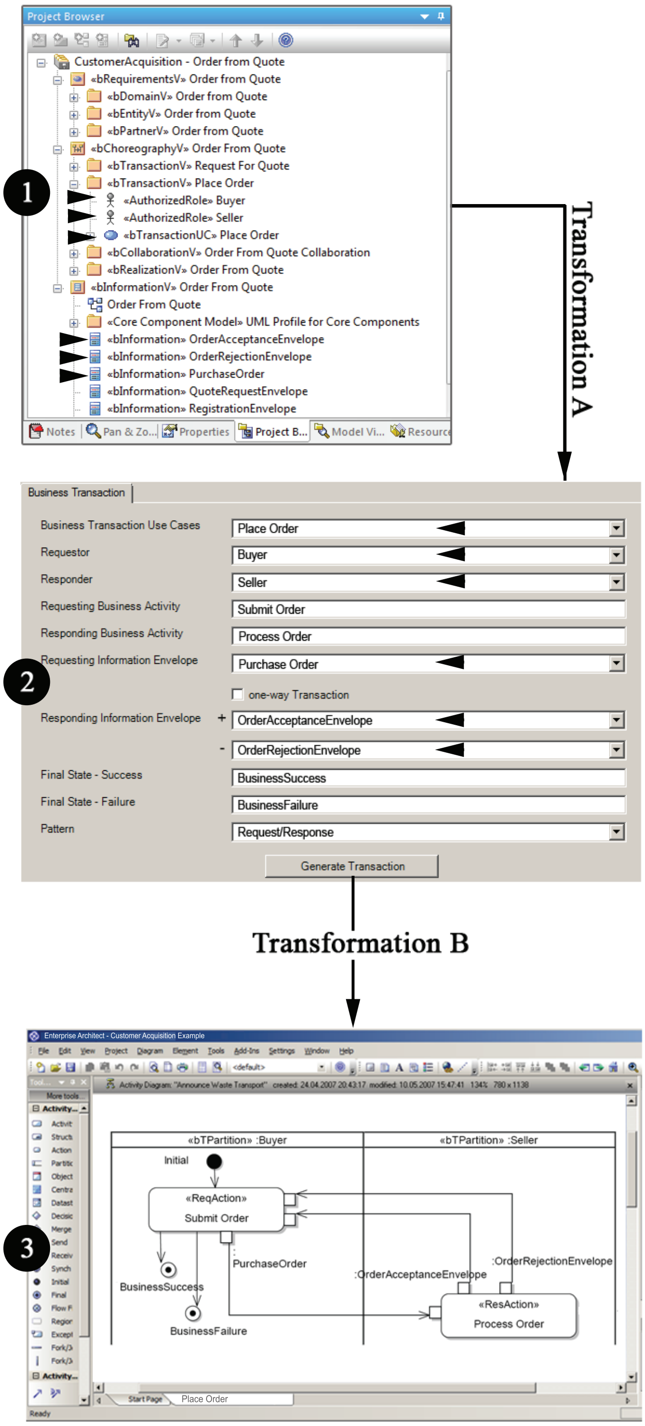


Figure 7.27 Worksheet editor - generation of a business transaction. 1: Transforming the modeling artifacts to the worksheet editor by using WDL, 2: Refining the parameters for the business transaction, 3: Generating the business transaction

ipating in a business transaction. These roles are known from a business transaction use case and are assigned to the input fields requestor and responder. Since it is not yet known who is the requestor and who is the responder, the two roles are arbitrarily assigned to the two input fields. However, as soon as the business process analyst changes the requestor to the other role, the responder role will change as well. Of course, this also holds for changing the role of the responder which triggers the change of the requestor.

Having completed the worksheet and hitting the *generate transaction* button automatically transforms the worksheet information to the activity diagram of a business transaction as presented in part 3 of Figure 7.27. The UMM Add-In generates all required modeling elements and creates the activity diagram based on the information gathered in the worksheet.

The code in Listing 7.5 is an excerpt of the UMM Add-In code for the semi-automatic generation of a business transaction. The code snippet represents a C-Sharp code using the Enterprise Architect's API. This snippet shows the generation of the business transaction swim lane, the requesting business activity, the initial state as well as the connection between the latter ones. Each element of the business transaction must be created prior to adding it to the activity diagram. When creating a new modeling element, it is also assigned to the appropriate namespace, which is a package or another UMM element serving as parent. For example, the creation of a requesting business transaction partition is depicted in line 313. Once the partition is created, the corresponding stereotype is set (line 315). An authorized role is added as a classifier to the partition. Which authorized role to assign as a classifier is known from the input field requestor of the worksheet (line 316).

```
define Vector BusinessTransactionUseCases
define Vector InformationEnvelopes
define Array AuthorizedRoles[2]
//Retrieving the UMM model
while UMMmodel.hasElements
        if modelingElement is stereotyped as BusinessTransactionView
                add modelingElement.childElements to BusinessTransactionUseCases
        endif
        elseif modelingElement is stereotyped as InformationEnvelope
                add modelingElement to InformationEnvelopes
        endif
endwhile
//Representation
set selectedBTUC to BusinessTransactionUseCases.choosenElement
while slectedBTUC.hasAssociatedActors
        if selectedBTUC.associatedActor is stereotyped as AuthorizedRole
                add modelingElement to AuthorizedRoles
        endif
endwhile
if requestor is AuthorizedRoles[1]
        set responder to AuthorizedRoles[2]
endif
elseif requestor is AuthorizedRole[2]
        set responder to AuthorizedRole[1]
endif
```

Listing 7.4
Pseudo Code for Transformation A

Having automatically created a modeling element, the generator is also able to automatically put this element on the modeling canvas of the activity diagram (line 319). The parameter of the *AddNew* method represents the coordinates of the position of the element on the canvas. In order to connect two modeling elements by a control flow, their IDs must be specified as clientID (line 341) and supplierID (line 342) of the control flow object.

```
310 private void generateTransaction(EA.Element BTVpackage) {
311  EA.Diagram activityDiagram = new EA.Diagram();
312  //create the requesting partition
313  EA.Element requPartition = (EA.Element)
314     BTVpackage.Elements.AddNew("Requestor", "ActivityPartition");
315  requPartition.Stereotype = UMM_Stereotype.BusinessTransactionSwimlane;
316  requPartition.ClassfierID = getSelectedItem(this.authorizedRoles).ElementID;
317  requPartition.ClassifierName = this.requChoiceBox.Text;
318  //add the requesting partition to the activity graph
319  EA.DiagramObject DiagObRequestorPartition = (EA.DiagramObject)
320     activityDiagram.DiagramObjects.AddNew("l=47;r=362;t=-34;b=-584;", "");
321  DiagObRequestorPartition.ElementID = requPartition.ElementID;
322  //create the requesting business activity
323  EA.Element requActivity = (EA.Element)
324     requPartition.Elements.AddNew(this.requBATextBox.Text, "Activity");
325  requActivity.Stereotype = UMM_Stereotype.RequestingBusinessActivity;
326  requActivity.ParentID = requPartition.ElementID
327  //add the requesting business activity to the activity graph
```

Listing 7.5
C-Sharp code for generating a business transaction

```
EA.DiagramObject DiagObRequActivity = (EA.DiagramObject)
    activityDiagram.DiagramObjects.AddNew("l=94;r=316;t=-120;b=-195;", "");
DiagObRequActivity.ElementID = requActivity.ElementID;
//create the initial state
EA.Element actInitial = (EA.Element)
    requPartition.Elements.AddNew("Initial State", "StateNode");
//add the initial state to the activity graph
EA.DiagramObject DiagObInitial = (EA.DiagramObject)
    activityDiagram.DiagramObjects.AddNew("l=194;r=214;t=-63;b=-83;", "");
DiagObInitial.ElementID = actInitial.ElementID;
//create the control flow between initial state and requesting business activity
EA.Connector initFlow = (EA.Connector)
    requPartition.Connectors.AddNew("", "ControlFlow");
initFlow.ClientID = actInitial.ElementID;
initFlow.SupplierID = requActivity.ElementID;
//[...]
}
```

7.5 Final assessment

In this chapter, we have shown an approach that improves requirements engineering of B2B models by means of worksheets. Our developed approach covers all three modeling methodologies (e^{3}value, REA, and UMM) and provides four major benefits:

Integrating worksheets for e^{3}value and REA

First, the traditional worksheet approach as proposed by UMM 1.0 only captures the requirements on the process perspective. Such an approach disregards the economic drivers, which provide relevant input for UMM's business requirements view. Into our worksheet driven approach we integrated the value network perspective by means of e^{3}value worksheets and the trading partner perspective by means of REA worksheets. These worksheets enrich the overall requirements specification for developing the B2B processes by means of UMM.

Avoid inconsistencies between worksheets

Second, the worksheet driven approach avoids inconsistencies between worksheets capturing the business experts' knowledge and the modeling artifacts. This is realized by integrating a worksheet editor into a tool environment and using the model as a single repository for capturing both, functional/non-functional requirements and model design.

WDL is fosters a flexible worksheet design

Third, our approach is characterized by its flexibility to adapt to specific needs in capturing business requirements. The worksheets are neither hard-coded into the modeling tool nor based on fixed templates, but created dynamically by the tool. The content and layout of the worksheets is dynamically loaded from XML-based configuration files. These XML files are based on the developed language called Worksheet Definition Language (WDL). The WDL does not only specify the visual rendering of the worksheets, but also enables the customization of the worksheet content. This guarantees the flexibility to adopt the worksheets to specific business needs - e.g., the ones of a certain industry domain. If a worksheet is extended by a feature not present in the underlying meta model, a tagged value is created on the fly for representing this feature within the model. Furthermore, the WDL allows to cross-reference stereotypes and their tagged values that are maintained by other worksheets. Thereby, it guarantees that the same information is not gathered twice, but available whenever needed. We demonstrated the technical implementations of the WDL files by means of our UMM Add-In tool. However, the WDL can be applied to any other modeling tool to integrate the advantages of a flexible form-based requirements engineering. For instance, WDL can be applied to any UML tool for gathering requirements of an REA model, or to any tool that supports the syntax and notation of e^3value.

A prototypical implementation - the worksheet editor

Fourth, our prototypical implementation called worksheet editor supports the semi-automatic generation of modeling artifacts. The worksheet-driven development speeds up the modeling process and ensures consistent results. Based on the information gathered in the worksheets, several B2B modeling artifacts may be calculated and added to the model. On the one hand side the generation feature does not require any specific modeling know-how from the business analyst and on the other hand it relieves the business analyst from the burden of routine tasks. The semi-automatic generation feature was demonstrated by automatically generating business transactions. In a next step a resulting UMM model may be transformed to code that is processed by B2B systems. In this thesis we only deal with the requirements engineering process and, therefore, we do not describe the mapping of a UMM *business transaction* to WSDL and BPEL. The interested reader is further referred to our work in [65], which describes a very detailed mapping of UMM to BPEL. In summary, the approach presented in this chapter describes an integrated development process starting from worksheets, leading to business models and resulting in business process models. Thereby, we improve the overall quality of requirements engineering in B2B projects based on e^3value, REA, and UMM.

We put a special focus on the worksheets integrated into UMM, since it delivers a high number of artifacts during the development process of a B2B model. Compared to a pure UMM based approach, our worksheet-driven solution requires an initial step to configure the worksheets - if the default ones are not used - and additional steps to gather the worksheet information. This overhead is compensated by the facts that the business domain expert does not require any know-how in creating or even interpreting models and the expert's knowledge captured in worksheets can further be used to (semi-)automatically generate modeling artifacts, which otherwise have to be created manually. We are confident that our solution results in a better overall quality for capturing the requirements of a B2B solution. In the next chapter we proof this hypothesis by a first evaluation.

8 First Evaluation

Measuring the success of a framework, a methodology, or a software of an information system is definitely not an easy task [92] [23]. The problem is to find the right metrics to provide most valuable feedback about the evaluated solution [16]. A survey comparing different evaluation methods and frameworks is provided by Babar et al. in [10]. However, within this chapter, we discuss the results of a first evaluation of our methodological approach. Within this thesis, we demonstrated the requirements engineering approach for B2B processes by means of an accompanying example which is located in the area of print media. This use-case scenario has been applied to a real-life project in which an Austrian newspaper publisher and two major Austrian business consulting and software companies specializing in IT-based management solutions were involved. These project partners were invited to contribute in the evaluation of the approach in order to highlight pros and cons of the methodology.

A newspaper publisher and two business consulting companies participated in the evaluation

The remainder of this chapter is structured as followed. In Section 8.1, we describe the participants of the evaluation by means of their skills and industrial affiliation. In Section 8.2 we give an overview of the methodological approach for conducting the evaluation. The approach is based on three steps. The results of each step are presented in Section 8.3. Finally, a conclusion of this chapter is provided in Section 8.4.

8.1 Overview of the participants

The IT expert and the business expert

As already mentioned, there were several companies from the industry sector involved in the evaluation. The first partner is one of the Austria's leading newspaper publishers. Due to the obligation to maintain confidentially, this partner asked not to be published by name. The company offers a huge application domain with the respective domain knowledge, but also in organizing business processes and how to base them on a SOA architecture. In such a way it represents a real challenging test bed, where the results of this thesis can be applied to. There are two representatives of this company involved as participants into the evaluation. Again, the participants are named *NN1* and *NN2* for the sake of confidentially. *NN1* has IT skills since he works in the field of information technology since 1983. He occupied positions as programmer, application maintenance, project manager, head of customer care and team manager. At the newspaper publisher, he is the head of IT software solutions (development, integration, architecture and operations). His group consists of approximately 55 employees. NN1 is referred as an *IT* *expert* in the further description of the evaluation. *NN2* works since 25 years in the newspaper publisher's company. Within the first 15 years of his employment he occupied positions in the sales department, whereby he was responsible for the customer acquisition section. For the recent decade of his employment he moved to the IT department taking over the connecting responsibility between the IT and the sales department. NN2 is the leader of the customer acquisition campaign delivering the functional- and non-functional requirements for the implementation within the IT department. He can be considered as a *domain expert*.

The senior business analyst from BOC Group

Apart from the newspaper publisher, two business consulting companies were invited to evaluate the concepts delivered by this thesis. The first one is *BOC Group* which emerged as a spin-off from the Business Process Management System (BPMS) group at the University of Vienna. Since then, BOC has created a set of tools for business process modeling and management. With their broadly known tool ADONIS [164], they are a direct competitor of IDS Scheer [168]. The representative of BOC is *Robert Strobl* who has been working for BOC Group since its foundation in 1995. Further to being the managing director of *BOC Unternehmensberatung GmbH* in Vienna he is also a member of the management board of BOC AG. His consultancy focus lies in the implementation of business process management as well as process based risk and compliance management. His experiences focus on the requirements analysis and definition of modeling methods in the area of strategic and business oriented process modeling and analysis, as well as in the extensions of these modeling approaches towards process based application development. Robert Strobl is referred as a *senior business analyst* within the evaluation.

The business analyst from Paradigma

The second business consulting company is *Paradigma* which is also located in Vienna. Paradigma is a consulting company operating in the fields of retail, e-logistics and the public sector. They already have profound experience in conducting consulting in the media industry by realigning the distribution channels for newspapers and magazines for a customer in Austria. Furthermore, Paradigma has been involved in several EU-funded projects. In one of them, they have successfully employed the UMM for modeling supply chains. As such, they are actively participating and contributing to UN/CEFACT standardization efforts. The representative of Paradigma is *Nikolaus Sahling*, who has been working with Paradigma since 1997. He is managing the company's transport and supply chain management group since 2001. His area of responsibility covers project management, and working as a consultant in projects facilitating multimodal supply chains, business process and trade procedure analysis to improve information flows between supply chain partners, as well as creating infrastructures supporting the management of international supply chains. Since 2003 he is an Austrian delegate to UN/CEFACT's TBG and contributes actively to the development of standards. He is referred as a *business analyst* in further descriptions.

8.2 The methodological approach of the evaluation

The methodological approach of this thesis as described in Chapter 1.5 foresees to conduct an evaluation that follows different predefined guidelines and involves different types of participants. The evaluation of our approach is done by three steps. Figure 8.1 presents an overview of the different steps, the aim of each step, their means, as well as their involved participants.

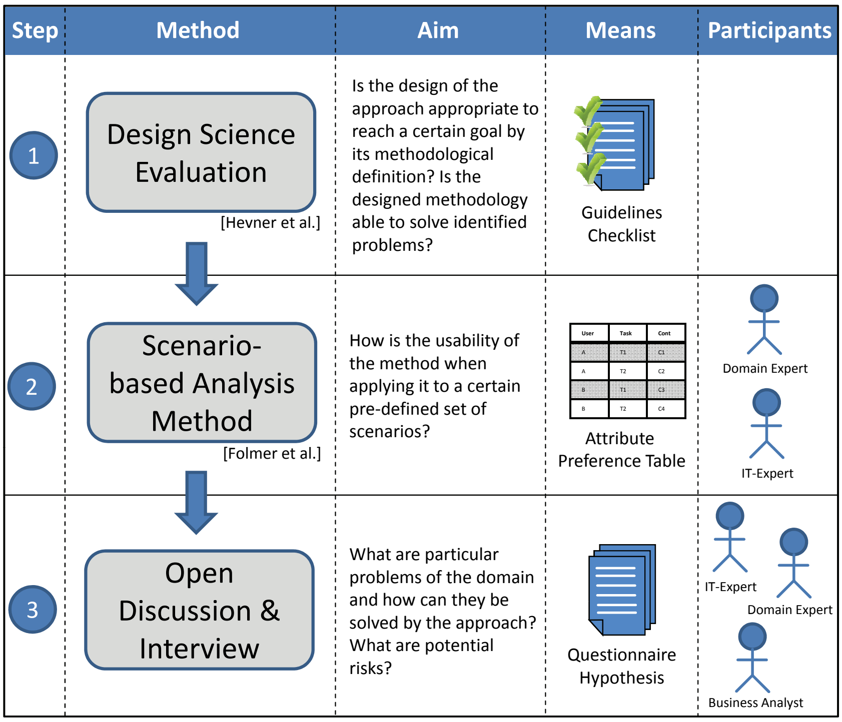


Figure 8.1 *Overview of the evaluation method*

The seven guidelines on Design Science by Hevner et al.

As a first step of our evaluation we use the seven guidelines provided by Hevner et al. [60] to validate our overall methodological approach (see step 1 of Figure 8.1). These guidelines helps us to evaluate our approach in terms of an effective Design Science research. The guidelines as outlines in Section 1.5 address issues on 1) the design as an artifact, 2) the problem relevance, 3) the design evaluation, 4) the research contributions, 5) the research rigor, 6) the design as a search process, and 7) the communication of research. However, it is not mandatory that all of the seven guidelines must be addressed in every research project [60]. It depends on the research area to determine when, where and how to apply each of the guidelines. For instance, guideline 6 (design as a search process) is not as relevant for our purposes as guideline 1 (design as an artifact). This evaluation was done by ourselves with the goal to investigate whether our approach is a "proper" one in terms of Design Science. However, some guidelines have been discussed with the business analysts to get an independent view on the observation of the guidelines.

Scenario-based Analysis Method by Folmer et al.

The second step of our evaluation is based on the *Software Ar-*

chitecture Analysis Method (SAAM) [84]. SAAM is a well-established framework for the evaluation of software artifacts by applying the delivered software to certain use case scenarios. This method is also known as *Scenario-based Assessment of Software Architecture* [12]. However, since our approach does not deliver an explicit software package we use an evaluation framework based on SAAM that focuses on the usability of a software architecture or methodology. Folmer et al. proposed such a framework in [38]. The goal of this framework is to validate the *provided usability* of the methodology whether it meets the *required usability* as expected by the domain and IT experts (see step 2 of Figure 8.1). Thereby, we introduced a new use case to the participants and defined several scenarios and tasks. The participants had to fulfill these tasks by applying the methodology and should provide feedback in regard to the usability of the methodology. The results were captured by so-called *Attribute Preference Tables (APT)*.

Final discussion as a conclusion of the evaluation

The third, and final step of our methodological evaluation resulted in interviews with all participants and a final open discussion. The objective of this step was to validate the hypothesis made in the beginning of the project. The hypothesis will be repeated in the next section when we discuss the results of this evaluation steps. The interviews were guided by the use of a pre-defined rudimentary opinionaire, covering basic questions such as:

- Is the approach understandable by different types of stakeholders?
- Is the definition of the methodology self-consistent and complete?
- How were B2B systems modeled in the past and what kind of issues were the modelers facing?
- Does the approach help to foster business modeling and business process modeling within the company?
- What are particular problems within the domain of print media in terms of Enterprise Modeling and how can our approach help?
- What are potential risks of the approach?

We strictly do not use the term "questionnaire" for this final evaluation, since the participants neither were supposed to answer all of the questions in a mandatory sequence nor should they get the impression of doing an exam. The questions rather served as an initiator of discussions in order to get the desired feedback from an industry perspective.

8.3 Evaluation results

Evaluation of the Methodology

Validating the seven guidelines by the business analysts

As introduced in the previous section, the design science approach was specified by Hevner et al. [60]. It is used to provide answers to the following questions: Is the design of the approach appropriate to reach a certain goal by its methodological definition? Is the designed methodology able to solve identified problems? Such issues are validated by using the guidelines as proposed by the authors. The following list provides summarizing results of each evaluated guideline:

- **Guideline 1: Design as an Artifact.** The approach delivers well-defined artifacts as well as a methodology that is clearly specified. Regarding the general requirements engineering approach (as described in Chapter 4) the method is defined by phases and iterations, in order to generate the desired artifacts toward the implementation of a new B2B system. Such a specification of phases come along with the definition of this guideline. The same applies to the business modeling based approach (as described in Chapter 6). The connections between the three interlinked methodologies e^{3}value, REA, and UMM are well defined and unambiguous. Moreover, the transformation rules in order to map the artifacts between the different methodologies are defined by formal specifications. Furthermore, this guideline foresees a precise and self-consistent definition of graphical representations of the delivered artifacts. The graphical notation for modeling process flows is based on UML and BPMN, which provide a well-understood vocabulary and symbols to make the solution transparent. In addition, REA is integrated into our approach to deliver models that have profound impact on the way in which B2B information systems are represented and developed.
- **Guideline 2: Problem Relevance.** Formally, a problem can be defined as the differences between a goal state and the current state of a system [60]. The fact, that most of the companies who are trying to "jump on the B2B train" have a big gap between the goal state and the as-is state, is confirmed by both business analysts. The business analysts also provide examples from different domains where they experienced such problems. One of the main problems is that companies disregard the economic drivers when developing B2B systems. They start with modeling the business processes and forget to make concerns about the economic sustainability of the system under development. The business modeling approach provided in this thesis considers such issues and thus, addresses important and relevant business problems.
- **Guideline 3: Design Evaluation.** Regarding the guideline on design evaluation mechanisms, Hevner et al. provide five different evaluation methods which should be used incrementally to improve the design of the approach: i) *Observational* (case study, field study); ii) *Analytical* (static analysis, architecture analysis, optimization, dynamic analysis); iii) *Experimental* (controlled experiment, simulation); iv) *Testing* (black box, white box) v) *Descriptive* (informed argument, scenarios). To improve our developed methodology we used the observational evaluation method. Thereby, we used a case study of a certain problem domain to get feedback from the stakeholders/users facing that problem. Furthermore,we also took advantage of

descriptive evaluation methods. For instance, the definition of certain scenarios built around the artifacts is a perfect way to demonstrate the utility of the method. As a critical reflection we need to state, that this evaluation method was only used for the final evaluation and not during the project.

- **Guideline 4: Research Contributions.** Effective design science must provide clear research contributions in the area of the designed artifacts. We are confident, that the five research contributions are explicitly specified, and more importantly, delivered by the approach: first, a general approach for business process based requirements engineering; second, the business modeling based approach by interlinking three well-established modeling methodologies; third, solving notational problems of two major modeling methodologies (REA and UMM) and improving their usability; fourth, formalizing transformation rules to map between the artifacts of the different methodologies; fifth, a worksheet-driven approach to guide modelers during the requirements engineering process in order to develop B2B specifications.
- **Guideline 5: Research Rigor.** This guideline addresses the way in which research is conducted. The design of a methodology requires the application of a rigorous method in the construction and evaluation of the designed artifact. However, this guideline applies primarily to methods that are based on a mathematical formalism and aims at validating criteria such as performance or adaptability. Thus, this guideline is not as important for the evaluation as the other guidelines. Nevertheless, Hevner et al. mentioned in the description of this guideline, that artifacts that are components of a human-machine problem-solving must be exercised within appropriate environments. Issues that are addressed include comparability, subject selection, training, time, and tasks. Therefore, we tested the approach by integrating our methodology into the tool environment of the BOC Group, which is called ADONIS [164]. ADONIS is a meta modeling tool tailored for the management of business processes. Since our approach is platform independent, this integration test was successfully conducted and resulted in positive feedback from both business analysts. Furthermore, parts of the methodology were developed as a Domain Specific Language (DSL) [124]. This provides a further example on *how well* the designed artifact works in different environments and not *why* the artifact works [60].
- **Guideline 6: Design as a Search Process.** This guideline addresses the need for an inherently iterative search process to develop a new methodology. As already mentioned, this guideline does not play a major role in the evaluation and thus, it is neglected.
- **Guideline 7: Communication of Research.** Following the overall message of this guideline, design science research must be presented both to technology-oriented as well as management-oriented audiences. The audience of the first category needs

sufficient detail to enable the methodology for implementation purposes. The management-oriented audiences need sufficient detail to determine if the organizational resources should be committed to constructing within their specific organizational context. In regard to our methodology, the key points of this guideline are definitely fulfilled. Our methodology delivers artifacts that are explicitly foreseen to be used for the management level (e^3value), for the IT level (UMM), as well as for both groups (REA). In summary, the business models (addressing managers or CEOs) are clearly understandable and deliver highly relevant economic input for further organizational decisions. Moreover, the business process models (addressing technicians) developed by UMM provide a significant requirements specification toward a service-oriented architecture.

Scenario-based Architecture Analysis Method

Introducing a new use case scenario for this study

After having conducted the evaluation of our methodology from a design science perspective, we validated the usability of the methodology by a scenario-based assessment. For this evaluation step, the representatives of the *newspaper publisher* were required to provide most valuable feedback about the usability of the methodology. The scenario-based evaluation was done within two steps. First, we demonstrated the methodology to the participants of the evaluation by ourselves. Thereby, we used a business scenario which was familiar to all participants - the customer acquisition example as used within this thesis. Second, we asked the domain and IT experts to apply the methodology by themselves to another use case, which addresses a similar problem in the print media domain. The participants were asked to model an end-to-end example by using our approach. The use-case scenario that was chosen as a demonstrator dealt with the management of the external procurement of printing paper. Since newspapers are printed on different types of paper, the sourcing of external paper vendors along the value chain requires - similar to the customer acquisition - several complex B2B processes. The evaluation was done within a one-day workshop at the newspaper publisher's headquarter.

Following Folmer et al. [38], the scenario-based evaluation method is based on a framework consisting of three concepts: i) *Usability Attributes* (satisfaction, learnability, efficiency, reliability), ii) *Usability Properties* (guidance, accessibility, feedback, error prevention), and iii) *Usability Patterns* (wizard, transformation patterns, undo, alert, progress indication). The first concepts evaluates the usability attributes. For instance, *learnability* is used to measure, how quickly and easily users can begin to do productive work. The second concept evaluates the heuristics and higher level design primitives which have known effects on usability. For instance, *guidance* is used to measure on-line guidance to support certain operations. The third concept evaluates the usability by means of patterns that are used to take delivery of recurring user tasks. For instance, the attribute *transformation patterns* evaluates to which extent the methodology supports semi-automatic transformation of artifacts.

Attribute Preference Tables are used to validate the usability

These usability metrics were captured during the evaluation by so-called *Attribute Preference Tables (APT)* as proposed by Folmer et al. APTs are used to rate the usability of certain scenarios or tasks by assigning quantitative values between 1 to 5 (1 means highly user-friendly, 5 means not at all user-friendly). However, we did not consider all attributes mentioned in the previous paragraph, since some of them were not applicable for modeling methodologies. Attributes that have been neglected are for instance: user feedback, user control, alter, or progress indication. The following list provides an overview on the six attributes that were evaluated and thus, included in the attribute preference table:

- **Learnability** (the ease of remembering the way the artifacts delivered by the methodology must be modeled)
- **Efficiency** of use (the number of tasks per unit time that the user can perform applying the approach)
- **Reliability** (the error rate in using the methodology and the time it takes to recover from errors)
- **Satisfaction** (the subjective opinion that users form in using the methodology)
- **Guidance** (the support during the modeling process)
- **Pattern** (the support of patterns to take over recurring user tasks)

Figure 8.2 depicts the scenario attribute table for the usability evaluation. As we can see on the left hand side, we identified 23 tasks which were fulfilled by both participants of the evaluation. The six different attributes that were subject of validation are shown on the right hand side. Each attribute was evaluated by both participants. The newspaper publisher's IT expert is referred as user A, and the domain expert as user B.

Interpretation of the results

For interpreting the results, we compare specific attributes that provide significant information about the usability of our approach. In order to demonstrate such special characteristics (e.g., attributes that are valuated in totally different directions), we highlighted some interesting results in Figure 8.2. For instance, the learnability attribute of task T4 differs significantly between the two users. Whereby the IT expert has rated the modeling of the e^{3}value scenario path as a rather "easy" task, the domain expert was not so satisfied in regard to learnability. In fact, it took a long time for the domain expert to understand the necessity and handling of this essential modeling step. However, this effect is evident, since the IT expert acquired workflow-oriented thinking from his daily work, which is a big advantage to understand the scenario path of e^{3}value . Another interesting difference of value pairs is provided in task T6 dealing with the automatic generation of the profitability sheet. Both users had no problem with the learnability curve of this task, but rated the reliability attribute as user-unfriendly. This is due to the fact, that the generation of the profitability sheet is an easy task once the model is correct and compliant to its meta model. However, if there are incon-

Scenario Attribute Preference Table: ***Paper Procurement***

User A: NN1 ***(IT-Expert)***
User B: NN2 ***(Domain Expert)***

Task	Description	Context	Learnability		Efficiency of use		Reliability		Satisfaction		Guidance		Pattern	
			A	B	A	B	A	B	A	B	A	B	A	B
T1	Modeling the *value hierarchy*	e^3Value	2	1	1	1	1	1	2	1	1	1	-	-
T2	Modeling the *partner network*	e^3Value	1	1	1	1	1	1	1	1	1	1	1	1
T3	*Value exchanges* between business partners	e^3Value	2	1	2	1	1	1	2	1	1	1	1	2
T4	Modeling of the *scenario path*	e^3Value	1	4	2	3	1	2	2	2	4	4	-	-
T5	*Valuation* of the exchanged value objects	e^3Value	1	2	1	2	2	1	1	1	1	1	-	-
T6	Generation of the *profitability sheet*	e^3Value	1	1	1	1	4	5	1	1	3	3	1	1
T7	Transform network view to *trading partner perspective*	e^3Value/ REA	2	4	2	2	1	1	1	1	1	1	1	1
T8	*Economic exchanges* for the trading partner perspective	REA	3	3	2	3	1	2	1	2	2	1	1	1
T9	Refining *economic agents*	REA	1	2	1	2	1	1	2	1	1	1	-	-
T10	Modeling of *economic events*	REA	2	3	2	2	2	2	2	2	1	1	-	-
T11	Definition of the *economic resources*	REA	1	1	1	2	1	1	1	1	1	1	-	-
T12	Defining *duality* relationships	REA	3	3	1	1	3	3	2	3	2	2	-	-
T13	Adding *additional REA concepts*	REA	3	3	2	2	2	2	1	2	3	3	-	-
T14	Transform REA model to *business process perspective*	REA/ UMM	2	4	1	2	1	1	1	2	1	1	1	1
T15	Complete *initial model structure* of UMM	UMM	1	2	2	2	1	2	1	2	2	2	2	2
T16	Refine *business partners* and *stakeholders*	UMM	1	1	1	2	1	1	1	1	1	1	-	-
T17	Modeling *business processes*	UMM	2	4	1	3	1	3	2	2	1	1	-	-
T18	Refining *business entities*	UMM	2	3	2	2	2	3	2	2	1	1	-	-
T19	Modeling *business entity lifecycles*	UMM	1	4	2	3	2	2	1	2	1	1	-	-
T20	Modeling *business transaction use cases*	UMM	1	3	2	2	2	2	1	2	1	1	-	-
T21	Modeling *business transactions*	UMM	3	4	2	2	2	4	1	3	1	2	1	2
T22	Modeling *business collaboration use cases*	UMM	1	3	2	2	2	2	1	2	1	1	-	-
T23	Modeling *business collaborations*	UMM	2	4	1	2	2	4	1	3	1	2	-	-

Figure 8.2 *Scenario Attribute Preference Table*

sistencies or errors (e.g., in the scenario path), it is hard to refactor to a stable version that is able to generate the desired profitability sheet.

Vertical interpretation of the results

The arrow in Figure 8.2 denotes, that there is also a vertical interpretation of the results. For instance, task T1 - T16 address rather economic issues and, hence, do not require any technical background. In general, within the evaluation table, we can spot the trend, that the business expert's learning curve is rated as rather user-friendly in regard to business models and rather user-unfriendly when it goes down to business process modeling (c.f. task T19 and T21).

Worksheets positively influence the guidance attribute

Another interesting trend is covered by the guidance attribute. All tasks that were guided by worksheets (as we proposed in Chapter 7) were easily manageable by both users. For instance, task T1 (modeling the value hierarchy) is guided by the *consumer need worksheet* and valuated with 1. In contrary, task T4 (modeling of the scenario path) was not supported by a specific worksheet and apparently effected difficulties during the modeling process.

Finally, the pattern attribute is used to investigate the methodology for the support of an automatic generation of artifacts that are delivered by recurring tasks. In our approach, we use patterns within different parts of the modeling methodology. For instance, the gener-

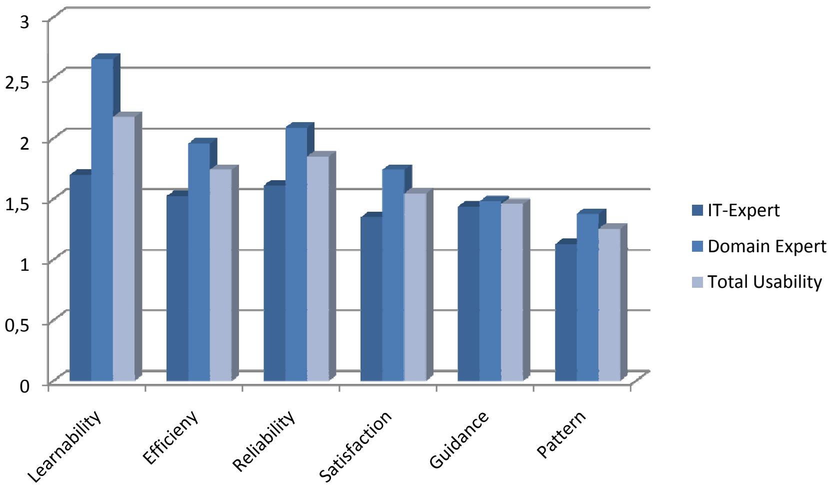


Figure 8.3 Summary of the scenario-based usability evaluation

ation of the economic exchange pattern in task 8, which always consists of an REA constellation (resource - event - agent). Another example is provided by our worksheet-driven approach where we generate UMM business transaction patterns out of worksheet descriptions (see task T21). The dashes in the last two columns of Figure 8.2 denote that not all of the tasks are required to use the advantages of patterns. Thus, these tasks were not validated against this attribute.

Figure 8.3 depicts the average rating of the different usability attributes. The diagram compares the valuation of the IT expert and the domain expert. Furthermore, it shows the total average usability of our approach. As we can see, the domain expert faces more problems than the IT expert in regard to understanding and learning the methodology. However, both users are on almost the same rating-level when the development of artifacts is supported by worksheets (guidance) or semi-automatic generation of modeling artifacts (pattern). As a consequence, we can say that the support by worksheets is an enhancement for an integration of the domain expert into the requirements elicitation phase toward the development of B2B processes.

Interviews and Open Discussions

Verifying a first support of the hypotheses

At the end of our evaluation we invited all participants to discuss the pros and cons of our approach. The overall task was to verify a first support of the hypotheses which were proposed in the beginning of the development process:

1. Using our business modeling approach for designing B2B processes helps the business analyst i) to design business processes from an economic point of view to ensure economic sustainability, ii) to semi-automatically generate process artifacts from business domain knowledge, and iii) to quickly adapt the B2B processes to changing requirements without the need to change the overall architecture.

2. A formalization of our approach improves the usability for the development of B2B processes i) by the definition of a unified process based on phases and iterations leading to a formalized and unambiguous requirements specification, ii) by the specification of well-defined transformation rules between the different methodologies, and iii) by the definition of worksheets for capturing and interlinking the domain knowledge.

Support of the first hypothesis

The two hypotheses addressed different stakeholders at this round table. The first one, which deals with the economic part of building partner networks, was rather discussed by the representatives of the newspaper publisher. This is due to the fact that the newspaper publisher is not satisfied with the "as-is" state of the company in regard to the development and introduction of new B2B systems. In fact, there is no connection between economic as well as strategic decisions made by the management and the implementers of the new solution. The discussion ended up into the numeration of several projects, where the missing connection between management and IT department led into late-design breakages. Furthermore, the stakeholders of the newspaper publisher agreed, that the integration of our approach would not only improve the vertical communication between the management level and the IT level, but also the horizontal communication between different departments. For instance, the controlling department faces an ease in their daily work, as well as other IT departments can benefit from the reusability of already developed modeling artifacts. Finally, the consortium agreed that the first hypothesis is supported by the results of our approach.

Support of the second hypothesis

The second hypothesis addresses issues that were rather interesting for the business analysts of the consulting companies. Most of the companies which are consulted by the business analysts lack of a clearly specified business process landscape (as well as the newspaper publisher). In order to introduce inter-organizational behavior into those companies, a formalized and unambiguous requirements specification by means of business processes is inevitable. Otherwise, B2B interaction with other companies is most unlikely if the external interfaces are not specified. Our approach brings major benefit to the development of B2B processes - especially by means of the UN/CEFACT's Modeling Methodology. It followed, that the consortium confirmed the support of the second hypothesis by the results of the approach.

Critics about the methodology address artifacts on the e^3value layer

However, there were also some critics about our developed methodology. Although the critics address general problems on making budget forecasts, we need to state these issues anyhow. In summary, the newspaper publisher sees two risks when integrating our methodology into the company. Both concerns address the business modeling level of our approach. Modeling a partner network and proofing the economic sustainability is a nice feature of the approach and fosters the communication between management and IT. However, they fear, that project leaders may tend to tamper the numbers of the valuation, to "make them fit" for getting their project approved by the managers. This would result in a high danger of inconsistencies between the business modeling layer and the business process modeling layer. A second risk that is raised by the newspaper publisher addresses a similar problem. If the operating department proposes economic forecasts by means of the e^{3}value profitability sheet, the management level is forced to meet those predictions. However, various projects in the past showed that it is difficult to follow the simulated economic requirements (e.g. due to a market crisis). As a matter of fact, companies do not always decline their confirmed budget by restricting the project goals, but (unfortunately) decline their number of employees.

8.4 Final assessment

In this chapter, we demonstrated a first evaluation of our approach. The methodological course of action for conducting the evaluation was threefold: first, we evaluated the approach from a scientific perspective in regard to the design of the methodology; second, we defined scenarios and tested the usability by applying the approach to a demonstrating real-world use case; third, we initiated a discussion with all participants about the pros and cons of the methodology. The reviewers participating in the evaluation were from the industry sector and brought in their knowledge and experience from similar projects. The results can be summarized as an approval from all participants in regard to the design science, the usability and the relevance for various problem domains. Finally, we also got some critical feedback in regard to the industrial practicability of predictions about the economic sustainability of B2B systems. However, the overall goal of the first evaluation was reached: i) to prove, whether the hypotheses made in the beginning of the project are supported by the results, and ii) to get valuable feedback from the industry partners to deliver further improvements of our approach.

9 Conclusion and open research issues

In this thesis we demonstrated a methodological approach for requirements management of B2B processes. Conventional approaches disregard the economic drivers of an inter-organizational IT system under development. The approach aims at combining the business model perspective with the business process model perspective. Thereby, we incorporated three well-established modeling methodologies in order to proof the economic sustainability by business models and the sequence of interactions between the business partners by business process models. Within this thesis we have addressed several gaps in current research which we summarize in the following.

A process-based requirements engineering approach

First, we have introduced a new value based requirements engineering solution by our general approach. The presented, process-based approach helps to overcome a set of limitations of classical requirements engineering approaches. We have introduced the six phases with their clearly specified iterations and have shown the application of the different phases using the customer acquisition example from the print media domain.

Combining three prominent modeling methodologies

However, the general approach is a rather light-weight methodology using fundamental modeling techniques and mostly word- processing tools. Although it delivers rudiments of business models, economic drivers for B2B systems can be captured more effectively by well-enhanced business modeling ontologies. Thus, we used the general approach as a staring point to incorporate three-well established modeling ontologies in the area of B2B - e^3value, REA, and UMM. The first two methodologies deliver business models and the latter one is already a popular modeling standard in the area of B2B maintained by UN/CEFACT [188]. Also, REA is on the way to become a UN/CEFACT standard soon, since there is already a draft version available of the so-called REA Specialization Module for UMM [189].

Findings of these thesis were contributed to the standardization efforts of UN/CEFACT

Another contribution of this thesis is the improvement of both UN/CEFACT methodologies - REA and UMM. In regard to REA, we discovered some limitations toward the adaptability of modeling multi-party collaborations. Furthermore, we provide preliminary concepts on a UML profile for REA as it is currently proposed in the draft version of the UN/CEFACT's REA Specification Module for UMM [189]. In terms of UMM, we migrated the UML profile to the recent UML version, which was strongly requested by stakeholders. Thereby, when working on the UMM we discovered some flaws and also several potential advancements, which we framed based on our findings. The result of this work, has flown back into the standardization efforts of UN/CEFACT resulting in version 2.0 of UMM, which has currently the status of a draft for implementation verification [193].

Interlinking e³value, REA and UMM

In this thesis we did not only show how the three methodologies are tailored for the use of managing B2B requirements. A key contribution is the definition of conceptual mapping rules between e^{3}value, REA and UMM. By applying these mapping rules, the business analyst will not run into inconsistencies between the different methodologies. We specify, which artifact delivered by a certain methodology maps to which artifact of the successive methodology. In order to support a Model Driven Engineering (MDE) we formalized these conceptual mapping rules by the ATLAS Transformation Language (ATL). Thereby, we defined transformation rules to initialize a target model from the information already captured by a source model.

Worksheet-based requirements elicitation and the generation of B2B artifacts

The final contribution of the thesis was the worksheet-driven requirements engineering approach, which delivers three major benefits to the requirements elicitation for B2B processes: First, it avoids inconsistencies between worksheets capturing the business experts' knowledge and the modeling artifacts. Second, our approach is characterized by its flexibility to adapt to specific needs in capturing business requirements. This is achieved by our XML-based Worksheet Definition Language (WDL), which can be used within any tool environment to define the structure of worksheets. Third, an interactive worksheet editor supports the semi-automatic generation of modeling artifacts.

First evaluation of the approach

At the beginning of this thesis, we were confident that our solution results in a better overall quality for the requirements management of B2B processes. Thus, we defined two hypotheses, which should be validated in the final development phase of this approach. A first evaluation, which was conducted with industry partners, showed that the methodology provides valuable improvements on the requirements engineering in the print media domain. Furthermore, two business analysts, who were also invited to participate in the evaluation, were confident that the approach can be easily applied to other business domains as well.

Open research issues and critical reflections

This thesis provides contributions to the field of requirements engineering for B2B processes. However, there are still some open research issues, which are subject to further research.

i) Extension of the general approach

As outlined in Chapter 4, the general requirements engineering approach concentrates on the process modeling phases of the business process lifecycle. Future work will concentrate on the extension of this approach in order to allow concrete service bindings to be added to the different activities and sub-activities - an activity typically performed in the process construction phase (cf. Figure 4.3). This approach will require the introduction of a new layer after the micro-modeling phase and an extension of the simulation/validation phase. If the general approach can be successfully used in the process construction phase as well, an easier integration of the developed requirements artifacts into the process execution phase would be possible.

ii) Additional REA concepts as input for UMM

In regard to our "top-down" approach from business models to business process models, there are still some open research issues as well. REA delivers the requirements from an economical point of view and serves as input for modeling a UMM compliant business process model. In our mapping we propose only a transformation from REA artifacts to UMM artifacts that are relevant in UMM's business requirements view. However, the attributes of REA's economic commitments capture significant information that may be relevant for further phases in UMM (e.g. the so-called "quality of service" parameters in the business choreography view). Furthermore, as a critical reflection we need to admit, that we did not consider REA's state machine driven approach [189]. This concept would help us to gather requirements for generating UMM artifacts, which are also used in later modeling steps (e.g. to generate concrete business transactions as part of UMM's business choreography view). Finally, REA's economic resources contain information which can be used to design the information envelopes that are being exchanged in UMM's business information view. Integrating those concepts into our approach is a major challenge for future work.

iii) Recommend business processes based on business models

One of the key contributions of this thesis is the integration of business models into the requirements engineering for inter-organizational processes. Thereby, we only provide conceptual rules on how to specify the fundamentals of the business process perspective by considering business modeling artifacts. As a consequence, it is the modeler's task to refine the business processes according to the business models manually. Thus, a potential milestone of future work could be the recommendation of business process models based on business models.

iv) Tool-support

Our approach incorporates conceptual mapping rules between the different modeling methodologies. Furthermore, the worksheet-driven approach provides guidelines on how to elicit the requirements by form-based documents. As a prototypical implementation, the UMM Add-In demonstrated the interactive integration of worksheets at the business process level by means of UMM. In fact, the approach lacks of a comprehensive tool that supports both - a modeling environment for all the different modeling methodologies and a wizard-driven requirements elicitation mechanism. Preliminary attempts to overcome those limitations are given in [124] by a Domain Specific Language (DSL).

v) Import functionality of already captured requirements and the integration of a domain vocabulary/ontology

Finally, in regard to our worksheet-driven approach, one may criticize the ability of a round-trip engineering. In certain cases, business domain experts have already captured several requirements before they start using our business model driven requirements engineering approach. In most cases these requirements are stored using word processing or spread sheet files (e.g. Microsoft Excel). In order to allow for an integration of this information into the worksheet editor, an import interface for Office Open XML [35] and Open Document Format [130] is planned for the future. Furthermore, it is likely that companies are already using certain vocabularies for specifying all kinds of requirements. An integration of such ontologies (e.g. by means of OWL) into our worksheet-driven approach would not only improve the usability of the worksheet editor, but also ensures to avoid inconsistencies in regard to the nomenclature of worksheet data.

Apart from these unresolved issues, we are confident that our approach eases the requirements engineering of B2B systems. Following the results of the first evaluation, we are confident that the "blank sheet of paper" phenomenon will not appear, if business analysts follow our proposed modeling guidelines.

List of Figures

Nomenclature

ATL	ATLAS Transformation Language
BCV	Business Choreography View
BIV	Business Information View
BMO	Business Modeling Ontology
BOV	Business Operational View
BRV	Business Requirements View
CC	Core Component
CCTS	Core Components Technical Specification
CRM	Customer Relationship Management
DSL	Domain Specific Language
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
EMF	Eclipse Modeling Framework
ERP	Electronic Resource Planning
FSV	Functional Service View
MDA	Model Driven Architecture
MDE	Model Driven Engineering
MOF	Meta-Object Facility
OCL	Object Constraint Language
PVC	Porter's Value Chain
RE	Requirements Engineering
REA	Resource Event Agent
RUP	Rational Unified Process
SOA	Service-oriented Architectures
UML	Unified Modeling Language
UMM	UN/CEFACT's Modeling Methodology
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
UN/ECE	United Nations Economic Commission for Europe
UN/TMG	UN/CEFACT's Techniques and Methodologies Group
UPCC	UML Profile for Core Components
WDL	Worksheet Definition Language
XMI	XML Metadata Interchange

Bibliography

[1] *BSopt - Business Sematics on Top of Process Technologies*, April 2010. http://www.bsopt.at.

[2] *SUPER - Semantics Utilized for Process Management within and between Enterprises*, March 2010. http://www.ip-super.org.

[3] Birger Andersson, Maria Bergholtz, Ananda Edirisuriya, Tharaka Ilayperuma, Paul Johannesson, Jaap Gordijn, Bertrand Grégoire, Michael Schmitt, Eric Dubois, Sven Abels, Axel Hahn, Benkt Wangler, and Hans Weigand. *Towards a Reference Ontology for Business Models*. In *Proc. of the 25th Int. Conf. on Conceptual Modeling*. Springer LNCS, 2006.

[4] Birger Andersson, Maria Bergholtz, Bertrand Grégoire, Paul Johannesson, Michael Schmitt, and Jelena Zdravkovic. *From Business to Process Models - a Chaining Methodology*. In *BUSITAL*, 2006.

[5] Daniel Ang, Lai-Huat Lim, and Hock-Chuan Chan. *Collaborative requirements engineering: an overview and a proposed integrated model*. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences, January 6-9, Hawaii, USA*, volume 5, pages 355–364, 1998.

[6] Jay April, Marco Better, Fred Glover, James Kelly, and Manuel Laguna. *Enhancing business process management with simulation optimization*. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 642–649. Winter Simulation Conference, 2006.

[7] Tomoyuki Arao, Eiji Goto, and Tomoko Nagata. *Business Process Oriented Requirements Engineering Process*. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE 2005), August 29 - September 2, Paris, France*, pages 395–402, 2005.

[8] Atlas Group. *ATL User Manual*, version 07. edition, http://wiki.eclipse.org/ATL/User_Guide, February 2006.

[9] ATLAS Group, INRIA, and LINA. *ATL (Atlas Transformation Language)*, http://www.eclipse.org/m2m/atl/doc/, May 2010.

[10] Muhammad Ali Babar, Liming Zhu, and D. Ross Jeffery. *A Framework for Classifying and Comparing Software Architecture Evaluation Methods*. In *Australian Software Engineering Conference*, pages 309–319, 2004.

[11] Amit Basu and Robert W. Blanning. *A graph-theoretic approach to analyzing knowledge bases containing rules, models and data*. *Annals of Operations Research*, 75:3–23, 1997.

[12] PerOlof Bengtsson and Jan Bosch. *Scenario-Based Software Architecture Reengineering*. In *ICSR '98: Proceedings of the 5th International Conference on Software Reuse*, page 308, Washington, DC, USA, 1998. IEEE Computer Society.

[13] Tim Berners-Lee and James Hendler. *Scientific publishing on the semantic web*. *Nature*, 410:1023–1024, 2001.

[14] P. Bertolazzi, C. Krusich, and M. Missikoff. *An Approach to the Definition of a Core Enterprise Ontology: CEO*, 2001.

[15] Barry W. Boehm, Paul Grünbacher, and Robert O. Briggs. *EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation*. *23rd International Conference on Software Engineering (ICSE'01)*, pages 720–721, 2001.

[16] Germinal Boloix and Pierre N. Robillard. *A Software System Evaluation Framework*. *IEEE Computer*, 28(12):17–26, 1995.

[17] Matthias Born, Christian Drumm, Ivan Markovic, and Ingo Weber. *SUPER - Raising Business Process Management Back to the Business Level*. *ERCIM News*, 70:43–44, Jul 2007.

[18] Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* Enschede, The Netherlands, Doctoral Thesis of the University of Tweenty, 1997.

[19] Kevin Brennan, editor. *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis, 2009.

[20] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. *Tropos: An Agent-Oriented Software Development Methodology*. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[21] R. J. A. Buhr. *Use case maps as architectural entities for complex systems*. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, December 1998.

[22] J.F. Burg. *Linguistic Instruments in Requirements Engineering*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1996.

[23] John S. Chandler. *A Multiple Criteria Approach for Evaluating Information Systems*. *MIS Quarterly*, 6:61–74, 1982.

[24] Peter Pin-Shan Chen. *The entity-relationship model—toward a unified view of data*. *ACM Trans. Database Syst.*, 1(1):9–36, 1976.

[25] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*. Kluwer Academic Press, Inc., 2000.

[26] Valeria de Castro, Esperanza Marcos, and Marcos López Sanz. *Service Composition Modeling: A Case Study*. In *ENC*, pages 101–108, 2006.

[27] Valeria de Castro, Juan M. Vara, Elisa Herrmann, and Esperanza Marcos. *A Model Driven Approach for the Alignment of Business and Information Systems Models*. In *ENC*, pages 33–43, 2008.

[28] Peter J. Denning and Robert Dunham. *Innovation as language action*. *Commun. ACM*, 49(5):47–52, 2006.

[29] Peter J. Denning and Peter A. Freeman. *The profession of IT - Computing's paradigm*. *Commun. ACM*, 52(12):28–30, 2009.

[30] Jan L. G. Dietz. *DEMO: Towards a discipline of organisation engineering*. *European Journal of Operational Research*, 128(2):351–363, January 2001.

[31] Jan L. G. Dietz and Hans B. F. Mulder. *Organizational Transformation Requires Constructional Knowledge of Business Systems*. In *HICSS (5)*, pages 365–375, 1998.

[32] Jan L.G. Dietz. *Enterprise ontology: theory and methodology*. Springer, 2006.

[33] Tharam S. Dillon, Chen Wu, and Elizabeth Chang. *Reference Architectural Styles for Service-Oriented Computing*. In *Network and Parallel Computing*, volume 4672/2008 of *Lecture Notes in Computer Science*, pages 543–555. Springer Berlin / Heidelberg, 2008.

[34] Jürgen Dorn, Christoph Grün, Hannes Werthner, and Marco Zapletal. *From business to software: a B2B survey*. *Information Systems and E-Business Management*, 7(2):123–142, March 2009.

[35] ECMA. *Office Open XML File Format*. ECMA International, Geneva, Switzerland, http://www.ecma-international.org/publications/standards/Ecma-376.htm, December 2008.

[36] Dieter Fensel. *Ontologies: a silver bullet for knowledge management and electronic commerce*. Springer-Verlag New York, Inc., 2001.

[37] Dieter Fensel, Stefan Decker, Michael Erdmann, and Rudi Studer. *Ontobroker: The Very High Idea*. In *Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference*, pages 131–135. AAAI Press, 1998.

[38] Eelke Folmer, Jilles van Gurp, and Jan Bosch. *Scenario-based Assessment of Software Architecture Usability*. In *ICSE Workshop on SE-HCI*, pages 61–68, 2003.

[39] Erwin Folmer and Joris Bastiaans. *Methods for Design of Semantic Message-Based B2B Interaction Standards*. In *Enterprise Interoperability III*, pages 183–196. Springer London, 2008.

[40] Mark S. Fox and Michael Gruninger. *Enterprise Modeling*. *AI Magazine*, 19(3):109–121, 1998.

[41] Frederik Gailly and Geert Poels. *Towards Ontology-driven Information Systems: Redesign and Formalization of the REA Ontology*. Working Papers of Faculty of Economics and Business Administration, Ghent University, Belgium 07/445, Ghent University, Faculty of Economics and Business Administration, January 2007.

[42] Frederik Gailly and Geert Poels. *Using the REA Ontology to Create Interoperability between E-Collaboration Modeling Standards*. In *Proceedings of the CAiSE'09*, 2009.

[43] Tracy Gardner. *UML Modelling of Automated Business Processes with a Mapping to BPEL4WS*. In *1st European Workshop on Object Orientation and Web Services (EOOWS'03)*. Springer, 2003.

[44] Guido L. Geerts and William E. McCarthy. *An Accounting Object Infrastructure for Knowledge-Based Enterprise Models*. *IEEE Intelligent Systems*, 14(4):89–94, 1999.

[45] Guido L. Geerts and William E. McCarthy. *An ontological analysis of the economic primitives of the extended-REA enterprise information architecture*. *International Journal of Accounting Information Systems*, 3(1):1 – 16, 2002.

[46] Jaap Gordijn, , Alexander Osterwalder, and Yves Pigneur. *Comparing two Business Model Ontologies for Designing e-Business Models and Value Constellations*. In *Proceedings of the 18th BLED conference (e-Integration in Action)*. University of Maribor, 2005.

[47] Jaap Gordijn. *The e3-value toolset*, May 2010. http://www.e3value.com.

[48] Jaap Gordijn and Hans Akkermans. *Designing and evaluating e-business models*. *IEEE Intelligent Systems*, 16(4):11–17, Jul–Aug 2001.

[49] Jaap Gordijn and Hans Akkermans. *Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas*. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[50] Jaap Gordijn, Hans Akkermans, and Hans van Vliet. *Business Modelling Is Not Process Modelling*. In *ER '00: Proceedings of the Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling*, pages 40–51, London, UK, 2000. Springer-Verlag.

[51] Jaap Gordijn, Sybren Kinderen, de, and Roel Wieringa. *Value-Driven Service Matching*. In *16th IEEE International Requirements Engineering Conference*, pages 67–70. EEE Computer Society Press, 2008.

[52] Jaap Gordijn, Pascal van Eck, and Roel Wieringa. *Requirements Engineering Techniques for e-Services*. Cooperative Information Systems series. The MIT Press, Cambridge, 2009.

[53] Ellen Gottesdiener. *Requirements by collaboration: getting it right the first time*. *Software, IEEE*, 20(2):52–55, 2003.

[54] The Open Group. *The Open Group Architecture Framework*. Van Haren Publishing, 2006. TOGAF 8.1.1.

[55] Thomas R. Gruber. *A Translation Approach to Portable Ontology Specifications*. *Knowledge Acquisition*, 6(2):199–221, 1993.

[56] Paul Gruenbacher. *Collaborative Requirements Negotiation with EasyWinWin*. *Database and Expert Systems Applications, International Workshop on*, 0:954, 2000.

[57] Paul Gruenbacher. *Integrating groupware and CASE capabilities for improving stakeholder involvement in requirements engineering*. In *Proceedings of the 26th Euromicro Conference, September 5-7, Maastricht, Netherlands*, volume 2, pages 232–239, 2000.

[58] Nicola Guarino. *Understanding, building and using ontologies*. *Int. J. Hum.-Comput. Stud.*, 46(2-3):293–310, 1997.

[59] Michael Hammer and James Champy. *Business Reengineering*. Heyne, Campus, 2. edition, 1999.

[60] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. *Design Science in Information Systems Research*. *MIS Quarterly*, 28(1), 2004.

[61] Vlatka Hlupic and Stewart Robinson. *Business process modelling and analysis using discrete-event simulation*. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 1363–1370, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[62] Birgit Hofreiter, Christian Huemer, and Ja-Hee Kim. *Choreography of ebXML business collaborations*. *Information Systems and E-Business Management*, 4(3):221–243, July 2006.

[63] Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *UMM Add-In: A UML Extension for UN/CEFACT's Modeling Methodology*. In *Proceedings of the European Conference on Model Driven Architecture (ECMDA'06), July 10-13, Bilbao, Spain*, pages 618–619. Springer, 2006.

[64] Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *UN/CEFACT's Modeling Methodology (UMM): A UML Profile for B2B e-Commerce*. In *Proceedings of the 2nd International Workshop on Best Practices of UML (ER BP-UML'06), November 6-9, Tucson, AZ, USA*, pages 19–31. IEEE, 2006.

[65] Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Deriving executable BPEL from UMM Business Transactions*. In *Proceedings of the IEEE International Conference on Services Computing (SCC2007), July 9-13, Salt Lake City, UT, USA*, pages 178–186. IEEE, 2007.

[66] Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *The UMM Add-In - Demo*. In *Proceedings of the International Conference on Services Oriented Computing (ICSOC2007), September 17-20, Vienna, Austria*, pages 618–619. Springer, 2007.

[67] Karen Holtzblatt and Hugh R. Beyer. *Requirements gathering: the human factor*, volume 38. ACM, New York, NY, USA, 1995.

[68] Pavel Hruby. *Model-Driven Design Using Business Patterns*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[69] Christian Huemer, Philipp Liegl, Thomas Motal, Rainer Schuster, and Marco Zapletal. *The Development Process of the UN/CEFACT Modeling Methodology*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08), August 19-22, Innsbruck, Austria*, pages 1–10. ACM, 2008.

[70] Christian Huemer, Philipp Liegl, Rainer Schuster, Hannes Werthner, and Marco Zapletal. *Inter-organizational Systems: From Business Values over Business Processes to Deployment*. In *Proceedings of the 2nd International IEEE Conference on Digital Ecosystems and Technologies (DEST2008), February 26-29, Phitsanulok, Thailand*, pages 294–299. IEEE, 2008.

[71] Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Modeling Business Entity State Centric Choreographies*. In *Proceedings of the IEEE Joint Conference on E-Commerce Technology (CEC07), July 23-26, Tokyo, Japan*, pages 393–400. IEEE, 2007.

[72] Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Worksheet Driven UMM Modeling of B2B Services*. In *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2007), October 24-26, Hongkong, China*, pages 30–38. IEEE, 2007.

[73] Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *A 3-level e-Business Registry Meta Model*. In *Proceedings of the IEEE International Conference on Services Computing (SCC2008), July 8-11, Honolulu, HI, USA*, pages 451–450. IEEE, 2008.

[74] Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *B2B Services: Worksheet-Driven Development of Modeling Artifacts and Code*. *The Computer Journal*, 52(8):1006–1026, 2009.

[75] Christian Huemer, Philipp Liegl, Rainer Schuster, Marco Zapletal, and Birgit Hofreiter. *Service-Oriented Enterprise Modeling and Analysis*. In *Handbook of Enterprise Integration*, pages 307–322. Auerbach Publications, 2009.

[76] IBM. *WebSphere*, `http://www-01.ibm.com/software/websphere/`, May 2010.

[77] ISO. *Information technology - Business Operational View - Part 1: Operational aspects of Open-edi for implementation*, ISO/IEC 2002, ISO 15944-1, 2002.

[78] ISO. *Open-edi Reference Model*, ISO/IEC JTC 1/SC30 ISO Standard 14662, Second Edition, `http://standards.iso.org/ittf/`

PubliclyAvailableStandards/c037354_ISO_IEC_14662_2004(E).zip, 2004.

[79] ISO. *Information technology - Business Operational View - Part 4: Business transaction scenarios*, ISO/IEC 2007, ISO 15944-4, 2007.

[80] Marina Jirotka and Joseph A. Goguen, editors. *Requirements engineering: social and technical issues*. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[81] Frédéric Jouault and Ivan Kurtev. *Transforming Models with ATL*. In *MoDELS Satellite Events*, pages 128–138, 2005.

[82] Jae-Yoon Jung, Hoontae Kim, and Suk-Ho Kang. *Business process choreography for B2B collaboration*. *IEEE Internet Computing*, 8(1):37–45, January 2004.

[83] R. Kalakota and A. B. Whinston. *Frontiers of Electronic Commerce*. Addison-Wesley, 1996.

[84] Rick Kazman, Len Bass, Mike Webb, and Gregory Abowd. *SAAM: a method for analyzing the properties of software architectures*. In *ICSE '94: Proceedings of the 16th international conference on Software engineering*, pages 81–90, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[85] Haim Kilov and Ian Simmonds. *Eliciting Requirements: Beyond the Blank Sheet of Paper*. In *RE '97: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, page 141.2, Washington, DC, USA, 1997. IEEE Computer Society.

[86] HyoungDo Kim. *Conceptual Modeling and Specification Generation for B2B Business Processes based on ebXML*. *ACM SIGMOD Record*, 31(1):137–145, 2002.

[87] Sybren Kinderen, de and Jaap Gordijn. *E3service : An ontological approach for deriving multi-supplier IT-service bundles from consumer needs*. In Ralph H. Sprague, editor, *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 318. IEEE CS, 2008. Best paper nomination.

[88] Birgit Korherr and Beate List. *Extending the UML 2 Activity Diagram with Business Process Goals and Performance Measures and the Mapping to BPEL*. In *Proceedings of the ER-Conference on Conceptual Modeling (Workshop BP-UML'06)*, volume 4231 of *LNCS*, pages 7–18. Springer, November 2006.

[89] Birgit Korherr and Beate List. *A UML 2 Profile for Event Driven Process Chains*. In *Proceedings of the 1st IFIP International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS 2006)*. Springer, 2006.

[90] Gerhard Kramler, Elisabeth Kapsammer, Gerti Kappel, and Werner Retschitzegger. *Towards Using UML 2 for Modelling Web Service Collaboration Protocols*. In *Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA'05)*, pages 227–238. Springer, February 2006.

[91] Per Kroll and Philippe Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley Professional, 2003.

[92] Kuldeep Kumar. *Post implementation evaluation of computer-based information systems: current practices*. *Commun. ACM*, 33(2):203–212, 1990.

[93] Brian Lawrence, Karl Wiegers, and Christof Ebert. *The top risk of requirements engineering*. *Software, IEEE*, 18(6):62–63, 2001.

[94] Allen S. Lee. *Remarks from MIS Quarterly Editor - Inaugural Editor's Comments*. *MIS Quarterly*, 23(1), 1999.

[95] Ronald M. Lee. *Documentary Petri Nets: A Modeling Representation for Electronic Trade Procedures*. In *Business Process Management: Models, Techniques, and Empirical Studies*, pages 359 – 375. Springer, 2000.

[96] Peter Lendermann, Nirupam Julka, Lai Peng Chan, and Boon Ping Gan. *The process of process reengineering: integration of discrete event simulation models with framework-based business applications*. In *WSC '03: Proceedings of the 35th conference on Winter simulation*, pages 1797–1804. Winter Simulation Conference, 2003.

[97] Kirsten Lenz and Andreas Oberweis. *Inter-organizational Business Process Management with XML Nets*. In *Petri Net Technology for Communication-Based Systems*, volume 2472, pages 243–263. Springer, 2003.

[98] Philipp Liegl. *Business Documents for Inter-Organizational Business Processes*. PhD thesis, Institute of Software Technology and Interactive Systems, 2010.

[99] Philipp Liegl, Robert Mosser, Rainer Schuster, and Marco Zapletal. *Modeling e-Government processes with UMM*. In *Proceedings of the VIP Symposia on Internet related research (VIPSI2007), June 7-10, Opatija, Croatia*, pages 1–11, 2007.

[100] Philipp Liegl, Thomas Motal, and Rainer Schuster. *An Add-In for UN/CEFACT's Modeling Methodology 2.0*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08), August 19-22, Innsbruck, Austria*, pages 1–2, 2008.

[101] Philipp Liegl, Rainer Schuster, Robert Mosser, and Marco Zapletal. *Modeling e-Government processes with UMM*. *Informatica Journal - An International Journal of Computing and Informatics*, 31(4):407–417, 2007.

[102] Philipp Liegl, Rainer Schuster, and Marco Zapletal. *A UML Profile and Add-In for UN/CEFACT's Modeling Methodology*. Master's thesis, University of Vienna, 2006.

[103] Philipp Liegl, Rainer Schuster, Marco Zapletal, Christian Huemer, Hannes Werthner, Michael Aigner, Martin Bernauer, Bjoern Klinger, Michaela Mayr, Ramin Mizani, and Martin Windisch. *[vem:xi:] - A methodology for process based requirements engineering*. In *Proceedings of the 17th IEEE International Requirements Engineering Conference (RE09), August 31 - September 4, Atlanta, GA, USA*, pages 193–202. IEEE, 2009.

[104] F.-R. Lin, M.-C. Yang, and Y.-H. Pai. *A generic structure for Business Process Modeling*. *Business Process Management Journal*, 8(1):19–41, 2002.

[105] Jane Linder and Susan Cantrell. *Changing Business Models: Surveying the Landscape*. *accenture Institute for Strategic Change*, May 2000.

[106] Sea Ling and Seng Wai Loke. *Advanced Petri Nets for modelling mobile agent enabled interorganizational workflows*. In *Proc. Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, pages 245–252, 8–11 April 2002.

[107] Beate List and Birgit Korherr. *A UML 2 Profile for Business Process Modelling*. In *Proceedings of the International Conference on Conceptual Modeling (ER 2005), October 24-28, Klagenfurt, Austria*, pages 85–96. Springer, 2005.

[108] Beate List and Birgit Korherr. *An evaluation of conceptual business process modelling languages*. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1532–1539, New York, NY, USA, 2006. ACM.

[109] Amy Lo and Eric Yu. *From Business Models to Service-Oriented Design: A Reference Catalog Approach*. In *ER*, pages 87–101, 2007.

[110] Joan Magretta. *Why Business Models Matter*. *Harvard Business Review*, 5:86–92, May 2002.

[111] Neil A. M. Maiden. *CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements*. *Autom. Softw. Eng.*, 5(4):419–446, 1998.

[112] Ivan Markovic. *Advanced Querying and Reasoning on Business Process Models*. In *Business Information Systems, 11th International Conference, BIS 2008*, pages 189–200, 2008.

[113] Ivan Markovic and Mario Karrenbrock. *Semantic Web Service Discovery for Business Process Models*. In *WISE Workshops*, pages 272–283, 2007.

[114] Peter Vervest-Rene Wagenaar Martijn R Hoogeweegen, Wim J Teunissen. *Modular network design: Using information and communication technology to allocate production tasks in a virtual organization*. *Decision Sciences*, 30:1073–1103, 1999.

[115] Richard Mayer, Christopher Menzel, Michael Painter, Paula deWitte, Thomas Blinn, and Benjamin Perakath. *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*. AL-TR-1995-XXXX, Knowledge Based Systems, Inc., September 1995.

[116] Richard J. Mayer, Perakath C. Benjamin, Bruce Caraway, and M. K. Painter. *A Framework and a Suite of Methods for Business Process Reengineering*. *Business Process Reengineering: A Managerial Perspective*, pages 245–290, 1995.

[117] William McCarthy, Rainer Schuster, and Thomas Motal. *Modeling Multi-party Collaborations in e3-Value and REA: An Example and Some Preliminary Observations*. In *International Workshop on Value Modeling and Business Ontologies, VMBO'09, Stockholm - Feb 8-9, 2009*, 2009.

[118] William E. McCarthy. *The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment*. *The Accounting Review*, 57(3):554–578, July 1982.

[119] Sheila A. McIlraith and Honglei Zeng. *Semantic web services*. *IEEE intelligent systems*, 16(2):46–53, 2001.

[120] Jan Mendling, Gustaf Neumann, and Markus Nüttgens. *Yet Another Event-Driven Process Chain*. In *3rd International Conference on Business Process Management (BPM 2005)*, volume 3649, pages 428–433. Springer, 2005.

[121] Jan Mendling and Markus Nüttgens. *EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC)*. *Information Systems and E-Business Management*, 4(3):245–263, 2006.

[122] Michael Merz. *E-Commerce und E-Business*. Dpunkt.Verlag GmbH, 9 2001.

[123] D. Mishra, A. Mishra, and A. Yazici. *Successful requirement elicitation by combining requirement engineering techniques*. In *Proceedings of Applications of the First International Conference on Digital Information and Web Technologies, (ICADIWT 2008), August 4-6, Ostrava, Czech Republic*, pages 258–263, 2008.

[124] Thomas Motal, Marco Zapletal, and Hannes Werthner. *The Business Choreography Language (BCL) - A Domain-Specific Language for Global Choreographies*. In *SERVICES II*, pages 150–159, 2009.

[125] Tadao Murata. *Petri nets: Properties, analysis and applications*. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[126] William A. Neilson, Thomas Knott, and Paul Carhart, editors. *Third New International Dictionary of the English Language*. Merriam-Webster, 2002.

[127] Joerg Nitzsche and Barry Norton. *Ontology-based Data Mediation in BPEL*. In *Semantics4WS 2008 Workshop, Milan, Italy*, 2008.

[128] Bashar Nuseibeh and Steve Easterbrook. *Requirements engineering: a roadmap*. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, pages 35–46, New York, NY, USA, 2000. ACM.

[129] OASIS. *ebXML Business Process Specification Schema Technical Specification*, OASIS Standard, Version 2.0.4, `http://docs.oasis-open.org/ebxml-bp/2.0.4/ebxmlbp-v2.0.4-Spec-os-en.html`, December 2006.

[130] OASIS. *Open Document Format*. Organization for the Advancement of Structured Information Standards, United States, `http://www.oasis-open.org/committees/download.php/19274/OpenDocument-v1.0ed2-cs1.pdf`, July 2006.

[131] OASIS. *Web Services Business Process Execution Language Version 2.0 (WS-BPEL)*, `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html`, 2007.

[132] Ernst Oberortner, Uwe Zdun, and Schahram Dustdar. *Domain-Specific Languages for Service-Oriented Architectures: An Explorative Study*. In *Proceedings of the First European ServiceWave Conference (ServiceWave), December 10 - 13, Madrid, Spain*, pages 159–170. Springer, 2008.

[133] Object Management Group. *Business Process Modeling Notation 1.1 (BPMN)*, `http://www.bpmn.org/Documents/BPMN1-1Specification.pdf`, 2006.

[134] Object Management Group. *Meta Object Facility (MOF)*, `http://www.omg.org/mof/`, 2006.

[135] Object Management Group. *Unified Modeling Language: Superstructure 2.2 (UML)*, `http://www.omg.org/spec/UML/2.1.1/Superstructure/PDF/`, 2009.

[136] Object Management Group (OMG). *MDA Guide*, Version 1.01, `http://www.omg.org/docs/omg/03-06-01.pdf`, June 2003.

[137] Object Management Group (OMG). *Unified Modeling Language Specification*, Version 1.4.2, also available as ISO/IEC 19501, `http://www.omg.org/spec/UML/ISO/19501/PDF/`, April 2005.

[138] Object Management Group (OMG). *Business Process Modeling Notation Specification*, Version 1.2, `http://www.omg.org/spec/BPMN/1.2/PDF`, January 2009.

[139] OMG. *Object Constraint Language, Technical Specification V2.0*, `http://www.omg.org/spec/OCL/2.0/PDF`, May 2006.

[140] OMG. *Business Motivation Model (BMM) Specification, Technical Specification V1.0*, `http://www.omg.org/cgi-bin/apps/doc?dtc/07-08-03.pdf`, September 2007.

[141] OMG - Object Management Group. *XMI Mapping Specification*, version 2.1.1. edition, December 2007.

[142] Oracle. *Oracle SOA Suite 11g*, 2010.

[143] Tim O'Reilly. *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*, 2005. `http://oreilly.com/web2/archive/what-is-web-20.html`.

[144] Alexander Osterwalder. *The Business Model Ontology. A Proposition in a Design Science Approach*. Dissertation, University of Lausanne, 2004.

[145] Alexander Osterwalder and Yves Pigneur. *An e-Business Model Ontology for Modeling e-Business*. In *Proceedings of the 15th Bled Electronic Commerce Conference, June 17-19, Bled, Slovenia*, pages 1–12. EconWPA, 2002.

[146] Barbara Paech. *What Is a Requirements Engineer? IEEE Softw.*, 25(4):16–17, 2008.

[147] Michael P. Papazoglou and Pieter Ribbers. *e-Business: Organizational and Technical Foundations*. John Wiley and Sons, 2006.

[148] Mike P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. *Service-Oriented Computing: State of the Art and Research Challenges*. *IEEE Computer*, 40(11):38–45, 2007.

[149] Louis-François Pau and Peter H. M. Vervest. *Network-based Business Process Management: a Discussion on Embedding Business Logic in Communications Networks*. In *CAiSE Workshops (2)*, pages 176–183, 2004.

[150] Debra Paul and Donald Yeates, editors. *Business Analysis*. BCS, 2006.

[151] Chris Peltz. *Web Services Orchestration and Choreography*. *IEEE Computer*, 36(10):46–52, 2003.

[152] Magnus Penker and Hans-Erik Eriksson. *Business Modeling With UML: Business Patterns at Work*. Wiley, 2000.

[153] Ken Preiss Louis-Francois Pau Peter H.M. Vervest, Eric Heck. *Smart Business Networks*. Springer, 2004.

[154] Otto Petrovic, Christian Kittl, and Ryan D. Teksten. *Developing Business Models for eBusiness*. In *International Conference on Electronic Commerce*, 2001.

[155] Mario Pichler and Hildegard Rumetshofer. *Business Process-based Requirements Modeling and Management*. In *Proceedings of the First International workshop on Requirements Engineering Vizualization (REV06), September 11, Minneapolis, USA*, 2006.

[156] Yves Pigneur. *Internet-based Information Systems for e-Business*. Swiss Computer Science Conference (SCSC99), Lausanne, Oct 1999.

[157] Dubosson-Torbay Magali Pigneur Yves, Osterwalder Alexander. *eBusiness Model Design, Classification and Measurements*. *Thunderbird International Business Review*, 44(1):5–23, January 2002.

[158] Vincent Pijpers and Jaap Gordijn. *Bridging Business Value Models and Business Process Models in Aviation Value Webs via Possession Rights*. In *Proceedings of the 20th Annual Hawaii International Conference on System Sciences*, page cdrom. Computer Society Press, 2007.

[159] Klaus Pohl. *Requirements Engineering - Grundlagen, Prinzipien, Techniken*. dpunkt, Heidelberg, 2008.

[160] Michael E. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, 1998.

[161] Michael I. Posner, editor. *The Foundations of Cognitive Science*. MIT Press, 1993.

[162] Chun Quyang, Marlon Dumas, Arthur H.M. ter Hofstede, and Wil M.P. van der Aalst. *From BPMN Process Models to BPEL Web Services*. In *Proc. International Conference on Web Services ICWS '06*, pages 285–292, Sept. 2006.

[163] Michael Rappa. *Managing the Digital Enterprise - Business Models on the Web*. http://digitalenterprise.org/, 2001.

[164] Faribors Ronaghi. *A modeling method for integrated performance management*. In *Sixteenth International Workshop on Database and Expert Systems Applications, 2005. Proceedings.*, pages 972 – 976, aug. 2005.

[165] RosettaNet. *RosettaNet Implementation Framework: Core Specification*, V02.00.01, `http://www.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/rnif/`, December 2002.

[166] Nick Russell, Wil M.P. van der Aalst, Arthur H.M. ter Hofstede, and Petia Wohed. *On the suitability of UML 2.0 activity diagrams for business process modelling*. In *APCCM '06: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, pages 95–104, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[167] Omar El Sawy. *Redesigning Enterprise Processes for e-Business*. McGraw-Hill/Irwin, 2000.

[168] August-Wilhelm Scheer. *ARIS - Business Process Modeling*. Springer, 2000.

[169] Douglas C. Schmidt. *Guest Editor's Introduction: Model-Driven Engineering*. *IEEE Computer*, 39(2):25–31, 2006.

[170] Michael Schmitt and Bertrand Grégoire. *Risk Mitigation Instruments for Business Models and Process Models*. *Proc. REBNITA*, 5, 2005.

[171] Rainer Schuster, Philipp Liegl, Marco Zapletal, Hannes Werthner, Christian Huemer, Christoph Grün, Dieter Mayerhofer, and Thomas Motal. *E-Business and Semantic Technologies*. In *Handbook of Semantic Web Technologies*. Springer, 2010 (to appear).

[172] Rainer Schuster and Thomas Motal. *A Holistic Approach Towards a UML Profile for Business Modeling*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08) CEUR-WS, August 19-22, Innsbruck, Austria*, 2008.

[173] Rainer Schuster and Thomas Motal. *From e3-value to REA: Modeling Multi-party E-business Collaborations*. In *CEC '09: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, pages 202–208, Washington, DC, USA, 2009. IEEE Computer Society.

[174] Rainer Schuster, Thomas Motal, Christian Huemer, and Hannes Werthner. *From Economic Drivers to B2B Process Models: a Mapping from REA to UMM*. In *13th International Conference on Business Information Systems BIS2010, Berlin, May 3-5, 2010*, 2010.

[175] Marlo Seppanen and Saku Makinen. *Business model concepts: a review with case illustration*. In *IEEE International Engineering Management Conference*, volume 1, pages 376 – 380, 11-13, 2005.

[176] Mary Shaw. *Prospects for an Engineering Discipline of Software*. *IEEE Softw.*, 7(6):15–24, 1990.

[177] John Sherwood, Andrew Clark, and David Lynas. *Enterprise Security Architecture: A Business-Driven Approach*. cmp, 11 2005.

[178] Munindar P. Singh, Amit K. Chopra, and Nirmit Desai. *Commitment-Based Service-Oriented Architecture*. *Computer*, 42:72–79, 2009.

[179] Eva Söderström, Birger Andersson, Paul Johannesson, Erik Perjons, and Benkt Wangler. *Towards a Framework for Comparing Process Modelling Languages*. In *CAiSE '02: Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pages 600–611, London, UK, 2002. Springer-Verlag.

[180] John F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole, 2000.

[181] Sparx Systems. *Enterprise Architect*. `http://www.sparxsystems.com.au`, 2009.

[182] Dave Steinberg, Frank Budinsky, Marcelo Paternostro, and Ed Merks. *EMF: Eclipse Modeling Framework*. Addison-Wesley, 2009.

[183] Richard Stevens, Peter Brook, Ken Jackson, and Stuart Arnold. *Systems Engineering: coping with complexity*. Prentice Hall, 1998.

[184] Supply Chain Council. *Supply-Chain Operations Reference-model - SCOR*, Version 8.0, `http://www.supply-chain.org`, 2006.

[185] Don Tapscott, David Ticoll, and Alex Lowy. *The rise of business webs*. *Ubiquity*, 1(3):2, 2000.

[186] Sebastian Thöne, Ralph Depke, and Gregor Engels. *Process-Oriented, Flexible Composition of Web Services with UML*. In *Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Proceedings*, LNCS. Springer, 2002.

[187] Paul Timmers. *Business Models for Electronic Markets*. *Electronic Markets*, 8(2):3–8, July 1998.

[188] UN/CEFACT. *UN/CEFACT's Modeling Methodology (UMM), UMM Meta Model - Foundation Module*, Technical Specification V1.0, http://www.unece.org/cefact/umm/UMM_Foundation_Module.pdf, March 2006.

[189] UN/CEFACT. *REA Specialization Module for UN/CEFACT's Modeling Methodology (UMM)*, Public Draft V1.0, 2008.

[190] UN/CEFACT. *UML Profile for Core Components Technical Specification 1.0 (UPCC)*, `http://www.untmg.org/upcc/spec/1_0`, 2008.

[191] UN/CEFACT. *Core Components Technical Specification 3.0 (CCTS)*, `http://www.untmg.org/ccts/spec/3_0`, 2009.

[192] UN/CEFACT International Trade and Business Process Group (TBG14). *UN/CEFACT Common Business Process Catalog*, Version 1.0, Technical Specification, http://www.uncefactforum.org/TBG/TBG14/TBG14Deliverables.htm, September 2005.

[193] UN/CEFACT Technologies and Methodologies Group (TMG). *UN/CEFACT's Modeling Methodology (UMM), UMM Meta Model - Foundation Module*, Candidate for V2.0, Draft for Implementation Verification, http://www.untmg.org/umm/spec/foundation/2_0, March 2010.

[194] Robert Urwiler and Mark N. Frolick. *The IT Value Hierarchy: Using Maslow's Hierarchy of Needs as a Metaphor for Gauging the Maturity Level of Information Technology Use within Competitive Organizations*. *Information Systems Management*, 25(1):83–88, December 2008.

[195] Mike Uschold, Martin King, Stuart Moralee, and Yannis Zorgios. *The Enterprise Ontology*. *The Knowledge Engineering Review*, 13(1):31–89, 1998.

[196] Wil M. P. van der Aalst. *The Application of Petri Nets to Workflow Management*. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[197] Wil M. P. van der Aalst. *Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets*. *Systems Analysis - Modelling - Simulation*, 34(3):335–367, 1999.

[198] Wil M. P. van der Aalst. *Making Work Flow: On the Application of Petri Nets to Business Process Management*. In *Proceedings of the 23rd International Conference on Applications and Theory of Petri Nets (ICATPN 2002), June 24-30, Adelaide, Australia*, pages 1–22. Springer, 2002.

[199] W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske. *Business Process Management: A Survey*. In *Business Process Management*, volume 2678/2003 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2003.

[200] Eric van Heck and Peter Vervest. *Smart business networks: how the network wins*. *Communications of the ACM*, 50(6):28–37, 2007.

[201] Axel van Lamsweerde, Robert Darimont, and Emmanuel Letier. *Managing Conflicts in Goal-Driven Requirements Engineering*. *IEEE Trans. Software Eng.*, 24(11):908–926, 1998.

[202] André Vasconcelos, Artur Caetano, ao Neves, Jo Pedro Sinogas, Ricardo Mendes, and José Tribolet. *A Framework for Modeling Strategy, Business Processes and Information Systems*. In *EDOC '01: Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*. IEEE Computer Society, 2001.

[203] Ingo Weber, Ivan Markovic, and Christian Drumm. *A Conceptual Framework for Composition in Business Process Management*. In *BIS*, pages 54–66, 2007.

[204] Hans Weigand, Paul Johannesson, Birger Andersson, Maria Bergholtz, Ananda Edirisuriya, and Tharaka Ilayperuma. *On the notion of value object*. *LNCS*, 4001:321, 2006.

[205] Hannes Werthner and Michael Borovicka. *E-commerce und Semantic Web*. In *Semantic Web*, X.media.press, pages 307–319. Springer Berlin Heidelberg, 2006.

[206] Stephen A. White. *Using BPMN to Model a BPEL Process*. Technical report, IBM, 2005.

[207] Roel J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. John Wiley, 1996.

[208] World Wide Web Consortium. *Web Services Description Language 2.0 (WSDL)*, `http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/`, 2007.

[209] World Wide Web Consortium (W3C). *OWL Web Ontology Language*, W3C Recommendation, `http://www.w3.org/TR/owl-features/`, February 2004.

[210] World Wide Web Consortium (W3C). *Resource Description Framework (RDF)*, W3C Recommendation, `http://www.w3.org/RDF/`, February 2004.

[211] Eric S. K. Yu. *Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering*. In *RE*, pages 226–235, 1997.

[212] John Zachman. *A Framework for Information Systems Architecture*. *IBM Systems Journal*, 31(3):445–470, 1999.

[213] Marco Zapletal. *A UML-based Methodology for Model-Driven B2B Integration: From Business Values over Business Processes to Deployment Artifacts*. PhD thesis, Vienna University of Technology, 2009.

[214] Marco Zapletal, Philipp Liegl, and Rainer Schuster. *UN/CEFACT's Modeling Methodology (UMM) 1.0 - A Guide to UMM and the UMM Add-In*. VDM Verlag Dr. Müller, 2008.

[215] Pamela Zave. *Classification of research efforts in requirements engineering*. *ACM Comput. Surv.*, 29(4):315–321, 1997.

[216] Valerie Zeithaml, Mary J. Bitner, and Dwayne D. Gremler. *Services Marketing*. McGraw-Hill/Irwin, New York, NY, May 2005.

Acknowledgments

Die Verfassung dieser Arbeit wäre ohne die direkte und indirekte Unterstützung von vielen Personen nicht möglich gewesen.

Allen voran möchte ich mich bei meinen beiden Betreuern, Hannes Werthner und Christian Huemer, für die Zeit und Geduld, die sie in den letzten Jahren aufgebracht haben, bedanken. Beide waren nicht nur in sportlicher Hinsicht exzellente Trainingspartner am Tennis- bzw. Fußballplatz, sondern hatten auch stets ein offenes Ohr für meine forschungsrelevanten Anliegen und Fragen.

Ganz besonders möchte ich Philipp Liegl und Marco Zapletal danken, mit denen ich die meisten meiner Publikationen gemeinsam verfasst habe. Die vielen Projekte, die wir in der *BSopt-Suite* schon umgesetzt haben, haben unseren bunten Haufen zu einem richtig freundschaftlichen Team zusammengeschweißt.

Weiters möchte ich auch Thomas Motal danken, der mich im Bereich der Geschäftsmodellierung stets mit guten Ideen und wertvollem Publikations-Input unterstützt hat.

Auch dem Umfeld rund um die *BSopt-Suite* bestehend aus Birgit Dippelreiter, Christoph Grün, Dieter Mayrhofer, Michael Pöttler und Mario Topf gebührt mein Dank für die oftmals sehr lustigen Stunden im Büro. Ebenso danke an Gerti Kappel (für das Feedback zu so manchen Konferenz-Probevorträgen), Manuel Wimmer (für die ATL Unterstützung), Birgit Hofreiter (für interessante Diskussionen während der vielen UN/CEFACT Meetings) und Bill McCarthy (it's all about REA).

Besonderer Dank gebührt auch meiner Familie, die mich während des Studiums nicht nur ökonomisch und önologisch, sondern auch motivierend und emotional unterstützt hat.

Zum Abschluss möchte ich meiner Freundin Simone danken, die in den letzten Monaten meiner Dissertation viel Verständnis bewiesen hat.

Curriculum Vitae

Curriculum Vitae: Rainer Schuster

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme
Electronic Commerce Group
Favoritenstrasse 9-11/188
A-1040 Wien, Österreich

e-mail: schuster@ec.tuwien.ac.at
tel: +43/1/588 01/18861
fax: +43/1/588 01/18896
mobil:+43/699/17910230
web: `http://www.ec.tuwien.ac.at/~schuster`

Curriculum Vitae: Rainer Schuster

Akademischer Grad	Mag.rer.soc.oec	
Persönliche Daten	*Geburtsdatum / Geburtsort:* 23.10.1979, Eisenstadt *Familienstand:* ledig	
Ausbildung	**Technische Universität Wien** Doktoratsstudium Wirtschaftsinformatik • Titel der Dissertation: *Requirements Management for B2B Processes: a Worksheet Driven Approach from e3-Value and REA to UMM*	Oktober 2006 - Mai 2010
	Technische Universität Wien Diplomstudium Wirtschaftsinformatik • Titel der Diplomarbeit: *A UML Profile and Add-In for UN/CEFACT's Modeling Methodology;* Die Arbeit definiert ein formales Vorgehensmodell sowie eine Modellierungsmethode zur Erfassung von inter-organisationalen Geschäftsprozessen im Bereich des B2B eCommerce.	September 2000 - März 2006
	Höhere Technische Lehranstalt (HTL), Mödling, Niederösterreich Matura mit ausgezeichnetem Erfolg bestanden	Juni 1999
Berufliche Tätigkeiten	**Projektassistent** *Institut für Softwaretechnik und Interaktive Systeme, TU Wien* FFG Projekt: Business Semantics on top of Process Technology (BSopt)	seit Jänner 2008
	Externer Lektor *Universität Wien* Lehrveranstaltungen: Electronic Commerce, eBusiness	seit Oktober 2006
	Researcher *Research Studios Austria* Forschungsgruppe Intelligent Business Process Management. Forschungsschwerpunkte: Geschäftsprozessmodellierung mit UMM und service orientierte Architekturen	März 2006 - Dezember 2007
	Projektmitarbeiter *Bundesministerium für Wirtschaft und Arbeit* Design und Entwicklung eines Portals zur automatisierten Förderungsabwicklung für Benutzer des ebInterface Rechnungsstandards	April 2006 - November 2006

Projektmitarbeiter und Diplomand März 2005 - März 2006
Dept. of Distributed and Multimedia Systems, Universität Wien
Verfassung der Diplomarbeit innerhalb des Forschungsprojekts *eGov-Simple* in Zusammenarbeit mit den Austrian Research Centers GmbH (ARC).

Studienassistent März 2005 - Jänner 2006
Dept. of Distributed and Multimedia Systems, Universität Wien
Lehrveranstaltung: Electronic Commerce

Freier Dienstnehmer Jänner 2001 - Dezember 2005
Los Angeles Times
IT Support, sowie Implementierung und Wartung eines Web-Portals zum kollaborativen Artikelaustausch zwischen Journalisten

Freier Dienstnehmer Juli 2000 - Mai 2010
Ullram Werbegrafik und Webdesign, Halbturn
Umsetzung und Realisierung zahlreicher Webportale, Homepages, Corporate Design sowie Messeauftritte und Firmenpräsentationen

Projektleitung Oktober 2003 - Mai 2004
Österreichische Weinmarketing GmbH
Entwicklung eines CRM-Tools zur Unterstützung des österreichischen Weinmarketings

Praktikant Sommer 1997/98/99/00/01
Schrack Seconet AG, Wien
Programmierung von Steuerelementen für Brandmeldeanlagen sowie Wartung und Support von Lichtrufsystemen

Tutor März 2005 - Jänner 2006
Dept. of Distributed and Multimedia Systems, Universität Wien
Lehrveranstaltungen: Modellierungstechniken und -methoden, Informationsmanagement

Marketing 2000 - heute
Weingut Kummer-Schuster
Tätigkeiten im Bereich Weinwerbung, IT-Integration, Messen sowie die Pflege internationaler Kundenkontakte

IT-Trainer 1999 - 2005
Berufsförderungsinstitut Wien
Schulungsgegenstand: Java, HTML, Webdesign, Grafikprogramme (Photo Shop, Corel Draw)

WISSENSCHAFTL. PUBLIKATIONEN

Publikationen seit April 2006:

- 1 Buchveröffentlichung
- 2 Buchkapitel
- 2 Journalbeiträge

- 16 Publikationen bei internationalen Konferenzen

AUSZEICHNUNGEN	**1. Platz INiTS Award 2006** Ausgezeichnet mit dem INiTS Award für die Diplomarbeit - 1.Platz in der Kategorie Informations- und Kommunikationstechnologie. Der INiTS Award wird in Österreich für innovative Diplomarbeiten mit großer Chance auf wirtschaftliche Verwertbarkeit verliehen.	Oktober 2006
WEITERBILDUNG	**Università di Siena, Italien** Auslandsstudium im Rahmen des Erasmus-Programms	Jänner 2004 - Juli 2004
MITGLIEDSCHAFTEN	United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), Vollmitglied der *Technology and Methodology (TMG) Group* ACM (Association of Computing Machinery) IEEE (Institute of Electrical and Electronics Engineers, Inc.)	
FREMDSPRACHEN	Englisch - fließend in Wort und Schrift Italienisch - grundlegende Kenntnisse	
GRUNDWEHRDIENST	Stabskompanie, TÜPL Bruckneudorf	Juli 1999 - März 2000
PERSÖNLICHE INTERESSEN	Tennis, Fußball, Snowboard, Reisen, Mode	

List of Publications

The up-to-date list of publications can be found on
http://www.ec.tuwien.ac.at/~schuster.

Books

1. Marco Zapletal, Philipp Liegl, and Rainer Schuster. *UN/CEFACT's Modeling Methodology (UMM) 1.0 - A Guide to UMM and the UMM Add-In*. VDM Verlag Dr. Müller, 2008

Book Chapters

2. Rainer Schuster, Philipp Liegl, Marco Zapletal, Hannes Werthner, Christian Huemer, Christoph Grün, Dieter Mayerhofer, and Thomas Motal. *E-Business and Semantic Technologies*. In *Handbook of Semantic Web Technologies*. Springer, 2010 (to appear)
3. Christian Huemer, Philipp Liegl, Rainer Schuster, Marco Zapletal, and Birgit Hofreiter. *Service-Oriented Enterprise Modeling and Analysis*. In *Handbook of Enterprise Integration*, pages 307–322. Auerbach Publications, 2009

Journal Papers

4. Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *B2B Services: Worksheet-Driven Development of Modeling Artifacts and Code*. *The Computer Journal*, 52(8):1006–1026, 2009
5. Philipp Liegl, Rainer Schuster, Robert Mosser, and Marco Zapletal. *Modeling e-Government processes with UMM*. *Informatica Journal - An International Journal of Computing and Informatics*, 31(4):407–417, 2007

Conference and Workshop Papers

6. Rainer Schuster, Thomas Motal, Christian Huemer, and Hannes Werthner. *From Economic Drivers to B2B Process Models: a Mapping from REA to UMM*. In *13th International Conference on Business Information Systems BIS2010, Berlin, May 3-5, 2010*, 2010
7. Philipp Liegl, Rainer Schuster, Marco Zapletal, Christian Huemer, Hannes Werthner, Michael Aigner, Martin Bernauer, Bjoern Klinger, Michaela Mayr, Ramin Mizani, and Martin Windisch. *[vem:xi:] - A methodology for process based requirements engineering*. In *Proceedings of the 17th IEEE International Require-*

ments Engineering Conference (RE09), August 31 - September 4, Atlanta, GA, USA, pages 193–202. IEEE, 2009

8. Rainer Schuster and Thomas Motal. *From e3-value to REA: Modeling Multi-party E-business Collaborations*. In *CEC '09: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, pages 202–208, Washington, DC, USA, 2009. IEEE Computer Society
9. William McCarthy, Rainer Schuster, and Thomas Motal. *Modeling Multi-party Collaborations in e3-Value and REA: An Example and Some Preliminary Observations*. In *International Workshop on Value Modeling and Business Ontologies, VMBO'09, Stockholm - Feb 8-9, 2009*, 2009
10. Christian Huemer, Philipp Liegl, Thomas Motal, Rainer Schuster, and Marco Zapletal. *The Development Process of the UN/CEFACT Modeling Methodology*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08), August 19-22, Innsbruck, Austria*, pages 1–10. ACM, 2008
11. Christian Huemer, Philipp Liegl, Rainer Schuster, Hannes Werthner, and Marco Zapletal. *Inter-organizational Systems: From Business Values over Business Processes to Deployment*. In *Proceedings of the 2nd International IEEE Conference on Digital Ecosystems and Technologies (DEST2008), February 26-29, Phitsanulok, Thailand*, pages 294–299. IEEE, 2008
12. Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *A 3-level e-Business Registry Meta Model*. In *Proceedings of the IEEE International Conference on Services Computing (SCC2008), July 8-11, Honolulu, HI, USA*, pages 451–450. IEEE, 2008
13. Rainer Schuster and Thomas Motal. *A Holistic Approach Towards a UML Profile for Business Modeling*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08) CEUR-WS, August 19-22, Innsbruck, Austria*, 2008
14. Philipp Liegl, Thomas Motal, and Rainer Schuster. *An Add-In for UN/CEFACT's Modeling Methodology 2.0*. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC08), August 19-22, Innsbruck, Austria*, pages 1–2, 2008
15. Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Deriving executable BPEL from UMM Business Transactions*. In *Proceedings of the IEEE International Conference on Services Computing (SCC2007), July 9-13, Salt Lake City, UT, USA*, pages 178–186. IEEE, 2007
16. Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *The UMM Add-In - Demo*. In *Proceedings of the International Conference on Services Oriented Computing (ICSOC2007), September 17-20, Vienna, Austria*, pages 618–619. Springer, 2007
17. Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Modeling Business Entity State Centric Choreographies*. In *Proceedings of the IEEE Joint Conference on E-Commerce Technology (CEC07), July 23-26, Tokyo, Japan*, pages 393–400. IEEE, 2007

18. Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *Worksheet Driven UMM Modeling of B2B Services*. In *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2007), October 24-26, Hongkong, China*, pages 30–38. IEEE, 2007
19. Philipp Liegl, Robert Mosser, Rainer Schuster, and Marco Zapletal. *Modeling e-Government processes with UMM*. In *Proceedings of the VIP Symposia on Internet related research (VIPSI2007), June 7-10, Opatija, Croatia*, pages 1–11, 2007
20. Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *UN/CEFACT's Modeling Methodology (UMM): A UML Profile for B2B e-Commerce*. In *Proceedings of the 2nd International Workshop on Best Practices of UML (ER BP-UML'06), November 6-9, Tucson, AZ, USA*, pages 19–31. IEEE, 2006
21. Birgit Hofreiter, Christian Huemer, Philipp Liegl, Rainer Schuster, and Marco Zapletal. *UMM Add-In: A UML Extension for UN/CEFACT's Modeling Methodology*. In *Proceedings of the European Conference on Model Driven Architecture (ECMDA'06), July 10-13, Bilbao, Spain*, pages 618–619. Springer, 2006

Master Thesis

22. Philipp Liegl, Rainer Schuster, and Marco Zapletal. *A UML Profile and Add-In for UN/CEFACT's Modeling Methodology*. Master's thesis, University of Vienna, 2006