FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Business Context Aware Core Components Modeling

PhD THESIS

submitted in partial fulfillment of the requirements of

Doctor of Technical Sciences

within the

Vienna PhD School of Informatics

by

Danijel Novakovic
Registration Number 1028296

to the Faculty of Informatics
at the Vienna University of Technology

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

Wien, 25.08.2014

(Signature of Author) (Signature of Advisor)

Declaration of Authorship

Danijel Novakovic
Peter Jordan Straße 1/w508, 1190 Vienna, Austria

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

(Place, Date) (Signature of Author)

Acknowledgements

Foremost, I would like to thank my advisor Prof. Christian Huemer for his continuous support during my PhD study and research, for his patience, encouragement, thoughtful guidance, critical comments and correction of this thesis. His guidance helped me in all the time of this research. However, he also gave me a great freedom to pursue an independent work. It was a great and exciting experience being his student and working with him.

I would like to thank Prof. Stefanie Rinderle-Ma from the University of Vienna and Prof. Henderik A. Proper from the Public Research Centre Henri Tudor for reviewing this thesis.

I am grateful to the Vienna PhD School of Informatics for making it possible for me to study here. My special and heartily thanks to Prof. Hannes Werthner, my first academic supervisor in Austria. His invaluably constructive criticism, friendly advices and continuous support helped me a lot to proceed through the PhD program and complete this thesis. Likewise, my sincerest thanks to Prof. Hans Tompits for his dedication in leading and organizing our PhD school.

I express my warm thanks to Prof. Gerti Kappel. In the early beginning of my PhD studies she encouraged me to define my research area and helped me in establishing the first contact with my PhD supervisor. Afterwards, as the head of our Business Informatics Group, she was unconditionally supporting our research.

Furthermore, I am also thankful to all my colleagues and friends from both the Vienna PhD School of Informatics and the Business Informatics Group. Their friendship and assistance has meant more to me than I could ever express.

Finally, I thank and dedicate this thesis to my parents who have believed in me and supported me through my entire life.

Abstract

Business document standards usually cover a hierarchical structure of thousands of elements that may be relevant in any business context (any geopolitical region, any industry, etc.). In order to use a business document standard in a specific context, user groups define so-called business document implementation guidelines based on a smaller subset consisting usually of 3 – 5% of the overall elements. When one defines a new implementation guideline for a specific business context, one has always to start from scratch, which is time-consuming and also leads to heterogeneous interpretations of the standard. It is our goal to speed up the development process and to create more homogeneous implementation guidelines by learning from existing models. If we could assign a formal context to existing implementation guidelines, we may predict the subset of a new implementation guideline for a given context.

We especially consider implementation guidelines built on the top of the Core Components Technical Specification (CCTS). These guidelines consist of business context specific data building blocks which are restricted from more general, semantically interoperable Core Components. In order to share, search, and (partially) re-use context specific restrictions of Core Components it is essential not only to store the restrictions, but also a business context model where these restrictions are valid. Therefore, we develop the Enhanced UN/CEFACT Business Context Model (E-UCM) and the Business Context Ontology Model (BCOnt) for representing business context in the domain of Core Components.

This thesis proposes an approach to contextualize already existing Core Components by means of our business context models. This contextual information is also used to predict a subset for to-be-developed electronic business document implementation guidelines based on existing ones. The underlying algorithms calculate degrees of business context match, detect different types of mappings between existing Core Components and generate Core Component based contents of new implementation guidelines.

Our research has been conducted following the design science research process. The corresponding evaluation is interpreted as a build and evaluate loop iterated a number of times before the final approach was developed. We evaluate the business context models on the basis of 16 evaluation criteria. The feasibility of the business context aware Core Components modeling approach is demonstrated by a prototype implementation. The analysis of the calculated precision and recall rates proves that our approach holds not only in theory, but also in practice.

Contents

CHAPTER 1

Introduction

Inter-organizational business processes, such as electronic procurement, are supported by the exchange of business documents. The concrete structures of these documents significantly differ depending on the business context (geopolitical region, industry branch, etc.) in which the underlying inter-organizational business processes are executed. In order to ensure interoperability of the exchanged information, parties in inter-organizational business processes must agree on a common business document standard used for building business documents.

No matter whether business document standards are traditional EDI standards [1] or XML-based standards [2], they are very generic including all elements that may be of need to any company in this world. Before being used in partnerships, business context specific subsets of these elements - called *business document (message) implementation guidelines* - have to be defined. However, implementation guidelines usually take only small portions of the document standard, not more than 3 – 5% of its overall elements. Definitions of the guidelines always start from scratch. This task is, thus, very time-consuming and leads to heterogeneous interpretations of the standard.

Instead of defining new business document implementation guidelines from scratch, it would be beneficial to provide methods for supporting the re-use of already existing documents which are exchanged between business partners when executing inter-organizational business processes. This Doctoral Thesis describes our approach to utilize the business contextual knowledge for (semi-) automatically generating semantically interoperable data building blocks, so-called Core Components, contained by electronic business documents. These documents conform to the business document standard built on the top of the Core Components Technical Specification (CCTS) [3].

In the remainder of this Chapter we (i) briefly introduce Core Components, (ii) explain the meaning of business context, (iii) describe the research problem statement, and (iv) present the overview of the methodology followed during the research.

1.1 Core Components

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) is a Standards Development Organization known for its work on trade facilitation recommendations and electronic business standards. A cornerstone of the UN/CEFACT standardization activities is the Core Components Technical Specification (CCTS) [3]. It is a methodology which aims at standardizing business documents for electronic interchange. Accordingly, electronic business documents are *structured* consisting of semantically interoperable data building blocks, called Core Components.

A *Core Component* is a general data building block which can appear in different business contexts such as the Automotive industry in Austria or the Food industry in Japan. It is either an atomic data block or an aggregate of data blocks which form a common semantic meaning. Before a Core Component can be used for assembling an electronic business document, it must be tailored to a specific business context. A Core Component used in a specific context represents the *Business Information Entity*. It is essential that every Business Information Entity is *derived* from the corresponding Core Component by *business context aware restriction*.

1.2 Business Context

A business context defines the set of circumstances, such as geopolitical region and industry branch, in which some CCTS based electronic business document is valid. In order to share, search, and (partially) re-use context specific restrictions of Core Components it is essential not only to store the restrictions, but also a business context model where these restrictions are valid.

This thesis develops two business context models: (i) the Enhanced UN/CEFACT Business Context Model (E-UCM) and (ii) the Business Context Ontology Model (BCOnt). The E-UCM model follows the logic based approach for context modeling. It is built upon the already existing UN/CEFACT Business Context Model (UCM) [4] which is designed to manage representations of business context under the scope of the CCTS standardization. In contrary, the BCOnt model follows the ontology based approach for context modeling. It comprises a hierarchy of ontologies for representing business context in which Business Information Entities of electronic business documents are valid.

1.3 Problem Statement

This Doctoral Thesis develops an approach to speed up the development of electronic business document implementation guidelines and to create more homogeneous business document implementation guidelines by learning from existing models. Generally speaking, it utilizes business contextual knowledge comprised by existing guidelines to generate a new guideline which is valid in the specific business context. The considered implementation guidelines conform to the CCTS business document standard.

Main Use Case Scenario. Figure 1.1 illustrates the set of at least two business domains which are valid in different business contexts. In the first domain there are at least two related

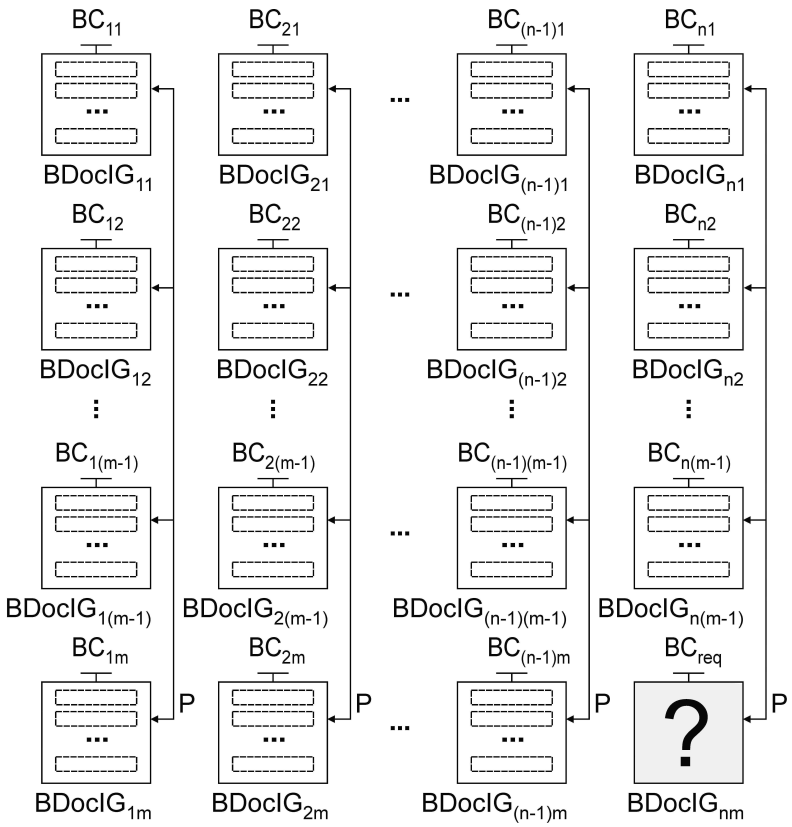


Figure 1.1: Problem statement

electronic business documents ($BDocIG_{11}$, $BDocIG_{12}$,..., $BDocIG_{1m}$). In the second domain one of the related documents is missing. This Doctoral Thesis develops the algorithm which generates the implementation guideline of the missing business document ($BDocIG_{nm}$ shown in Figure 1.1). The corresponding problem statement is expressed by Formula:

$$\begin{aligned} BDocIG_{nm} = Alg(&P(BDocIG_{11}, BDocIG_{12}, ..., BDocIG_{1(m-1)}, BDocIG_{1m}), \\ &P(BDocIG_{21}, BDocIG_{22}, ..., BDocIG_{2(m-1)}, BDocIG_{2m}), ..., \\ &P(BDocIG_{(n-1)1}, BDocIG_{(n-1)2}, ..., BDocIG_{(n-1)(m-1)}, BDocIG_{(n-1)m}), \\ &BDocIG_{n1}, ..., BDocIG_{n(m-1)}, BC_{req}). \end{aligned} \tag{1.1}$$

Alg represents the algorithm which calculates the Core Component structure of the missing guideline ($BDocIG_{nm}$). The relationship $P(BDocIG_1, BDocIG_2, ..., BDocIG_m)$ denotes that the specified implementation guidelines are related. The BC_{req} is the requested business context in which the generated $BDocIG_{nm}$ will be valid. *n* and *m* are natural numbers greater than 1 ($n, m \in N,\ and\ n, m > 1$). *n* denotes the number of business domains, while *m* denotes the number of related documents in business domains.

1.4 Overview of the Applied Methodology

This Doctoral Thesis follows the design science research (DSR) [5] methodology. Accordingly, this research is based on a heuristic and iterative design search process until the final algorithms are built. It is divided into well-defined research phases which are highlighted in the overview of the underlying research framework in Figure 1.2.

Overview of the Research Framework. The specification language for representing business context of arbitrary complexity is the main prerequisite to utilize business contextual knowledge and to develop missing business document implementation guidelines. Therefore, the first research phase described in this Doctoral Thesis (Figure 1.2, Mark 1) defines business context and business context awareness. The following research phases 2 and 3 (Figure 1.2, Marks 2 and 3, respectively) denote the concurrent work which develops two models of the previously defined business context ((i) the Enhanced UN/CEFACT Model and (ii) the Business Context

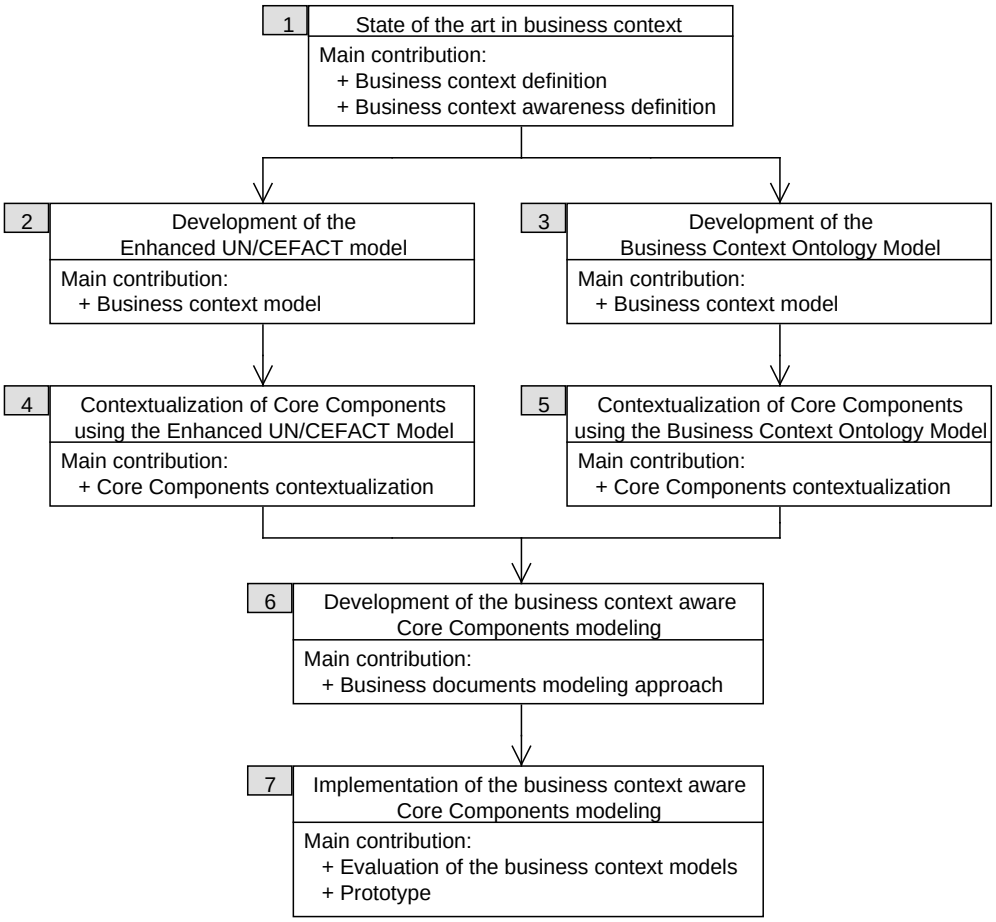


Figure 1.2: Overview of the research framework

Ontology Model).

The developed business context models serve as a basis to contextualize the Core Components comprised by the existing business document implementation guidelines. The corresponding work is conducted within the research phases 4 and 5 (cases using the Enhanced UN/CEFACT Model and the Business Context Ontology Model, respectively). The contextualized Core Components contain the business contextual knowledge which can be used to (semi-) automatically *derive* the missing sets of Core Components *by business context aware restriction*. The corresponding algorithm (represented by Formula 1.1) is designed within the research phase 6 (Figure 1.2, Mark 6). The following research phase 7 (Figure 1.2, Mark 7) implements the prototype of the developed conceptual solutions and concludes the evaluation remarks ((i) the evaluation of the business context models and (ii) the evaluation of the business context aware Core Components modeling approach).

The contributions of the work presented in this Doctoral Thesis address the gaps in different domains, such as the standard context theory [6], UN/CEFACT Context Methodology [4] and the Core Components Technical Specification [3]. The most important of them are: (i) the definition of business context (research phase 1), (ii) the definition of business context awareness (research phase 1), (iii) the Enhanced UN/CEFACT Business Context Model (research phase 3), (iv) the Business Context Ontology Model (research phase 4), (v) the Core Components contextualization (research phases 5 and 6), and (vi) the business context aware Core Components modeling approach (research phase 7).

The more detailed explanations of the applied methodology - including the complete list of the research contributions - are discussed in the following Chapters of this thesis.

1.5 Structure of the Thesis

The Chapters of this Doctoral Thesis are organized in a self-contained manner. It is, thus, possible to read each Chapter without knowing the previous Chapters. However, since each Chapter describes a certain phase of the coordinated research, it is recommended to read them in chronological order.

Chapter 2 ”UN/CEFACT’s Core Components” explains the foundations of the UN/CEFACT’s Core Components Technical Specification. Therefore, it defines the Core Component based communication models for exchange of business data. It highlights the importance of context in business collaborations and explains the influence of business context on the introduced data model components. Finally, this Chapter describes how business messages can be assembled from Core Components and represented by XML based syntax.

Chapter 3 ”Business Context” addresses the research phase 1 of our research framework (Figure 1.2, Mark 1). It summarizes relevant definitions of context and context awareness applied in different domains. Based on the outcomes of the corresponding survey, it defines context and context awareness in the domain of electronic business documents which are exchanged between business partners when executing inter-organizational business processes. Furthermore, this Chapter discusses different context modeling techniques, and highlights their benefits and

drawbacks. Finally, it presents guidelines for modeling the business context in which electronic business documents are valid.

Chapter 4 ”Enhanced UN/CEFACT Business Context Model” describes the research phases 2 and 4 of our research framework (Figure 1.2, Marks 2 and 4, respectively). First, it introduces the UN/CEFACT Context Methodology [4] and explains the shortcomings of its application. Afterwards, it resolves the detected shortcomings and develops the Enhanced UN/CEFACT Model (E-UCM). This is a context specification model used to represent business context as defined in Chapter 3. Finally, this Chapter 4 shows how the already existing Core Components can be contextualized by means of the Enhanced UN/CEFACT Business Context Model.

Chapter 5 ”Business Context Ontology” addresses the research phases 3 and 5 of our research framework (Figure 1.2, Marks 3 and 5, respectively). First, it develops the Business Context Ontology Model (BCOnt) which can be used to represent business context as defined in Chapter 3. Afterwards, it elaborates on the benefits of the chosen business context modeling approach. Finally, this Chapter explains how the already existing Core Components can be contextualized by means of the Business Context Ontology Model.

Chapter 6 ”Business Context Aware Core Components Modeling” presents the research phase 6 of our research framework (Figure 1.2, Mark 6). It describes the core of this Doctoral Thesis - our business context aware Core Components modeling approach. First, this Chapter defines different types of business context matchings. Second, it introduces the business context reasoning. Afterwards, it describes different types of Core Components mappings. Finally, this Chapter exploits these capabilities and proposes the algorithm which realizes our business context aware Core Components modeling approach.

Chapter 7 ”Implementation” represents the research phase 7 of our research framework (Figure 1.2, Mark 7). It describes the implementation of the business context aware Core Components modeling approach (discussed in Chapter 6). First, Chapter 7 explains how the contextualized Core Components can be represented using XML based syntax. Second, it develops the prototypical architecture which realizes our business context aware Core Components modeling approach. Finally, this Chapter explains the most important algorithms which implement the execution unites of the proposed architecture.

Chapter 8 ”Methodology and Evaluation” describes the methodology which was followed during the research on the business context aware Core Components modeling. It addresses all research phases illustrated in our research framework in Figure 1.2. It especially highlights the evaluation and contributions of the work presented in the previous Chapters. Furthermore, it elaborates on the main artifacts of the research and lists our main academic publications.

Finally, Chapter 9 concludes the research described in this Doctoral Thesis. It summarizes the main remarks of the presented work and gives an outlook on possible future research directions.

All Figures, Formulas and Definitions are sequentially numbered for each Chapter. All Chapters contain the illustrative examples which describe the concrete research problems and demonstrate the proposed solutions.

CHAPTER 2

UN/CEFACT's Core Components

Before business partners can initiate a collaboration, they must reach an agreement on the structure and content of the communication models established between them. These models are usually constrained by some of the standardized business document formats for exchange of business data. The research described in this thesis builds upon the electronic business document formats which are proposed by the United Nations Centre for Trade Facilitation and Electronic Business's (UN/CEFACT's) Core Components based document standard. This standardization approach defines transfer syntax neutral and technology independent building blocks that can be used for data modeling. Consequently, the already existing data contents can be re-used and the interoperability of the exchanged information is significantly improved.

The remainder of this Chapter is structured as follows. First, Section 2.1 introduces the main foundations of the UN/CEFACT's Core Components based business document standard. The corresponding communication models for exchange of business data are defined in Section 2.2 and Section 2.3. The exchange data formats are explained using the illustrated examples. Section 2.4 highlights the importance of context in business collaborations and discusses the influence of business context on the introduced message communication models. Section 2.5 describes how these transfer syntax neutral models can be represented using the XML based syntax. The combined instances of the UN/CEFACT's communication models form the business messages which are built upon the Business Message Model introduced in Section 2.6. Finally, Section 2.7 summarizes the most important segments of the UN/CEFACT's Core Components based standardization effort.

2.1 Core Components Technical Specification

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) [7] is a Standards Development Organization established by the United Nations. It is known for its work on trade facilitation recommendations and electronic business standards, such as UN/EDIFACT [8] and ebXML [9]. A cornerstone of the UN/CEFACT standardization activities is the

Core Components Technical Specification (CCTS) [3]. It is a methodology which main goal is the standardization of business documents for electronic interchange.

The Core Components Technical Specification defines models and rules for describing the structures and contents of information exchange models. This is an implementation neutral standardization approach which defines information exchange models on a generic and conceptual level without considering any specific implementation syntax. According to the standard, every business document consists of business data which are included by semantically interoperable data building blocks.

The Core Components Technical Specification distinguishes between two primary concepts: (i) Core Components (CCs) and (ii) Business Information Entities (BIEs). Core Components are context free models of data building blocks for assembling business messages and documents. These general types of business data can be additionally tailored to address the particular circumstances, such as geopolitical region, industry and activity, which characterize the particular business collaboration. Therefore, Business Information Entities are context specific models of data building blocks derived by restriction of the underlying Core Components. It is essential that existing Core Components can only be restricted. Therefore, it is not possible to add new elements, such as additional attributes or relationships. Consequently, the interoperability of the business documents based on the CCTS business document standard is guaranteed.

The research described in this thesis builds upon the CCTS standard. Therefore, in the rest of this Chapter we explain the core concepts of the CCTS business document standardization effort which are important for understanding the following Chapters.

2.2 Core Components

Core Components (CCs) represent conceptual data model components for the creation of business messages and business documents. They are not specific to any particular business context, and, thus, can be used in any business scenario.

The Core Component Library (CCL) [10] represents the collection of all Core Components approved by UN/CEFACT. The Core Components stored in this library can be used for assembling business messages and business documents exchanged within inter-organization business processes in different business contexts. In case that some Core Component has not yet been defined, a request for inclusion of the new Core Component in the Core Component Library can be submitted to UN/CEFACT. New versions of the Core Component Library are usually published twice a year.

Core Components consist of three main entity types: (i) Basic Core Components (BCCs), (ii) Aggregate Core Components (ACCs) and (iii) Association Core Components (ASCCs). A Basic Core Component is a piece of information which is located in a business document. Each Aggregate Core Component represents a collection of Basic Core Components. Relations between Aggregate Core Components are established by Association Core Components. For a better understanding one could – even if not 100% correct – compare Aggregate Core Components as classes, Basic Core Components as attributes and Association Core Components as associations.

2.2.1 Aggregate Core Components

An Aggregate Core Component (ACC) represents a grouping of related pieces of information that together form a distinct meaning. It is essential that Aggregate Core Components are generic, and, thus, can be used in any business domain. Expressed in data modeling terms, Aggregate Core Components are represented as entity/object classes which are independent of any business context. They contain attributes/properties and may be associated with other Aggregate Core Components.

According to CCTS, every Aggregate Core Component must follow the naming rule which conforms to the lexical notation:

$$ACCNamingNotation = < ACCObjectClassTerm > .\_Details.$$

An *ACC object class term* is a semantically meaningful name for a unique identification of the particular Aggregate Core Component.

Aggregate Core Components - Example. The set of the Core Components is illustrated in the example shown in Figure 2.1. For instance, this set contains the Aggregate Core Component named *ProductGroup. Details*. This Core Component comprises the attributes/properties (*Size*, *Identification*, *Name*, etc.) and is associated with the Aggregate Core Component named *Product. Details*.

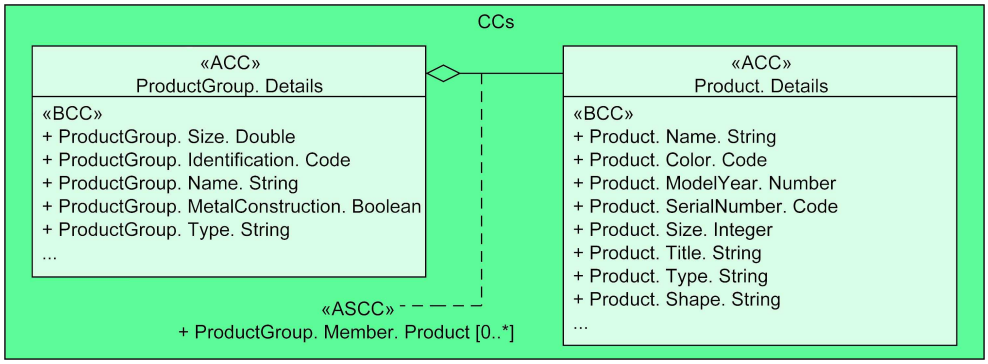


Figure 2.1: Example - UN/CEFACT's Core Components

2.2.2 Basic Core Components

A Basic Core Component (BCC) represents a property of an Aggregate Core Component. It is essential that Basic Core Components are generic, and, thus, can be used in any business domain. Expressing in data modeling terms, Basic Core Components are represented as entity attributes or class properties which are independent of any business context.

Each Basic Core Component must be uniquely identified. According to CCTS, Basic Core Components must follow the naming rule which conforms to the lexical notation:

$$BCCNamingNotation = < ACC\_ObjectClassTerm > .\_ < BCC\_Property > .$$

An *ACC object class term* identifies the Aggregate Core Component which comprises the specified Basic Core Component. In the corresponding example shown in Figure 2.2, the identified Aggregate Core Component is denoted by *ProductGroup. Details*. This Core Component is already introduced in our example illustrated in Figure 2.1.

A *BCC property* represents a generic re-usable data element independent of an *ACC object class*. Each *BCC property* must follow the naming rule which conforms to the lexical notation:

$$<BCC\_Property>=<BCC\_PropertyTerm>.\_<BCC\_RepresenationTerm>.$$

A *BCC property term* is a semantically meaningful name for a unique identification of the Basic Core Component in the scope of its including Aggregate Core Component (specified by the *ACC object class term*). A *BCC representation term* identifies the Core Data Type of the Basic Core Component. For instance, in the example shown in Figure 2.2, the Basic Core Component is uniquely identified by its name *Size* within the Aggregate Core Component named *ProductGroup. Details*. Furthermore, this Basic Core Component resolves the value which belongs to the Core Data Type denoted by *Double*.

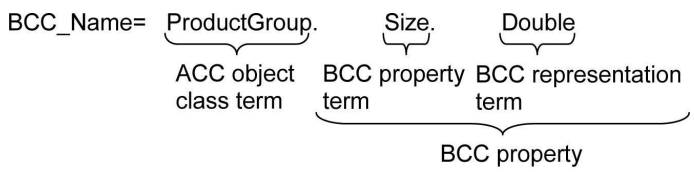


Figure 2.2: Example - Basic Core Components naming rule

Core Data Types. A Core Data Type (CDT) defines the value domain of the Basic Core Components. The allowed values of the Basic Core Components can be expressed by either primitive data types or schemas or lists of the allowed values. A Core Data Type must be one of the approved types published in the UN/CEFACT's Data Type Catalogue [11] (Binary, Boolean, Decimal, Double, Float, Integer, NormalizedString, String, TimeDuration, TimePoint, and Token).

Basic Core Components - Example. The Aggregate Core Component named *Product. Details* is presented in our already introduced example shown in Figure 2.1. This Core Component comprises the Basic Core Components named *Product. Name. String*, *Product. Color. Code*, *Product. ModelYear. Number*, etc. For instance, the Basic Core Component named *Product. Name. String* is identified by its *BCC property term* expressed as *Name* and refines the values of the type *String*.

Furthermore, as already emphasized, it is possible that some Basic Core Component may be re-usable across different Aggregate Core Components. For instance, in our example in Figure 2.1, this is the Basic Core Component which *BCC property* is expressed by *Name. String*. This Core Component is used in two different Aggregate Core Components named *ProductGroup. Details* and *Product. Details*. However, once a re-usable Basic Core Component has been assigned to the Aggregate Core Component, it becomes the Basic Core Component which is unique to its including Aggregate Core Component. In the example shown in Figure 2.1 these are the following uniquely identified Basic Core Components: *Product. Name. String* (belongs

to the Aggregate Core Component named *Product. Details*) and *ProductGroup. Name. String* (belongs the Aggregate Core Component named *ProductGroup. Details*).

2.2.3 Association Core Components

An Association Core Component (ASCC) defines the relation between one Aggregate Core Component (denoted by *associating ACC*, or by *parent ACC*) and another Aggregate Core Component (denoted by *associated ACC*). It is essential that Association Core Components are generic, and, thus, can be used in any business domain. Expressing in data modeling terms, Association Core Components are represented as aggregations which are independent of any business context.

Every Association Core Component must be uniquely identified. According to CCTS, Association Core Components must follow the naming rule which conforms to the lexical notation:

$$ASCCNamingNotation = <ASCC\_AssociatingObjectClassTerm>.\sqcup \\ <ASCC\_Property> .$$

An *Associating object class term* identifies the associating (parent) Aggregate Core Component. In the corresponding example shown in Figure 2.3, the identified associating Aggregate Core Component is denoted by *ProductGroup. Details*. This Core Component is already introduced in our example illustrated in Figure 2.1.

Analogously to the *BCC properties* introduced in the previous Subsection, an *ASCC property* represents a generic re-usable data element which is independent of an *object class*. Every *ASCC property* must follow the naming rule which conforms to the lexical notation:

$$<ASCC\_Property>= <ASCC\_PropertyTerm>.\sqcup \\ <ASCC\_AssociatedObjectClassTerm> .$$

An *ASCC property term* is a semantically meaningful name which uniquely identifies the Association Core Component in the scope of its associating Aggregate Core Component. An *Associated object class term* uniquely identifies the associated Aggregate Core Component. For instance, in the example shown in Figure 2.3, the Association Core Component is uniquely specified by its name *Member*. Its associated Aggregate Core Component is uniquely specified by its name *Product. Details*. These Core Components are already introduced in our example shown in Figure 2.1.

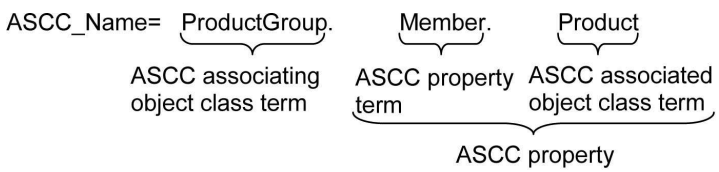


Figure 2.3: Example - Association Core Components naming rule

Association Core Components - Example. The Association Core Component named *ProductGroup. Member. Product* is shown in the example illustrated in Figure 2.1. It defines the association between its associating Aggregate Core Component named *ProductGroup. Details* and its associated Aggregate Core Component named *Product. Details*.

2.3 Business Information Entities

Business Information Entities (BIEs) represent logical data model components which are, in contrary to the Core Components explained in the previous Section, business context specific. Therefore, each Business Information Entity is created through the application of the business context, and, thus, can be used to assemble the business documents valid only in the context restricted business scenarios.

Analogously to the Core Component concept, Business Information Entities consist of three main entity types: (i) Basic Business Information Entities (BBIEs), (ii) Aggregate Business Information Entities (ABIEs) and (iii) Association Business Information Entities (ASBIEs). It is essential that each Business Information Entity is *derived by restriction* from the corresponding Core Component in respect to the particular business context. Consequently, a Business Information Entity must not contain attributes/properties or associations which are not defined in the underlying Core Component. The relationships between Business Information Entities and Core Components are presented in Figure 2.4 and explained in the following.

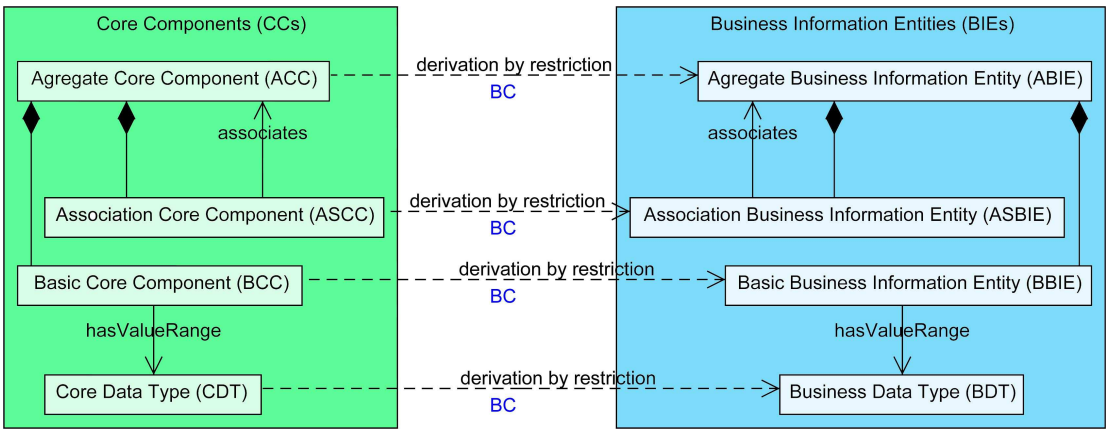


Figure 2.4: Relationships between Core Components and Business Information Entities

2.3.1 Aggregate Business Information Entities

An Aggregate Business Information Entity (ABIE) represents the business contextual specialization of the underlying Aggregate Core Component (ACC). Therefore, as shown in Figure 2.4, every Aggregate Business Information Entity is business context restricted, and, thus, can be used only in context specific business scenarios. In data modeling terms, analogously to Aggregate Core Components, Aggregate Business Information Entities are represented as entity/object

classes. They contain attributes/properties and may be associated with other Aggregate Business Information Entities.

According to CCTS, Aggregate Business Information Entities must follow the naming rule which conforms to the lexical notation:

$$ABIENamingNotation = <ABIEObjectClassTerm>.\_Details.$$

An *ABIE object class term* is defined as:

$$<ABIEObjectClassTerm>=<BC\_Qualifier><ACCObjectClassTerm>.$$

A *BC qualifier* specifies the business contextual restriction of the underlying Aggregate Core Component which is identified by its *ACC object class term*. The definition of the *ACC object class term* is already explained in Section 2.2.1.

Aggregate Business Information Entity Naming Rule - Example. The ABIE naming rule is presented in Figure 2.5 on the example of the Aggregate Business Information entity named *FoodIndustry_ProductGroup. Details*. This Business Information Entity is *derived by restriction* from the Aggregate Core Component named *ProductGroup. Details*. This Core Component is already introduced in our example shown in Figure 2.1. The restriction is conducted in respect to the business context which specifies the business scenario valid in the Food industry.

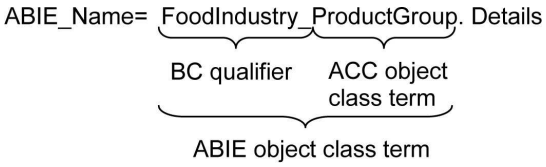


Figure 2.5: Example - Aggregate Business Information Entities naming rule

2.3.2 Basic Business Information Entities

A Basic Business Information Entity (BBIE) represents the business contextual specialization of the underlying Basic Core Component (BCC). Therefore, as shown in Figure 2.4, every Basic Business Information Entity is business context restricted, and, thus, can be used only in context specific business scenarios. In data modeling terms, analogously to Basic Core Components, Basic Business Information Entities are represented as attributes or class properties.

Every Basic Business Information Entity must be uniquely identified. The corresponding naming rule follows the lexical notation:

$$BBIENamingNotation =<ABIE\_ObjectClassTerm>.\_<BBIE\_Property>.$$

An *ABIE object class term* is defined as the business context restricted *ACC object class term* (explained in Section 2.2.2). Likewise, a *BBIE property* is defined as the business context restricted *BCC property* (explained in Section 2.2.2). These business contextual restrictions are

denoted by the corresponding qualifiers. In case that a business context qualifier of an *ABIE object class term* is the same as the business context qualifier of the following *BBIE property*, the business context qualifier of the *BBIE property* can be omitted.

Basic Business Information Entities Naming Rule - Example. The BBIE naming rule is presented in Figure 2.6 on the example of the Basic Business Information entity named *FoodIndustry_ProductGroup. MilkIndustry_Size. CondensedMilkIndustry_Double*. This Business Information Entity is *derived by restriction* from the Basic Core Component named *ProductGroup. Size. Double* (already introduced in our example shown in Figure 2.1). The restriction is conducted in respect to the business context which specifies the business scenario valid in the Milk industry. In our example shown in Figure 2.6, this is denoted by the *MilkIndustry_* business context qualifier followed by the *BCC property term*. The definition of the *BCC property term* is already explained in Section 2.2.2.

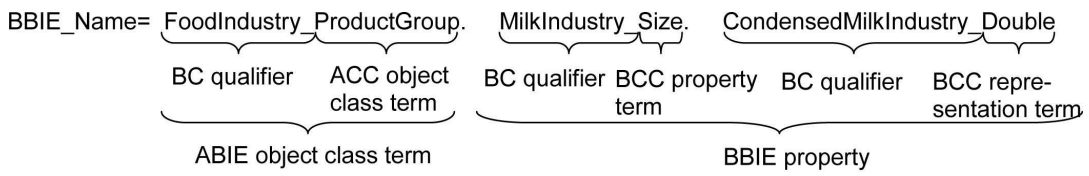


Figure 2.6: Example - Basic Business Information Entities naming rule

Furthermore, the Basic Business Information Entity named *FoodIndustry_ProductGroup. MilkIndustry_Size. CondensedMilkIndustry_Double* belongs to the Aggregate Business Information Entity named *FoodIndustry_ProductGroup. Detalis*. This Business Information Entity is *derived by restriction* from the Aggregate Core Component named *ProductGroup. Details* (already introduced in our example shown in Figure 2.1). The restriction is conducted in respect to the business context which specifies the business scenario valid in the Food industry. In our example shown in Figure 2.6, this is denoted by the *FoodIndustry_* business context qualifier followed by the *ACC object class term*. The definition of the *ACC object class term* is already explained in Section 2.2.2.

Finally, the Basic Business Information Entity named *FoodIndustry_ProductGroup. MilkIndustry_Size. CondensedMilkIndustry_Double* can resolve the values which belong to the Business Data Type denoted by *CondensedMilkIndustry_Double*. This set of possible values is derived by restriction from the set of the business context values identified by the Core Data Type denoted by *Double*. In our example shown in Figure 2.6, this restriction is marked as the *CondensedMilkIndustry_* business context qualifier followed by the *BCC representation term*. The definition of the *BCC representation term* is already explained in Section 2.2.2.

Business Data Types. A Business Data Type (BDT) defines the value domain of the Basic Business Information Entities. It is based on the business contextual specialization of the underlying Core Data Type. Therefore, as shown in Figure 2.4, every Business Data Type is business context restricted, and, thus, can be used only in the context specific business scenarios.

2.3.3 Association Business Information Entities

An Association Business Information Entity (ASBIE) represents the business contextual specialization of the underlying Association Core Component (ASCC). Therefore, as shown in Figure 2.4, every Association Business Information Entity is business context restricted, and, thus, can be used only in the context specific business scenarios. In data modeling terms, analogously to Association Core Components, Association Business Information Entities are represented as aggregations.

Every Association Business Information Entity must be uniquely identified. The corresponding naming rule follows the lexical notation:

$$ASBIENamingNotation = <ASBIE\_AssociatingObjectClassTerm>.\_ \\ <ASBIE\_Property>.$$

An *ASBIE associating object class term* is defined as the business context restricted *ASCC associating object class term* (explained in Section 2.2.3). Likewise, an *ASBIE property* is defined as the business context restricted *ASCC property* (explained in Section 2.2.3). These business contextual restrictions are denoted by the corresponding qualifiers. In case that a business context qualifier of an *ASCC associating object class term* is the same as the business context qualifier of the following *ASBIE property*, the business context qualifier of the *ASBIE property* can be omitted.

Association Business Information Entities Naming Rule - Example. The ASBIE naming rule is presented in Figure 2.7 on the example of the Association Business Information Entity named *FoodIndustry_ProductGroup. MilkIndustry_Member. CondensedMilkIndustry_Product*. This Association Business Information Entity correlates the Aggregate Business Information Entity named *FoodIndustry_ProductGroup. Details* with the Aggregate Business Information Entity named *CondensedMilkIndustry_Product. Details*. It is *derived by restriction* from the Association Core Component named *ProductGroup. Member. Product* (already introduced in our example shown in Figure 2.1). The restriction is conducted in respect to the business context which specifies the business scenario valid in the Milk industry. In our example shown in Figure 2.6, this is denoted by the *MilkIndustry_* business context qualifier followed by the *ASCC property term*. The definition of the *ASCC property term* is already explained in Section 2.2.2.

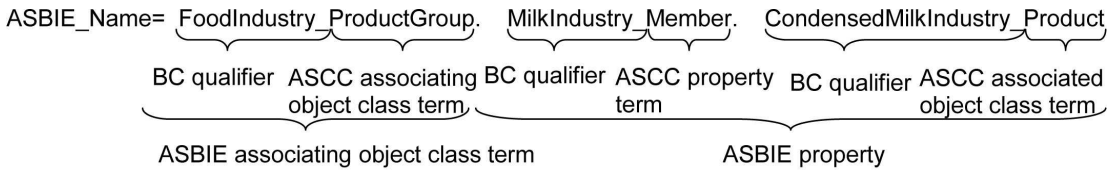


Figure 2.7: Example - Association Business Information Entities naming rule

2.3.4 Example of Derivation by Restriction

A business document - based on the CCTS document standard and exchanged between business partners when executing inter-organization business processes - comprises the set of the Business

Information Entities. The included Business Information Entities are derived by restriction from the corresponding Core Components in respect to the particular business context. This context addresses all circumstances, such as geopolitical region, industry and activity, which describe the collaboration between the business partners.

The example of the *derivation by restriction* is illustrated in Figure 2.8. For educational reasons, the business context qualifiers (explained in the previous Sections) only highlight industry business context domain and discard the geopolitical region and activity business context domains.

The set of the Core Components - already introduced in the example shown in Figure 2.1 - is presented in Figure 2.8, Mark 1. The set of the Business Information Entities derived from these Core Components is illustrated in Figure 2.8, Mark 2. The corresponding *derivation by restriction* is conducted in respect to the business context which involves the business scenario related to the invoicing in the Automotive industry in the European Union. For instance, the Ag-

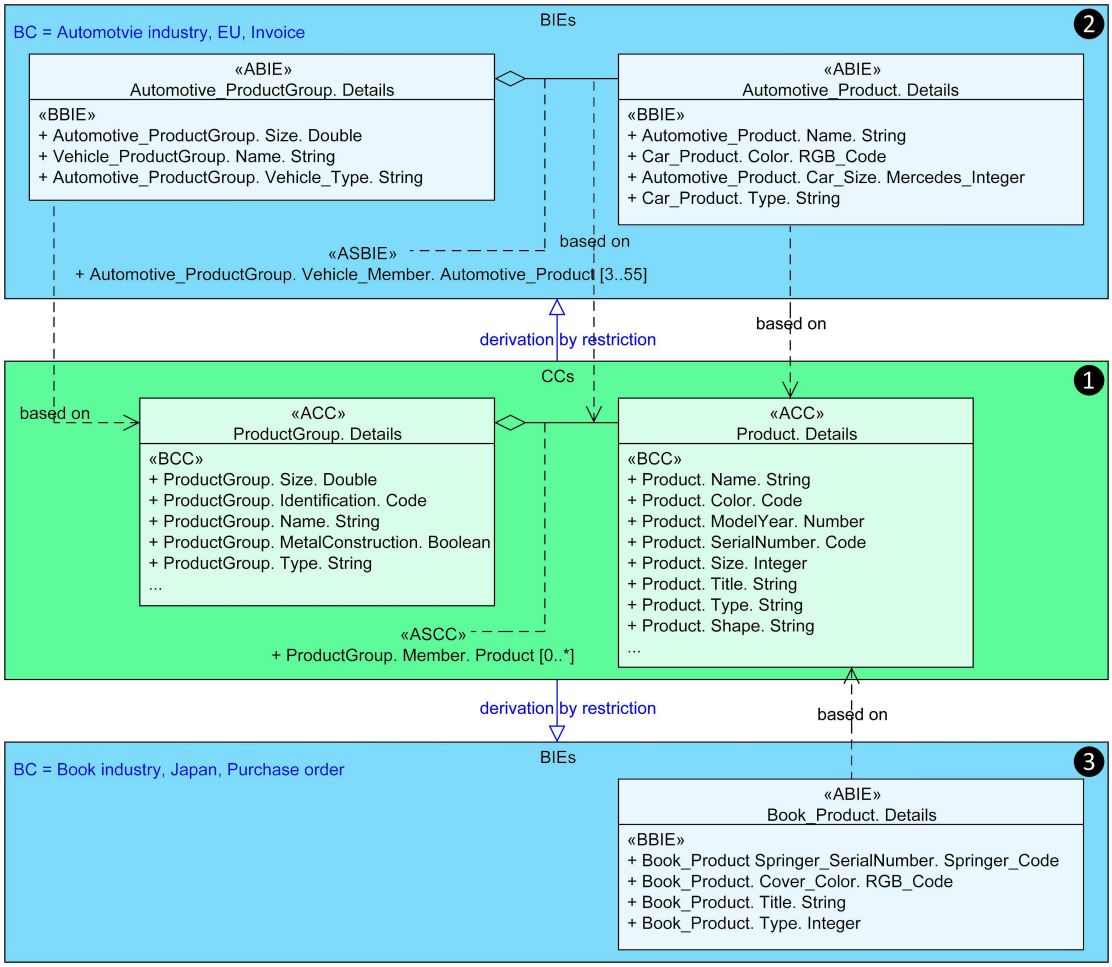


Figure 2.8: Example - derivation by restriction

gregate Core Component named *ProductGroup. Details* is restricted to the Aggregate Business Information Entity named *Automotive_ProductGroup. Details*. Likewise, the Aggregate Core Component named *Product. Details* is restricted to the Aggregate Business Information Entity named *Automotive_Product. Details*. Finally, the Association Core Component named *ProductGroup. Member. Product* with its cardinality *0..** is restricted to the Association Business Information Entity named *Automotive_ProductGroup. Vehicle_Member. Automotive_Product* with the new cardinality *3..55*.

Another set of the Business Information Entities is illustrated in Figure 2.8, Mark 3. This set of the Business Information Entities is derived from the same set of Core Components as the set of the Business Information Entities illustrated in Figure 2.8, Mark 2. However, the corresponding *derivation by restriction* is conducted in respect to a different business context. This new context describes the business scenario related to the purchase ordering in the Book industry in Japan. For instance, the Aggregate Core Component named *Product. Details* is restricted to the Aggregate Business Information Entity named *Book_Product. Details*. Furthermore, this Aggregate Business Information Entity comprises the different set of the Basic Business Information Entities than the previously explained Aggregate Business Information Entity named *Automotive_Product. Details*. Finally, in this business context, the Aggregate Core Component named *ProductGroup. Details* and the Association Core Component named *ProductGroup. Member. Product* are completely excluded.

The previous example highlights the importance of the business context for conducting the restrictions of Core Components. This Doctoral Thesis describes our approach (i) to represent these business contextual meanings and (ii) to utilize business contextual knowledge in order to develop new sets of Business Information Entities which are valid in the specified business contexts.

2.4 UN/CEFACT's Core Components Context Definition

In the previous Sections we have stipulated that the content of a UN/CEFACT based business document strictly depends on the business context of its usage. This Section elaborate on the context definition and context segmentations proposed by UN/CEFACT.

2.4.1 UN/CEFACT's Core Components Context Definition Model

UN/CEFACT defines a business context by a set of the context values associated to their corresponding context categories [3, 4]. The underlying context definition model is illustrated in Figure 2.9. A *context value* is an atomic piece of knowledge that represents one aspect of the context (industry, geopolitical region, official constraints, etc.). These values must be taken from the code lists of values previously approved by UN/CEFACT.

A *context category* represents a group of one or more related context values used to express a specific business context. As shown in Figure 2.9, UN/CEFACT recognizes eight types of business context categories. We describe them in the following.

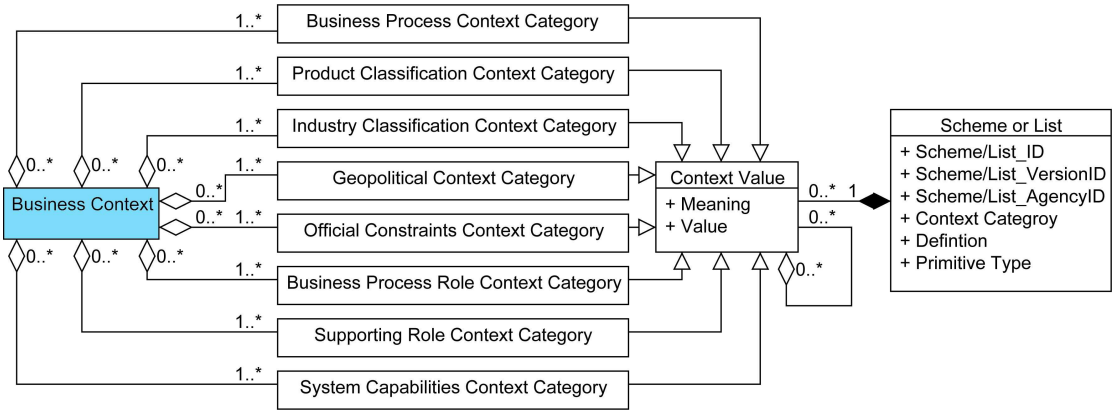


Figure 2.9: UN/CEFACT's Core Components context definition model

- Business Process (Activity) Business Context Category - defines those aspects of the context which are related to the business activity being conducted (goods ordering, goods shipping, etc.);

- Product Classification Business Context Category - defines those aspects of the context which are related to the goods and services being exchanged, handled, paid for, concerned or otherwise manipulated in the business process (consulting service, electronic flight ticket, book, etc.);

- Industry Classification Business Context Category - describe those aspects of the context which are related to the industry or subindustry in which the business process takes place (automotive industry, food industry, milk industry, etc.);

- Geopolitical Business Context Category - provides a description of those aspects of the context which are related to region, nationality or geopolitically based cultural factors (structure of the address, type of the alphabet, local time zone, etc.);

- Official Constraints Business Context Category - describes those aspects of the context which are related to legal and regulatory requirements and similar official categories. This context category consists of two distinct parts: (i) Regulatory and Legislative (unilaterally regulations, such as customs authority regulations and bank transfer regulations), and (ii) Conventions and Treaties (bi- or multilateral agreements, such as Central European Free Trade Agreement [12] and World Trade Organization Agreement on Textiles and Clothing [13]);

- Business Process Role Business Context Category - defines those aspects of the context which are related to an actor or actors that actively participate in the business process (product development, marketing management, etc.);

- Supporting Role Business Context Category - describes those aspects of the context which are related to the parties which are not active participants in the business process but who are interested in the process (third party shipper, business consultant, etc.);
- System Capabilities Business Context Category - defines the limitations of the system or of the class of the systems (date format or address format which can be processed, etc.).

We describe our research using three of these categories which we consider to be most important for the characterization of business context: (i) geopolitical business context category, (ii) industry business context category and (iii) activity business context category. However, the solutions proposed in the following Chapters of this thesis can be easily applied to process the knowledge which originates from other business context categories as well.

2.4.2 Contextualizing UN/CEFACT's Core Components

Every Business Information Entity is valid in its *assigned business context*. This business context does not depend on the business contexts in which other Business Information Entities are valid. Based on the type of the Business Information Entity, its assigned business context either is the same as its overall business context or represents only one of the contexts that contributes to its overall business context.

An *overall business context* is a context in which some Business Information Entity is valid considering the business contexts in which other Business Information Entities from the same electronic business document are valid. Speaking briefly, an overall business context is calculated based either on the assigned business contexts or the overall business contexts of the Business Information Entities which are comprised by the same business document as the target Business Information Entity. These calculations follow the rules which are proposed by UN/CEFACT:

Rule 2.4.1 *Each Basic Business Information Entity has a context assigned to it, and the overall context of the Basic Information Entity is just that assigned context, independent of the context of any other Basic Information Entity.*

Rule 2.4.2 *An Aggregate Business Information Entity does not have a context assigned to it. The overall context of the Aggregate Business Information Entity is the union of the overall contexts of the Basic Information Entities and Association Business Information Entities within the Aggregate Business Information Entity.*

Rule 2.4.3 *Each Association Business Information Entity has a context assigned to it. The overall context of the Association Business Information Entity is the intersection of the context assigned to it and the overall context of the Aggregate Business Information Entity it is associated with.*

Contextualizing UN/CEFACT's Core Components - Example. The applications of the presented rules are illustrated in the example shown in Figure 2.10. The BBIE1 is the Business Information Entity which is valid at design time in the business context of *Switzerland* (assigned

business context), and the BBIE2 is the Business Information Entity which is valid at design time in the business context of *Germany* (assigned business context). The overall business contexts of the introduced Basic Business Information Entities are the same as their corresponding assigned business contexts. (Rule 2.4.1).

Furthermore, as shown in Figure 2.10, the ABIE1 is the Aggregate Business Information Entity which includes the BBIE1 and BBIE2. The ABIE1 is valid in its overall business context which resolves the union of the overall business contexts in which its included BBIE1 and BBIE2 are valid (Rule 2.4.2). Therefore, the overall business context of the ABIE1 is defined by both *Switzerland* and *Germany*.

Finally, the ASBIE1 is the Association Business Information Entity which correlates some ABIE2 with the introduced ABIE1. It is valid in the assigned business context which resolves the *European Union*. The overall business context of this Association Business Information Entity is dependent, and it resolves the intersection of its assigned business context and the overall business contexts in which its associated ABIE1 is valid (Rule 2.4.3). Therefore, as shown in Figure 2.10, the overall business context of the ASBIE1 is defined by *Germany*.

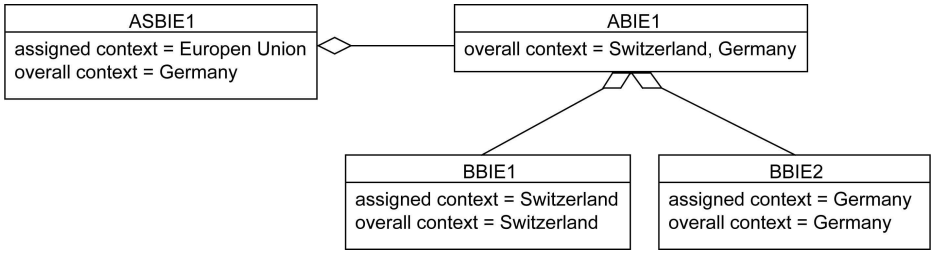


Figure 2.10: Example - UN/CEFACT's rules for contextualizing Core Components

The contextualization rules explained in this Section serve as a basis for our research described in the following Chapters of this thesis.

2.5 UN/CEFACT's XML Naming and Design Rules

Conceptual data model components, such as transfer syntax neutral UN/CEFACT's Core Components, can not be directly processed by IT systems. Therefore, UN/CEFACT has proposed the XML Naming and Design Rules (NDR) [14]. This is a specification which formulates the set of the rules necessary to develop XML schemas and XML schema based documents which conform to the CCTS business document standard. The rules for the transformation of Core Components, Business Information Entities and Data Types into the corresponding XML elements are illustrated in Figure 2.11 and explained in the following.

XML Representation - Aggregate Business Information Entities. An Aggregate Business Information Entity is represented as a type definition by the *xsd:complexType* XML schema component. Additionally, it has to be declared as a global element by the *xsd:element* XML schema component.

XML Representation - Basic Business Information Entities. A Business Information Entity is represented by the local *xsd:element* XML schema component within the *xsd:complexType*

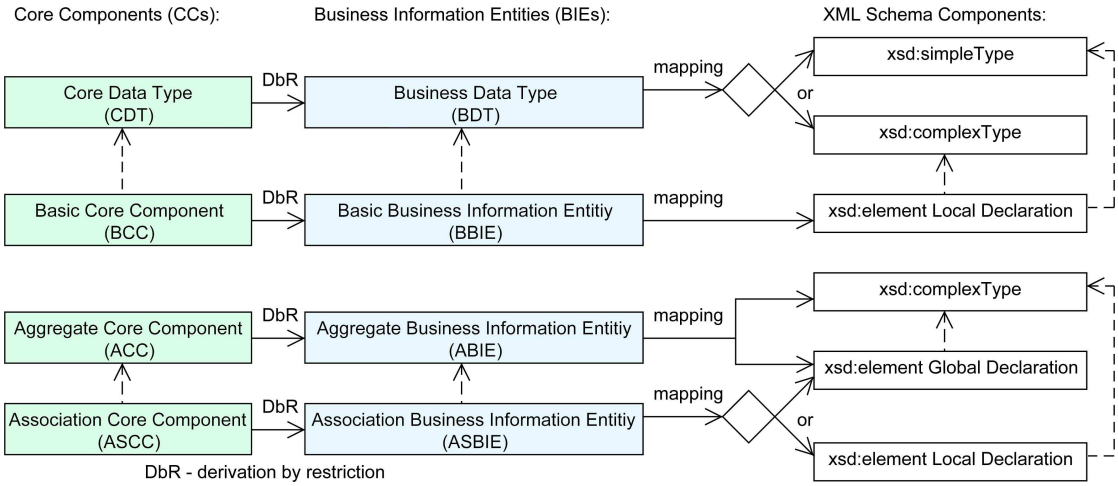


Figure 2.11: Transformation rules, UN/CEFACT's Core Components - XML schema elements

XML schema component. This *xsd:complexType XML* schema component is the XML representation of the including Aggregate Business Information Entity. The *xsd:element* XML schema component belongs to the *xsd:simpleType* or *xsd:complexType* which specify the Business Data Type of the target Basic Information Entity.

XML Representation - Association Business Information Entities. An Association Business Information Entity is represented by either a local or global XML component, depending on its type of aggregation. We describe both representations in the following.

If an Association Business Information Entity specifies a composite aggregation, it is represented by the local *xsd:element* XML component within the *xsd:complexType* XML component. This *xsd:complexType* XML component represents the associating Aggregate Business Information Entity. The local *xsd:element* belongs to the type *xsd:complexType* which specifies the associated Aggregate Business Information Entity.

If an Association Business Information Entity specifies a shared aggregation, it is represented by the global *xsd:element* XML component. This component is declared in the same namespace as the associating Aggregate Business Information Entity. Furthermore, it belongs to the type *xsd:complexType* which specifies the associated Aggregate Business Information Entity.

XML Representation - Business Data Types. A Business Data Type is declared by either an *xsd:simpleType* or *xsd:compextType* XML schema component. In case that the value domain resolved by the Business Data Type can be expressed using some of the XML schema built-in data types, this Business Data Type is presented by the *xsd:simpleType* XML schema component. In contrary, this Business Data Type is presented by the *xsd:complexType* XML schema component.

The implementation of our business context aware Core Components modeling approach processes the electronic business documents which are represented by the UN/CEFACT's NDR specification. The corresponding procedure is explained in Chapter 7 of this thesis.

2.6 UN/CEFACT Business Message Model

UN/CEFACT business message structures are defined using the Business Message Model which is represented in Figure 2.12. Accordingly, each business message structure consists of a single Message Assembly (MA). Each Message Assembly represents the specific business message. It contains (i) one or more Association Message Assembly Components (ASMAs) and (ii) zero or one Standard Business Document Header Component (SBDH).

An Association Message Assembly Component represents the first level Association Business Information Entity. Therefore, it acts as a proxy for a first level Aggregate Business Information Entity in a specific business message. The optional Standard Business Document Header Component contains application specific information unique to the message instance.

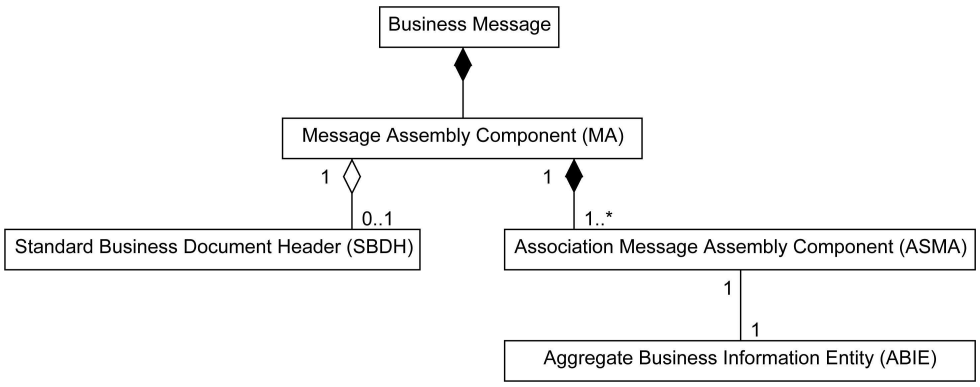


Figure 2.12: UNCEFACT Business Message Model

The electronic business documents processed in the following of this Doctoral Thesis are built upon the introduced UN/CEFACT Business Message Model.

2.7 Final Assessments

The Core Components Technical Specification (CCTS) [3] is a methodology proposed by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). It aims at standardizing the exchange data models which are established between different business partners. Accordingly, every electronic business document is built upon the UN/CEFACT Business Message Model and comprises the set of semantically interoperable Core Components.

Core Components are business context independent data communication models which can be used in any business scenario. It is possible to distinguish between three types of Core Components: (i) Basic Core Components, (ii) Aggregate Core Components and (iii) Association Core Components. A Basic Core Component is a piece of information which is located in a business document. Each Aggregate Core Component represents a collection of Basic Core Components. Relations between Aggregate Core Components are established by Association Core Components. For a better understanding one could – even if not 100% correct – compare

Aggregate Core Components as classes, Basic Core Components as attributes and Association Core Components as associations.

Depending on the context of the current business scenario, Core Components are *restricted* into Business Information Entities. Therefore, a precise definition of business context and an instrument to represent business contexts of arbitrary complexities are the main prerequisites to derive new Business Information Entities from already existing Core Components based communication models. Analogously to the Core Component concept, it is possible to distinguish between three types of Business Information Entities: (i) Basic Business Information Entities, (ii) Aggregate Business Information Entities and (iii) Association Business Information Entities.

Furthermore, transfer syntax neutral Core Components can not be directly processed by IT systems. Therefore, UN/CEFACT has proposed the XML Naming and Design Rules [14]. This is a specification which formulates a set of transformation rules necessary to develop XML schemas and XML schema based documents which conform to CCTS.

UN/CEFACT defines a business context by a set of context values which are associated to their corresponding context categories. A context value is an atomic piece of knowledge that represents one aspect of a business context (geopolitical region, industry, official constraints, etc.). In the following Chapters this Doctoral Thesis specifies the meaning of business context more precisely. Afterwards, it defines business context and describes our business context models. Finally, this thesis develops an approach to utilize business contextual knowledge to (semi-) automatically *derive* the Business Information Entity contents of the missing electronic business documents *by business context aware restriction*.

CHAPTER 3

Business Context

The previous Chapter has described the Core Components based electronic business documents modeling approach. It is defined using the Core Components Technical Specification (CCTS) which is proposed by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). Accordingly, more specific components of electronic business documents are *derived* from more general components *by business context aware restriction*. Therefore, the main prerequisite to develop structures of new business documents is to have an approach to define the business contextual knowledge in which these documents will be valid.

This Chapter summarizes relevant definitions of context and context awareness in different domains. Based on the corresponding survey, we define context and context awareness in the domain of the electronic business documents. These documents are built upon the CCTS business document standard. They are exchanged between business partners when executing inter-organizational business processes. The main conclusions of this phase of our research are discussed in [15, 16].

The remainder of this Chapter is structured as follows. In Section 3.1 we summarize and compare all general context definitions. Afterwards, we explain problems related to context detection and define context awareness. In Section 3.2 we narrow the scope of context to the more specific domain of electronic business documents. Therefore, we define business context and business context awareness. In Section 3.3 we discuss different context modeling techniques, and highlight their benefits and drawbacks. Based on the corresponding outcomes, we present guidelines for modeling the business context in which electronic business documents are valid in Section 3.4. Finally, the concluding remarks of this phase of our research are outlined in Section 3.5.

3.1 Context

The word context originally comes from the Latin language. Its root consists of two words: *con*, which means with or together, and the word *texere*, which means to weave. Accordingly, the original meaning of the word context is join together.

3.1.1 Definitions of Context

There have been many different scientific attempts to accurately define the real meaning of context. Most of them can be grouped under two main tenets: (i) Enumeration of examples and (ii) Choosing synonyms for context.

One of the first scientifically important and widely cited context definition is formed by Schilit and Theimer [17] in 1994. They characterize context by following the enumeration of examples approach, where context is enumerated as location, identities of nearby people and objects, and changes to those objects. Following the same tenet, Brown *et al.* [18] describe context as location, identities of the people around the user, the time of day, season, temperature, etc. Similarly, Ryan *et al.* [19] intend to specify context as the user's location, environment, identity and time. In addition to these definitions, Dey *et al.* [20] refer to context as the user's emotional state, focus on attention, location and orientation, date and time, as well as objects and people in the user's environment. Likewise, Brézillon *et al.* [21, 22] analyze a corpus of 166 definitions of context found in a number of domains and come to the conclusion that context can be derived from anything that is significant in a given moment and potentially including the environment, an item within that environment, a user, or even an observer. Finally, Klemke [23] provides a general classification of different context types and represents each of them in an enumeration: Physical context (location, time), Personal context (interest profiles, user profiles/models, knowledge profiles), Domain/Content based context (knowledge profiles, domain ontology), and Organizational context (structure, progress).

In parallel to the enumeration of examples approach, there are several attempts to interpret context by using adequate synonyms. In this vein, context is commonly defined either by a user or an application environment, or by a user or an application situation. For example, Brown [24] describes context as the element of the user environment that the user's computer knows about. Ward *et al.* [25] interpret context as the state of the application surroundings, while Franklin and Flaschbart [26] interpret it as the user situation. Additionally, Rodden *et al.* [27] understand Context as the application setting. Hull *et al.* [28] upgrade these ideas by including the whole environment, and they represent context as the aspect of the current situation.

Finally, considering the previous definitions and taking into account all their benefits and limitations, Dey and Abowd [6] provide one of the most exploited definition of context: *«Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. »*

The main advantage of this context interpretation is that it simplifies an application developer's problem to list all context enumeration elements for some chosen use case scenario. According to the definition, in every interaction it is possible to distinguish between the following three types of entities: a person (an individual, a group of persons, etc.), a place (a country, a city, an office, etc.) and an object (an interchanged electronic business document, a good transport vehicle, etc.). Any piece of information which can be used to characterize the situation of some of them is considered to be context. Furthermore, according to the same authors, these entities can be described by different attributes which all belong to one of four *primary context categories*: location (where is an entity located), identity (who? - every entity must be uniquely identified), activity (what is occurring in a situation) and time (when is an activity happening).

The primary pieces of context of one entity can be used to find secondary context for the same entity as well as primary context for other related entities. For instance, given a person's identity, we can acquire many pieces of related information such as phone numbers, addresses, email addresses, birth date, occupation, list of children, list of colleagues, etc.

3.1.2 Definition of Context Awareness

Similar to the previous scientific attempts to explain context, a significant effort has been taken in order to define context awareness. Our further analysis shows that terms: reactive [29], adaptive [30], responsive [31], situated [28], environment directed [32] and context sensitive [33] are usually considered to be synonyms for context awareness. In essence, all context awareness definitions belong to one of following two definition categories: (i) Using context or (ii) Adapting to context.

Using context is the more general context awareness definition category. Hull *et al.* [28], and Pascoe [34] follow this direction, and define context aware computing as the ability of computing devices to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices themselves. On the other hand, adapting to context is the more concrete context awareness definition category. Schilit *et al.* [17], Brown *et al.* [18], Davies *et al.* [35], and Ward *et al.* [25] are some of the most important representatives of this approach. They see context aware applications as applications that dynamically change or adapt their behavior based on the context of the application and the user.

Finally, taking into account previously presented understandings, Dey and Abowd [6] propose one of the most applied definitions of context awareness. Accordingly, *«A system is context aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.»*

3.1.3 Sensor and Context Types

In context aware systems pieces of contextual information are provided by sensors. Sensors can be realized either by hardware or by data sources. Indulska and Sutton [36] distinguish three different types of them: (i) Physical sensors, (ii) Virtual sensors and (iii) Logical sensors.

Physical sensors, which are also well known as *hardware sensors*, are the most often used today. Microphones, cameras, thermometers, motion detectors and magnetic field detectors are some of their most frequent representatives. In contrast, *Virtual sensors* are not dependent on hardware devices. They detect pieces of context information from software applications or services. For example, querying an electronic planner can be the source of the current user location, user nearby objects or technical demands. *Logical sensors* are usually composite. They encompass physical and virtual sensors with additional data acquired by web services, databases, tables and other data structures. For example, a logical sensor may provide the user current location by using a web service that looks up IP addresses.

Depending on the sensor type which is used for detection, it is possible to distinguish between two different types of context: (i) External (physical) context and (ii) Internal (logical) context. *External context* is the context which is observed by hardware sensors. The possible representatives are: location, temperature, pressure, weight, noise, touch, color, etc. On the

other hand, *Internal context* is particularly important for the scope of business processes. It is the context which is sensed by logical sensors. Therefore, Internal context is usually formed directly by user or from user interactions (business processes, interchanged electronic business documents, open web pages, task problems, user emotions, etc.).

3.2 Business Context

Context is widely used in pervasive systems, most often in the case when mobile devices apply a user location to conduct different kinds of calculations. However, our research aims at utilizing contextual knowledge in a different domain. It is a domain of the electronic business documents which are assembled from Core Components. These documents are exchanged between business partners when executing inter-organizational business processes.

3.2.1 Definition of Business Context

In the previous Sections we specified context and context awareness in general. Now we will narrow these definitions to the business domain particularly. Starting from our previously represented understandings, and considering Dey and Abowd's [6] main context definition, we define business context (BC) in the following way:

Definition 3.2.1 *Business context (BC) is any information that can be used to characterize the situation of an entity within a scope where business operates. An entity is a person, place, or object that is considered relevant to the execution of a business process in a business environment, including the business process and business environments themselves.*

Similarly to Dey and Abowd's approach, the entities which are introduced by our business context definition can be described by different attributes. Each of these attributes can be grouped into one of the primary business context categories. Thus, in the following of this Doctoral Thesis we define business context as:

$$BC_1 = (ctg_1, ctg_2, ..., ctg_c),$$

where ctg_j is the primary business context category identified by its index j. c specifies the total number of the primary business context categories.

The UN/CEFACT's Core Components Technical Specification (explained in Chapter 2) distinguishes between the following primary business context categories: (i) geopolitical BC category, (ii) industry classification BC category, (iii) business process (activity) BC category, (iv) business process role BC category, (v) supporting role BC category, (vi) product classification BC category, (vii) official constraints BC category and (viii) system capabilities BC category. In this thesis we present an approach that is in principle independent on the context categories. When explaining our approach, we use three of these categories which we consider to be most important for the characterization of business context: (i) geopolitical BC category, (ii) industry BC category and (iii) activity BC category. However, the proposed solutions can be easily applied to process the knowledge which originates from the other primary BC categories

as well. The geopolitical BC category refers to the geographical factors that influence business semantics (e.g. the structure of an address). The industry classification BC category identifies the semantic influences related to the industry or industries of the trading partners (e.g. units of measure). Finally, the activity BC category describes the business activity being conducted (e.g. type of the interaction between business partners).

Therefore, in our research the following holds: $c = 3 \Rightarrow BC_1 = (ctg_1, ctg_2, ctg_3)$. In particular, the chosen primary business context categories serve as a basis for providing contextual metadata on electronic business documents exchanged between business partners when executing inter-organizational business processes. For example, Japan and the Book industry, Austria and the DVD industry, or Canada and the Aircraft industry, can be used to describe the situation of the electronic business documents which are involved within a particular user activity, such as invoicing, ordering and notification of shipment.

3.2.2 Definition of Business Context Awareness

In the previous Section we have stipulated that before some information can be used as context by some system, this system must have the ability to detect context. Likewise, the same holds in the business environment where each business system must be business context aware. Based on our presented understandings and analogously to the general definition of context awareness formed by Dey and Abowd [6], we define business context awareness in the following way:

Definition 3.2.2 *A system is business context aware if it uses business context to provide relevant information and/or services to the business process, where relevancy depends on the task encapsulated by the business process.*

In the typical business environment contextual information is usually formed directly by business partners or by monitoring interactions which are established between them (e.g., from interchanged electronic business documents, database structures and electronic planners). Thereby, according to the previous classification of different sensors presented in Section 3.1.3, virtual and logical sensors are those which are mainly applied to detect business contextual metadata contained by electronic business documents. Thus, from the aspect of the business context aware systems, we more precisely specify business context as *internal (logical) business context*.

3.3 Context Modeling

One of the most detailed classification of different context modeling techniques is done by Strang and Linhoff-Popien [37]. They consider data structures, used to represent and exchange contextual information in the respective system, as main classification criteria. Starting from their most important results and including outcomes achieved by other relevant researches [38–41] we distinguish between the following main context modeling instruments: (i) Key Value Models, (ii) Markup Scheme Models, (iii) Graphical Models, (iv) Logic based Models and (v) Ontology based Models. We elaborate on each of them in the rest of this Section. Furthermore, we show how the contextual information which specifies some specific industry and some specific geopolitical region can be presented by means of these models.

3.3.1 Key Value Models

The *Key Value Model* is a context modeling approach where the simplest data structures are used to represent and exchange contextual information. More precisely, context is described by the list of attributes, where each of them is presented as a *context key* and *context value* pair. These pairs are usually realized as environment variables. Therefore, this context modeling approach is often used to represent capabilities or incapabilities of services in service frameworks, where capability discovery functions involve attribute matching algorithms.

```
Model:
[<ContextKey> Context Value]*
-----------------
Instances:
<EconomicGroup> European Union; <MonetaryGroup> Eurozone; <Currency> Euro; <Continent> Europe;
<Country> Austria; <Industry> Automotive; <Product> Tire; <ProductType> Winter Tire; ...
```

Figure 3.1: Key Value Model - example

The main advantage of the Key Value modeling approach is its simplicity of its implementation. However, it can not be used to model more sophisticated context definitions. Some of the examples of its usage are: Schilit *et al.* [42–44] and Capeus framework [45]. In Figure 3.1 we present our simple example of the Key Value Model where the corresponding instances (*<Economic Group> European Union*, etc.) describe the Automotive industry in Austria.

3.3.2 Markup Scheme Models

Markup Scheme Models represent a context modeling approach where hierarchical data structures, which consist of markup tags with attribute values, are used to represent contextual information. The content of a markup tag is often recursively defined by other markup tags.

In particular, this modeling approach is simple to manage. It can be used to model sophisticated context. An existence of a scheme definition provides a stable foundation for developing validation tools. However, strict hierarchical structures imply lacks of flexibility.

Profiles are typical representatives of this Context modeling instrument. They are usually derived from Standard Generic Markup Language (SGML) [46], the super class of all markup

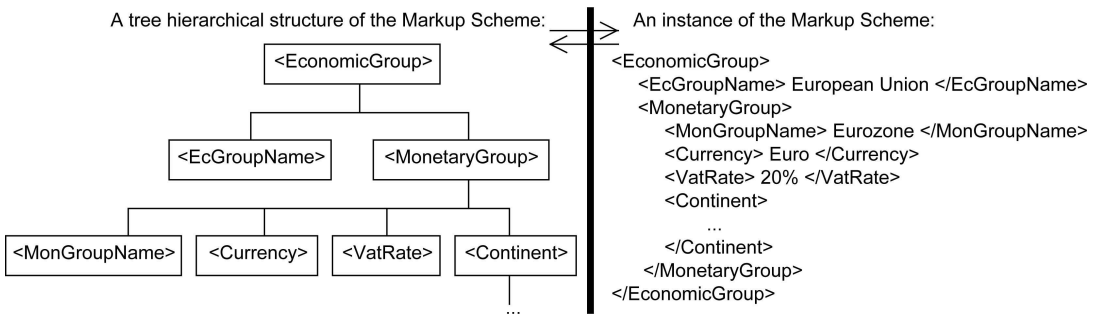


Figure 3.2: Markup Scheme Model - Example

languages. Some examples are: Composite Capabilities/Preference Profile (CC/PP) [47], User Agent Profile (UAProf) [48], Comprehensive Structure Context Profiles (CSCP) [49], Pervasive Profile Description Language (PPDL) [50], Centaurus Capability Markup Language (CCML) [51].

In Figure 3.2 we present our simple example of the Markup Scheme Model and the corresponding instance which describes the Automotive industry in Austria. Accordingly, the context specific markup tags are defined. For instance, *<EconomicGroup>* is the markup tag which encapsulates the markup tags *<EcGroupName>*, *<MonetaryGroup>*, and so forth.

3.3.3 Graphical Models

Graphical Models are used to express relationships between context entities. This is realized by two main methods described in the following. The first method is known as Diagrammatic modeling. It uses the Unified Modeling Language (UML) [52] as the main instrument to model context. UML consists of graphical elements (UML diagrams) which are excellent foundations for context modeling. The corresponding examples are implemented by Sheng and Benatallah [53]. The second commonly used method to develop Graphical Context Models extends the Object Role Modeling (ORM) [54] with contextual information. This procedure is presented by Henricksen *et al.* [55], Halpin [56], Nijssen and Halpin [57], Elmasri and Navathi [58], etc.

In Figure 3.3 we present the Graphical Context Model of our already introduced simple example using UML Class diagrams. Accordingly, a class *Country* is defined. This class is related with the class *AdminArea* through the association of type composition and with the class *Industry* through the association which has the multiplicity 1..*, etc.

Graphical Models are the best for structuring contextual data. The corresponding representations are based on graphical visualizations which can easily be supported by user friendly graphical design tools. Additionally, Graphical Models can be automatically translated into code representations or Entity Relationship (ER) models [59] for databases. However, the lack of reasoning capabilities is the most important drawback of this context modeling approach.

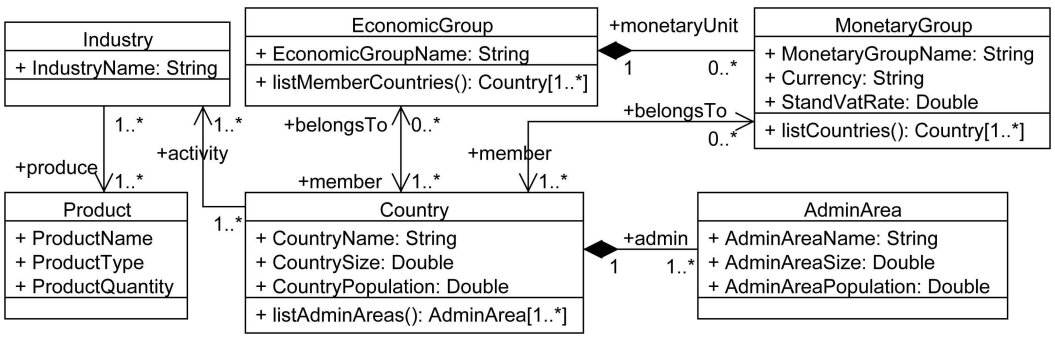


Figure 3.3: Graphical Model - example

3.3.4 Object Oriented Models

Object Oriented Models represent a context modeling approach where context is structured into classes of objects and their relationships. Contrary to the previously described graphical context modeling approach, this strategy uses the complete power of the object oriented paradigm, such as abstraction, encapsulation, polymorphism, re-usability and inheritance. Therefore, different context types are represented by different classes. Consequently, details of context representation are completely hidden.

The general principles of the object oriented modeling are widely used in computer science. Thereby, this context modeling approach is very suitable for the majority of developers. Furthermore, the requirements of distributed context compositions can easily be fulfilled by Object Oriented Models. On the other hand, invisibility caused by encapsulation is the main drawback of the approach. Consequently, the requirements of formalism are undermined. Some cases using the object oriented context modeling approach are: Hydrogen project [60], TEA project [61], Active Object Model in the GUIDE project [62] and Object Oriented Paradigm to model context [63].

In Figure 3.4 we show how the contextual information that correlates some specific industry with some specific geopolitical region can be presented by our simple Object Oriented Context Model. Accordingly, contextual data are accessible only through specially defined class methods, such as *setEconomicGroupName(s: String)* and *getEconomicGroupName(): String*. The

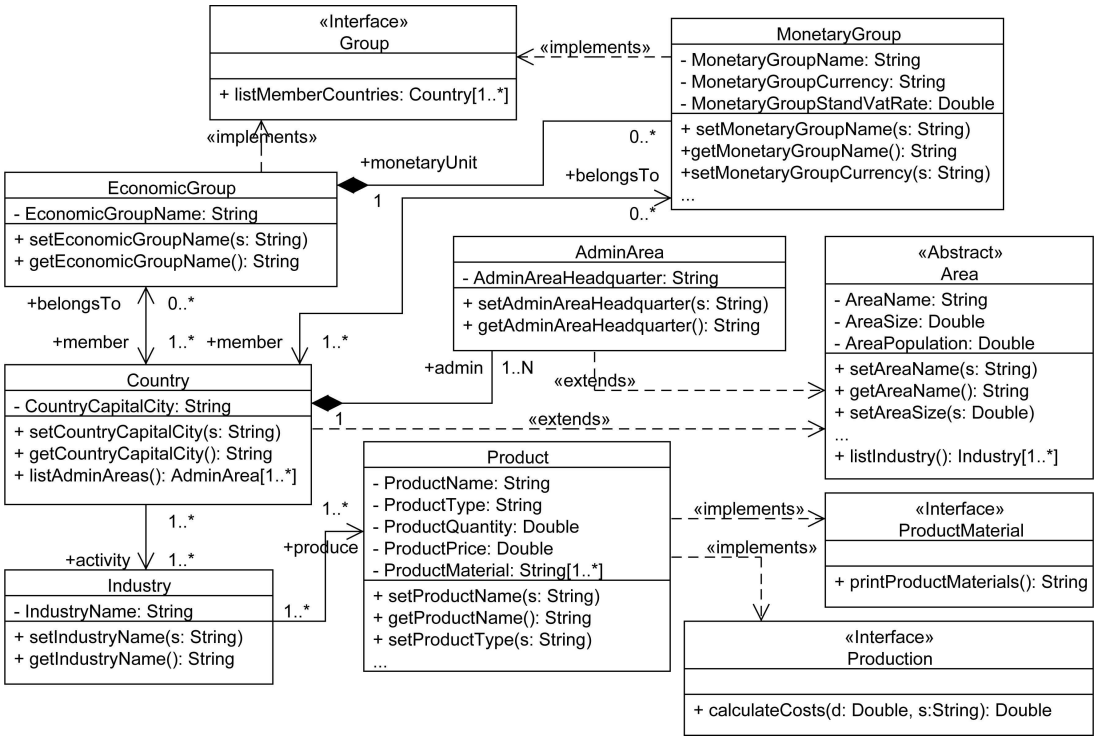


Figure 3.4: Object Oriented Model - example

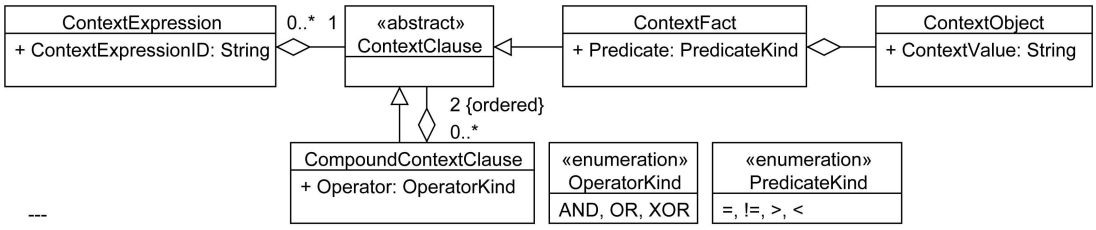
abstract class *Area* is inherited by the classes *Country* and *AdminArea*. Finally, polymorphism is realized through the method *listMemberCountries()* which is implemented by both *Economic-Group* and *MonetaryGroup* classes.

3.3.5 Logic Based Models

Logic based Models define context in terms of formal facts, expressions and rules. In a nutshell, new contextual knowledge can be derived by applying existing rules on already existing facts, logic conditions and knowledge bases. This process is defined as reasoning or inferencing. A derived contextual information is represented as a new fact in a formal way.

Although the high degree of formalism and support for reasoning are considered to be the most important benefits of the logic based context modeling approach, these characteristics also bring significant drawbacks. For instance, it can be very difficult to construct, implement and maintain complex logic systems which should cover a wide range of context. Furthermore, reasoning can be applied only under these demanding systems as a whole. Consequently, partial reasoning and validations are not possible. Therefore, the lack of scalability is a very serious issue of this context modeling technique. Finally, the Logic based Models also show important weaknesses in detection and solving incompleteness, ambiguity, and low quality of information.

Some cases using the logic based context modeling approach are: McCarthy's approach [64], Giunchiglia's approach [65], Extended Situation Theory [66], and Sensed Context Model [67]. In Figure 3.5 we present our simple example of the Logic based Context Model. Accordingly, each context expression consists of a context clause. A context clause can be either a context fact with an assigned predicate or a compound context clause. A compound context clause consists of two ordered context facts connected by an operator. For instance, the expression: $A < B$ denotes that the context in which the entity *A* is valid is the subset of the context in which the entity *B* is valid. The exemplary reasoning rule denotes that some contextual information, such as *Standard Vat Rate*, assigned to the entity *A* is valid in the case of the entity *B* as well.



Reasoning:
KB1: ContextExpressionID: (=A) AND (<B) ~ (A < B), A,B = Context Objects
KB2: ContextualKnowledge: (A_StandVatRate = '20%'), A_StandVatRate, B_StandVatRate = Context Values
---
New knowledge: KB1 AND KB2 = (A < B) AND (A_StandVatRate = '20%') ⟶ (B_StandVatRate = '20%')

Figure 3.5: Logic based Model - example

3.3.6 Ontology Based Models

Ontology based Models represent a context modeling approach where logic and object oriented modeling techniques are combined. Accordingly, context is specified by classes (concepts), individuals (facts) and properties (roles, relationships). In essence, an ontology describes the concepts in some particular domain and the relationships that hold between them. More complex concepts are usually defined by derivation from simpler concepts.

In line with the Logic based Models, the high degree of formalism and support for reasoning are the main advantages of the ontology based context modeling technique. Furthermore, there are many already existing tools for reasoning, known as reasoners [68], which can check whether some concepts and their definitions are or are not consistent. Moreover, these tools can classify all concepts from some ontology into hierarchical structures and maintain them. Some examples of commonly used reasoners are: FaCT++ [69], HermiT [70], Pellet [71] and KAON2 [72].

Likewise, the capabilities of knowledge sharing and knowledge reuse additionally underpin the usability of the approach. Generally speaking, each ontology can be published as Linked Open Data [73] and interrelated with other well known ontology models. Considering the level of richness of the internal structures of these models, it is possible to distinguish between two main types of them: *light-weight* and *heavy-weight* ontologies. Light-weight ontologies describe taxonomies of certain domains, such as business contact managements (vCard) [74], document metadata (Dublin Core) [75, 76], description of projects (DOAP) [73], social networks (FOAF) [77], and products and services for use in eCommerce (UNSPSC) [78]. In contrary, heavy-weight ontologies are more complex and they are not limited only to some specific domain. They model a certain knowledge in *«a deeper way and provide more restrictions on the domain semantics »* [79]. The corresponding examples are: DBpedia [73, 80] and Semantically-Interlinked Online Communities (SIOC) [81, 82].

Finally, the ontology design is supported by many user friendly graphical tools, for instance,

```
<owl:Class rdf:about="http://ontexample#Country">
        <owl:sameAs rdf:resource="http://umbel.org/umbel/sc/Country"/>
        <rdfs:label>Land</rdfs:label>
        <rdfs:label>Država</rdfs:label>
</owl:Class>
<owl:Class rdf:about="http://ontexample#Automotive">
        <rdfs:subClassOf rdf:resource="http://ontexample#Industry"/>
</owl:Class>
<owl:ObjectProperty rdf:about="http://ontexamplel#hasIndustry"/>
<owl:NamedIndividual rdf:about="http://ontexample.owl#Austria">
        <rdf:type rdf:resource="http://ontexample.owl#Country" />
        <hasIndustry rdf:resource="http://ontexample.owl#Automotive"/>
        <owl:sameAs rdf:resource="http://dbpedia.org/page/Austria" />
        <rdfs:label>Österreich</rdfs:label>
        <foaf:homepage rdf:resource="http://www.bundeskanzleramt.at/" />
</owl:NamedIndividual>
-----------------
Reasoning example:
(?A rdfs:subClassOf ?B) AND (?B owl:hasStandVATRate ?C)  ⟶  (?A owl:hasStandVATRate ?C)
```

Figure 3.6: Ontology based Model - example

Protégé [83], Sigma [84], Altova SemanticWorks [85] and Amine [86]. These tools speed up the process of the ontology modeling and additionally ease the ontology management.

In contrary to the highlighted benefits, the ontology based context modeling approach has certain obstacles. For instance, in order to fulfill the requirements that all reasoning conclusions must be computable and finished in finite time, ontologies might not have a mechanism rich enough to define complex contextual data. Additionally, ontology reasoning is a very demanding process with high performance requirements.

Some cases using the ontology based approach to model context are: Context Ontology (CONON) model [87], Context Broker Architecture (CoBrA) model [88], Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) model [89], Context Management Ontology (COMANTO) model [90] and Aspect Scale Context Information (ASC) model [91]. In Figure 3.6 we express the context model of our already described simple example using the OWL DL sublanguage [92–94]. Accordingly, we define classes, such as *Country* and *Automotive*, the object property *hasIndustry*, the instance *Austria*, etc. The knowledge sharing and interconnection with external ontologies are achieved using the OWL DL included property *owl:SameAs* and using direct linking, such as *foaf:name* and *foaf:homepage*.

3.4 Guideline for Business Context Modeling

According to the conclusion presented by Strang and Linnhoff-Popien [37], based on the six requirement parameters that they introduced, ontologies are the most promising approach for modeling context in ubiquitous environments. However, the same authors also underline that this final conclusion does not mean that other context modeling approaches are not applicable.

With respect to the aforementioned context modeling contribution and in correspondence to the already explained advantages and disadvantages of different context models, we can conclude that the uniform context modeling approach does not exist. A context modeling choice depends on the particular domain, and it is tightly related with desired performances and requirements. As explained earlier in this Chapter, in particular situations some types of models show their strengths, while other show their weaknesses. In other situations, it is the other way round. In order to simplify the comparisons between different context modeling approaches, we have underlined the main benefits and shortcomings of the main context modeling techniques. The corresponding results are shown in Figure 3.7.

For instance, the Key Value and Markup Scheme Models are both based on the simple data structures which can be implemented easily with minimal error risks. The existence of the scheme in the Markup Scheme modeling approach underpins the data validations and incompleteness analyses. However, flexibility demands, reasoning capabilities, knowledge sharing and requirements for modeling more complex contextual information are poorly addressed by these context modeling approaches. In contrary, the flexibility and solutions for modeling contexts of arbitrary complexities are feasible by applying the Graphical Models. In case that the concrete structure of the context and its corresponding implementation should be hidden, it is better to apply the object oriented modeling approach. Although the development of Logical Models is a difficult task, these models can address the needs for formalism and reasoning very

Context Model	Data Structures	Data Structure Components	Pros	Cons	Examples
Key Value	Key value pairs	- Context key - Context value	- Simple data struct. - Simplicity of implementation - Low error risk	- Limitation in expressiveness of complex context - Reasoning is not supported	Schilit's approach Capeus framework
Markup Scheme	Markup scheme languages (SGML)	- Profiles	- Simple data struct. - Simplicity of implementation - Expressiveness of complex context - Low error risk - Validation tools - User friendly editors	- Flexibility - Reasoning is not supported	CC/PP UAProf CSCP PPDL CCML
Graphical	Graphs, diagrams (UML, ORM)	- UML diagrams - UML extensions stereotypes - ORM fact types	- Structuring of context - Flexibility - User friendly development tools - Representation code / ER models derivation	- Lack of formalism - Reasoning is not supported	Shengs' approach Henricksen's approach Halpin's approach Nijssen's approach Elmasri's approach
Object Oriented	Object oriented paradigm	- Class - Object - Function (method) - Attribute	- Abstraction - Encapsulation - Polymorphism - Re-usability - Inheritance - Well known paradigm - Suitable for distributed compositions	- Invisibility caused by encapsulation - Lack of formalism - Reasoning is not supported	Hydrogen TEA GUIDE Bouzy's approach
Logic based	Logical axioms	- Fact - Expression - Rule	- High degree of formalism - Reasoning	- Implementation - High error risk - Scalability	UCM approach McCarthy's approach Giunchiglia's approach Akman's approach Gray's approach

Ontology based	Ontologies (RDF, OWL-Lite, OWL-DL, OWL-Full)	- Class (concept)
		- Property (role)
		  - Object prop.
		  - Data prop.
		- Individual (fact)
			- High degree of formalism
			- Reasoning
			- Knowledge sharing
			- Knowledge re-using
			- Interoperability
			- User friendly development tools
				- Limitations in expressiveness
				- Performance issues
					BCOnt
					CONON
					CoBrA
					SOUPA
					COMANTO
					ASC

Figure 3.7: Context models appropriateness

well. However, knowledge sharing, re-use and linking are fulfilled by the ontology based modeling techniques.

3.4.1 Business Context Modeling Choice

The goal of the research described in this Doctoral Thesis is to develop a specification model of the internal business context defined in Section 3.2. Based on the nature of the business context and the previously discussed characteristics of different context models (summarized in Figure 3.7), we have decided to follow (i) the *logic based* context modeling techniques and (ii) the *ontology based* context modeling techniques. The most important reasons for our business context modeling choice are underlined once again in the following.

First, both logic based and ontology based context modeling approaches provide a high degree of formalism and support reasoning capabilities. Second, both approaches are widely accepted by the scientific community and serve as a basis for developing context models in different domains. On the one hand, the scientific community [37, 40, 95, 95–101] shares the opinion that ontologies are the most promising approach for modeling context in ubiquitous environments. On the other hand, the UN/CEFACT Context Model (UCM) [4] is an already existing attempt to develop Logic based Models of context in our domain of electronic business documents. Therefore, our Logic based Business Context Model - described in the following Chapters - builds upon the UCM model.

Additionally, the ontology based context modeling approach provides the already discussed capabilities (such as the automatic classification, reasoning and sharing of business contextual knowledge) which underpin our ontology based business context modeling choice. Furthermore, the existing user-friendly ontology development tools significantly ease the research in this context modeling direction.

3.5 Final Assessments

We define (internal) business context as metadata that specify the attributes (e.g., geopolitical region, industry and business activity) in which a particular electronic business document is or is not valid. Business context awareness represents the capability to sense, process and (partially) re-use this business contextual information.

Each of the introduced business contextual attributes can be grouped into one of the primary business context categories. The solutions described in this thesis are independent on the number of context categories. When explaining our solutions, we use three of these categories which we consider to be most important for the characterization of business context: (i) geopolitical BC category, (ii) industry BC category and (iii) activity BC category. However, the proposed solutions can process the knowledge which originates from additional primary business context categories (e.g., business process role, product classification and system capability business context categories) as well.

Our investigation presented in this Chapter has shown that the uniform context modeling technique does not exist. In practice, the choice of the most proper context modeling approach tightly depends on the domain specific nature of a context. Therefore, our analysis of different context modeling techniques (summarized in Figure 3.7) has directed our research on the business context modeling to the logic based and ontology based context modeling directions.

The work presented in this Chapter provides the following main contributions: (i) the analysis of existing context definitions, (ii) the analysis of existing context awareness definitions, (iii) the definition of (internal) business context, (iv) the definition of business context awareness, (v) the analysis of existing context modeling techniques, and (vi) the modeling guidelines for developing specification business context models.

In the following we develop the Logic based Business Context Model (described in Chapter 4) and the Ontology based Business Context Model (described in Chapter 5). Both models can be used to represent the business context which is defined in this Chapter. Furthermore, we show how the existing electronic business documents (composed from Core Components as

explained in Chapter 2) can be contextualized by means of our business context models. This approach serves as a basis to generate the building blocks of the missing electronic business documents (explained in Chapter 6).

CHAPTER 4

Enhanced UN/CEFACT Business Context Model

The previous Chapter 3 has defined business context of electronic business documents. These documents conform to the Core Components Technical Specification (CCTS) which is proposed by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) (explained in Chapter 2). Considering our investigation on different context modeling approaches, we have concluded that (i) the logic based context modeling techniques and (ii) the ontology based context modeling techniques are more suitable than others for modeling our definition of business context.

This Chapter describes our Logic based Model to represent business context. First, we introduce the already existing UN/CEFACT Business Context Model [4] defined by UN/CEFACT. Afterwards, we explain the shortcomings of the approach and show how we can overcome them by our *Enhanced UN/CEFACT Business Context Model*. Finally, we describe how the electronic business documents can be contextualized by means of our model. The main conclusions of this phase of our research are discussed in [102, 103] and exploited in [104].

The remainder of this Chapter is structured as follows. First, Section 4.1 describes the necessary background on the UN/CEFACT Context Methodology [4] and presents the corresponding UN/CEFACT Business Context Model. In Section 4.2 we elaborate on the main shortcomings of the introduced approach. Afterwards, we define our Enhanced UN/CEFACT Business Context Model and explain its main capabilities. In Section 4.3 we demonstrate the usability of our enhanced model on the concepts defined by CCTS. Finally, the concluding remarks of this phase of our research are outlined in Section 3.5.

Our concurrent Ontology based Business Context Model is explained in Chapter 5. In order to ease the understandings and comparisons between our two approaches, we try, when it is possible, to describe both models by introducing the common terminology and equivalent examples.

4.1 UN/CEFACT Business Context Model

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) has proposed the Context Methodology (UCM) [4] to manage representations and applications of business context (BC), especially under the scope of the UN/CEFACT's Core Components based standardization effort (explained in Chapter 2). This methodology is underpinned by two main pillars: (i) the UCM business context graph (UCM BCG) and (ii) the UCM business context expression.

4.1.1 UCM Business Context Graph

The *UCM business context graph (UCM BCG)* is defined as a directed acyclic graph (DAG). The corresponding metamodel is presented in Figure 4.1. Accordingly, every BCG is uniquely identified by its *BCG_ID*. Its current version is specified by *BCG_Version*. The organization which has developed the graph is represented by *BCG_AgencyID*. The name and description of the BCG are not mandatory. However, these data can be provided as *BCG_Name* and *BCG_Description*, respectively.

Furthermore, as shown in the metamodel in Figure 4.1, every BCG must have at least one BC node (uniquely identified by its *BCNode_ID*) and zero or more BC edges. Each *BC node* denotes some specific BC value. A *BC value* is an atomic piece of knowledge that represents one aspect of the business context (e.g., geopolitical region, industry or activity). It must be taken from the code lists of values previously approved by UN/CEFACT. A *BC edge* denotes the directed relationship between two BC nodes. A BC edge that originates from a particular BC node is an outgoing BC edge of that node, and this node is defined as a source (parent) BC node. Analogously, a BC edge that points to a particular BC node is an incoming BC edge of that node, and this node is defined as a target (child) BC node. In a nutshell, an edge narrows the scope of a BC value which is assigned to its source BC node to a BC value which is assigned to its target BC node.

Each BCG must have at least one *root BC node*. It is a node that has only outgoing BC edges, and it can not be a target BC node of any BC edge. In practice, each root BC node represents

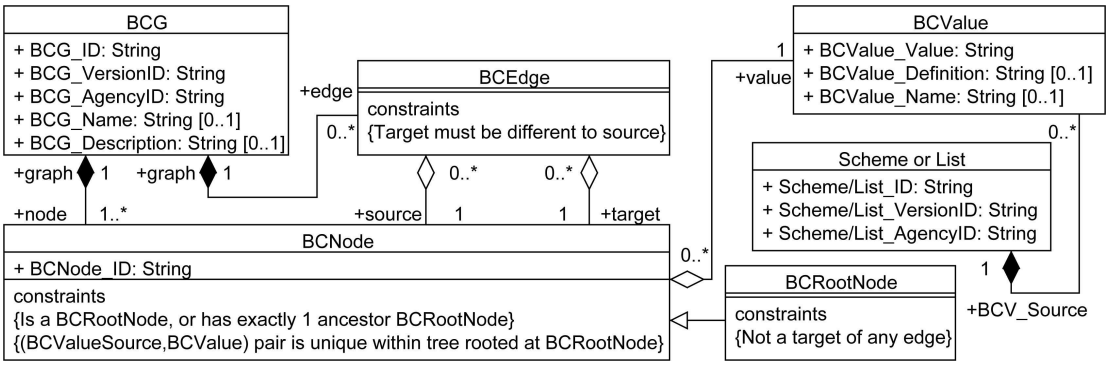


Figure 4.1: UCM business context graph metamodel

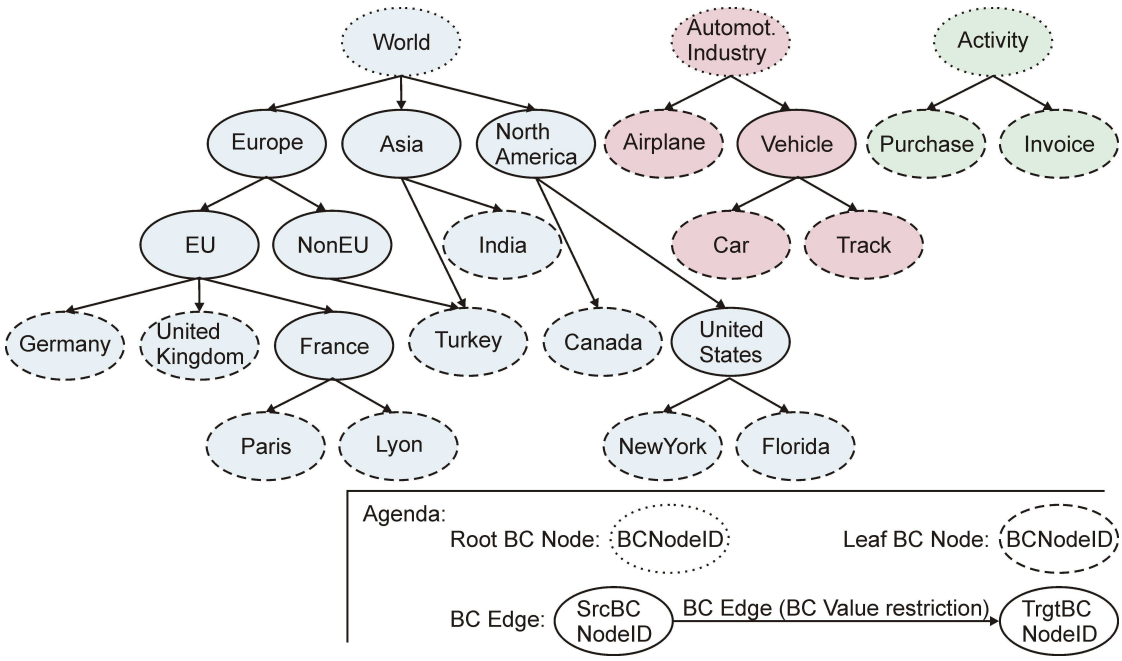


Figure 4.2: Excerpt of the UCM business context graph

the beginning of a new categorized set of related BC values. As stipulated in Chapter 3.2, when explaining our research, we consider the geopolitical region, industry and activity categorized sets of BC values. However, the solutions proposed in the following of this thesis can be easily extended to process the knowledge which originates from other business context categories as well.

An excerpt of the BCG is illustrated in the example in Figure 4.2. A BC node is represented by an ellipse which is uniquely identified by a text label. A restriction of the BC value is shown as the directed edge which connects the corresponding BC nodes. For instance, the node marked *AutomotiveIndustry* represents a business context in the Automotive industry. Furthermore, the node marked *Airplane* represents a restriction of the *AutomotiveIndustry* node. This is expressed by the directed BC edge between the aforementioned nodes.

4.1.2 UCM Business Context Expression

A *BC expression* identifies a set of BC nodes from the BCG which can be resolved to concrete BC values. These combined BC values represent a specific business context. The underlying metamodel, based on [3], is presented in Figure 4.3. Accordingly, a *BC clause* is part of a particular BC expression. Furthermore, there are two types of BC clauses: (i) Simple BC clauses and (ii) Compound BC clauses. We explain them in the following.

A *Simple BC clause* is the clause which has only one predicate assigned to a BC value. The corresponding lexical notation is presented by the schema:

$$<SimpleBCClause>=<predicate>\sqcup<BCNodeID>.$$

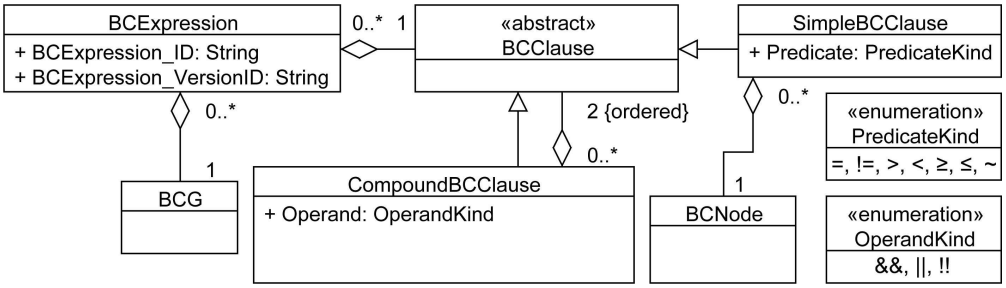


Figure 4.3: UCM business context expression metamodel

UCM provides the list of the following predicates: (i) Equal to ($=$), (ii) Not equal to ($! =$), (iii) Greater than ($>$), (iv) Less than ($<$), (v) Greater than or equal to ($\geq$), (vi) Less than or equal to ($\leq$), and (vii) Less than greater than or equal to ($\sim$). The predicate $\geq$ identifies the BC nodes from the BCG which are predecessors of the specified BC node including the specified BC node itself. The predicate $\sim$ identifies the BC nodes from the BCG which are predecessors of or successors of the specified BC node including the specified BC node itself. The other predicates can be described analogously. For instance, the BC expression ($< Europe$) applied on the excerpt of the BCG presented in Figure 4.2 identifies the following set of BC nodes: *EU*, *Germany*, *UnitedKingdom*, *France*, *Paris Lyon*, *NonEU* and *Turkey*.

A *Compound BC clause* consists of two ordered BC clauses connected by an operand. The corresponding lexical notation is presented by the following schema:

$$<CompoundBCClause>= (<SimpleBCClause|CompoundBCClause>)\sqcup$$
$$<operand> \sqcup (<SimpleBCCluase|CompoundBCClause>).$$

The list of the allowed operands is: (i) Intersection ($\&\&$), (ii) Union ($||$), and (iii) Exclusion ($!!$). For instance, the operand $\&\&$ identifies the BC nodes from the BCG which are resolved by both related BC clauses. The other operands can be described analogously. However, currently, UCM does not provide an unambiguous formal definitions of its operands. Therefore, in Section 4.2, we show how this gap in the standard UCM approach can be addressed.

4.2 Enhanced UCM Business Context Model

As part of our research we encountered several shortcomings of the standard UCM approach which significantly undermine its application. Therefore, in the following we describe these issues more precisely and explain how they can be overcome using the *Enhanced UN/CEFACT Business Context Model (E-UCM)*. E-UCM represents our enhancement of the original UCM approach based on (i) the extension of the centralized UCM BCG approach, and on (ii) the formal definition of the UCM operands.

4.2.1 Enhanced UCM Business Context Graph

As highlighted in the previous Section, the UCM BCG is a centralized, hierarchical, directed, acyclic graph (DAG) which reflects the business context in which some particular electronic business document is valid. Therefore, it must comprise the complete geopolitical organization, industry classification, and all possible user activities. Hence, this graph structure is bewildering complex and usually consists of an overwhelming number of BC nodes. For example, only the industry classification domain of the UCM BCG, based on the International Standard Industrial Classification (ISIC) [105], covers more than 760 BC nodes. At the same time, the geopolitical organization of the same graph contains at least as many nodes as the industry classification. Furthermore, the maximal number of potential edges in every DAG depends on its total number of nodes and can be calculated by the following formula:

$$\max(e) = \frac{1}{2}n(n-1) = f(n^2) \sim O(n^2), \tag{4.1}$$

where *n* represents a total number of nodes, and *e* is a total number of edges. Thus, the number of possible edges in the UCM BCG is expressed by millions.

According to graph theory, memory and time complexities of the graph management operations strictly depend on the total number of nodes and edges [106]. Thereby, the representation of business context in the form of the UCM BCG which contains thousands of nodes and millions of possible edges is a significant shortcoming of UCM. This causes the following negative consequences. First, the construction and initialization of a UCM BCG is a laborious and time-consuming task. Second, it is very difficult to efficiently maintain this graph structure. For example, frequent management operations applied on a huge amount of BC nodes, such as search, remove, or include, are time and memory demanding. Finally, an overloaded system based on a centralized UCM BCG is error prone and shows poor scalability.

Conceptual Solution. In contrary to the standard UCM approach, our research shows that it is not necessary to have a complete BC blueprint assembled by thousands of nodes and millions of possible edges. More precisely, the standard UCM BCG covers not only relevant but also a significant amount of superfluous elements which can not be used in the current business scenario. For example, the business document $BDoc_1$, created by an inter-organizational business process p_1, is relevant in the scope of the vehicle production in the European Union. The corresponding business context can be described by only two BC nodes including one for reflecting the industry, and one for representing the geopolitical region. However, the underlying centralized UCM BCG comprises not only these but thousands of additional BC nodes as well. For example, the BC nodes which refer to the Paper Industry of New Zealand represent only a smaller subset of these superfluous elements. Thereby, in practice, it is very probable that these irrelevant pieces of contextual information presented by thousands of BC nodes will not be considered during any kind of customization of the $BDoc_1$. Therefore, exclusion of the superfluous nodes and edges from the graph will not ruin the relevancy of the graph in the particular business process (p_1 in our example) and will not bring any additional undesired consequences. This is the core of our E-UCM BCG approach described in the following.

The E-UCM BCG approach represents the transition from the centralized UCM BCG approach to a new, decentralized (distributed) approach. Accordingly, we define a decentralized

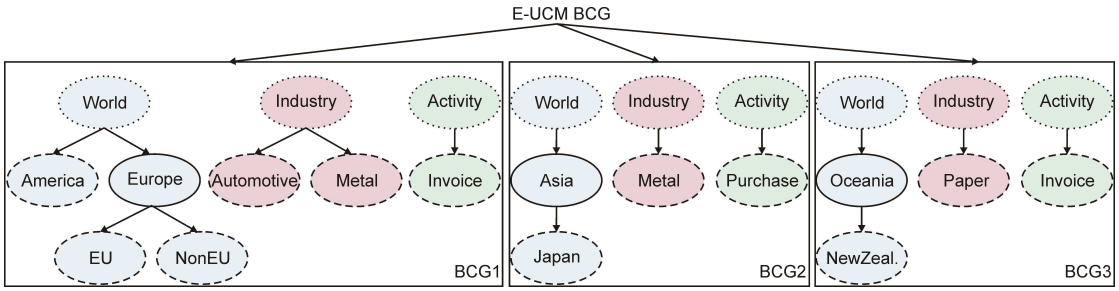


Figure 4.4: Decentralized enhanced UCM business context graph

E-UCM BCG as the subgraph of the centralized UCM BCG. This subgraph comprises only the geopolitical, industry and activity subdomains of business context which are relevant to the scope where a particular inter-organizational business process operates.

Generally speaking, inter-organizational business processes are agile, adaptable and prone to changes. It is essential that our conceptual solution addresses these issues by establishing the capability to dynamically combine different decentralized BC subgraphs into a unique graph. In other words, it is possible to manipulate with the decentralized BC subgraphs based on the current business needs. For example, a company in the Paper industry located in New Zealand is considered. This company extends its business to the Glass industry in Japan and requires additional electronic business documents for new business operations. Consequently, the business environment is changed and these new contextual meanings must be contained by the BCG. Having our decentralized approach at hand, new relevant subgraphs can be dynamically embedded to the existing BCG structure. Furthermore, it is also possible that some business context, which was considered to be relevant for some specific business scenario, can lose its relevancy. In this case, the corresponding subgraphs can be dynamically excluded from the E-UCM BCG.

The example of the E-UCM BCG is presented in Figure 4.4. For instance, its subgraph BCG_1 encapsulates the contextual information related to the previously introduced business document $BDoc_1$. The conceptual implementation of the E-UCM BCG approach is described in the following.

E-UCM Metamodel, Enhancement I. We present our E-UCM metamodel in Figure 4.5. It is the extension of the original UCM BC expression metamodel (Figure 4.3). The proposed E-UCM BCG conceptual solution is implemented by an aggregation which connects BCG entities (Figure 4.5, Mark 1). This new relation enables the union of an indefinite number of BC subgraphs into the unique BCG. Thus, the E-UCM BCG is less ramified than the original UCM BCG. Furthermore, we provide an instrument for its incremental growth and decremental reduction.

Our enhancement of UCM brings the meaningful consequences in the application of this methodology. In a nutshell, the total number of nodes and edges in BCG is significantly reduced. Thereby, decentralized BCG uses less memory. Furthermore, this automatically eases graph initialization and decreases complexities of the maintenance operations. Moreover, in order to describe a new target business process, it is now possible to re-use already existing subgraphs and to rebuild them into decentralized BCG structures. Hence, the levels of interoperability,

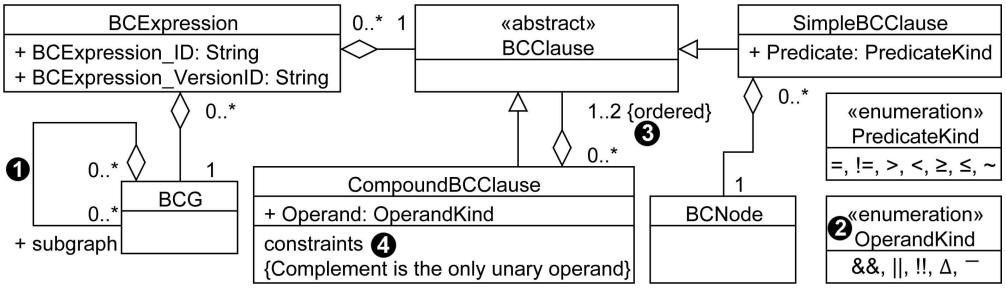


Figure 4.5: Enhanced UCM business context expression metamodel

scalability and flexibility are significantly increased.

Finally, we now introduce the *restriction weight (rw)* of the business contextual derivation. This measure represents the minimal number of the BC edges (restriction steps) between two BC values which are resolved from two related BC nodes defined by the E-UCM BCG. For instance, let us consider the BCG_1 illustrated in Figure 4.4. The BC node denoted by *World* is directly connected to the BC node denoted by *Europe*. Furthermore, the BC node denoted by *Europe* is directly connected to the BC node denoted by *EU*. Hence, the following holds: $rw(World, Europe) = 1$, $rw(Europe, EU) = 1$, and $rw(World, EU) = 2$. The application of the *restriction weight* is meaningful for determining the type of the business context matching (described in Chapter 6).

4.2.2 Enhanced UCM Business Context Expression

In Section 4.1.2 we have presented the UCM operands which have not been formally defined yet. The corresponding standard UCM definitions are incomplete and inconsistent. However, as it is shown in Chapter 6, a formal specification of these operands is an essential prerequisite to model business context and to utilize the contained contextual knowledge. Thereby, (i) we have enhanced the UCM approach by providing the formal definitions of the existing operands, and (ii) we have enriched the expressiveness of business context by introducing two additional operands (Symmetric Exclusion and Complement). These extensions of UCM are developed by utilizing the corresponding concepts introduced by set theory [107].

In the following we denote BC clauses applied under some E-UCM BCG by capital letters $(A, B, C, ...)$ and resolved BC nodes by small letters $(a, b, c, ...)$. The notation $a \in A$ identifies that the BC node a is resolved from (belongs to) the BC clause A. The symbol $\emptyset$ is used for the specification of an empty BC clause. It is the BC clause which does not resolve any BC node.

Identity, Cardinality and Subset. The Identity, Cardinality and Subset of BC clauses are defined in the following. These elements are necessary for the formal expression of the UCM operands.

Definition 4.2.1 *Two BC clauses are identical iff they resolve to exactly the same BC nodes. The number of resolved BC nodes from a BC clause A is called the cardinality of A, represented by: $\mid A \mid$. It is always a natural number.*

$$A = B \overset{def}{\Longleftrightarrow} (\forall x)(x \in A \Leftrightarrow x \in B)$$

Rule 4.2.2 *For BC clauses A, B and C the following Identity rules hold:*
1) $A = A$
2) $A = B \Rightarrow B = A$
3) $A = B \wedge B = C \Rightarrow A = C$

Definition 4.2.3 *A BC clause A is a subset of a BC clause B, represented by $A \subseteq B$, iff every BC node which is resolved from A is also resolved from B. B is in that case a superset of (or includes) A. We denote the opposite by: $A \nsubseteq B$.*

$$A \subseteq B \overset{def}{\Longleftrightarrow} (\forall x)(x \in A \Rightarrow x \in B)$$

A BC clause A is a proper (strict) subset of B, represented by $A \subset B$, iff the BC clauses A and B are not identical and the BC clause A is a subset of the BC clause B. B is in that case the proper (strict) superset of A. We denote the opposite as: $A \not\subset B$.

$$A \subset B \overset{def}{\Longleftrightarrow} (A \neq B \wedge A \subseteq B)$$

Rule 4.2.4 *For BC clauses A, B and C the following Superset rules hold:*
1) $A \subseteq A$
2) $A \subseteq B \wedge B \subseteq A \Rightarrow A = B$
3) $A \subseteq B \wedge B \subseteq C \Rightarrow A \subseteq C$
4) $A \subset B \wedge B \subset C \Rightarrow A \subset C$

Formal Definitions of Operands. In the following we formally define the existing UCM operands (Union, Intersection and Exclusion).

Definition 4.2.5 *The union of BC clauses A and B, represented by $A || B$, is the set of BC nodes which are resolved from A, or B, or from both.*

$$A || B \overset{def}{\Longleftrightarrow} \{x \mid x \in A \vee x \in B\}$$

Rule 4.2.6 *For BC clauses A, B and C the following Union rules hold:*
1) $A || B = B || A$
2) $(A || B) || C = A || (B || C)$
3) $A || A = A$
4) $A || \emptyset = A$
5) $A \subseteq A || B, B \subseteq A || B$
6) $A \subseteq B \Rightarrow A || B = B$

Definition 4.2.7 *The intersection of BC clauses A and B, represented by $A \,\&\&\, B$, is the set of BC nodes which are resolved from both A and B.*

$$A \,\&\&\, B \overset{def}{\Longleftrightarrow} \{x \mid x \in A \wedge x \in B\}$$

If $A \,\&\&\, B = \emptyset$, then A and B are disjoint BC clauses.

Rule 4.2.8 *For BC clauses A, B and C the following Intersection rules hold:*

1) $A \,\&\&\, B = B \,\&\&\, A$
2) $(A \,\&\&\, B) \,\&\&\, C = A \,\&\&\, (B \,\&\&\, C)$
3) $A \,\&\&\, A = A$
4) $A \,\&\&\, \emptyset = \emptyset$
5) $A \,\&\&\, B \subseteq A, A \,\&\&\, B \subseteq B$
6) $A \subseteq B \Rightarrow A \,\&\&\, B = A$
7) $A \,||\, (B \,\&\&\, C) = (A \,||\, B) \,\&\&\, (A \,||\, C)$
8) $A \,\&\&\, (B \,||\, C) = (A \,\&\&\, B) \,||\, (A \,\&\&\, C)$

Definition 4.2.9 *The exclusion of BC clauses A and B, represented by $A \,!!\, B$, is the set of BC nodes which are resolved from A and are not resolved from B. The exclusion of BC clauses A and B is also referred to as the relative complement of B in A.*

$$A \,!!\, B \overset{def}{\Longleftrightarrow} \{x \mid x \in A \wedge x \notin B\}$$

Rule 4.2.10 *For BC clauses A, B and C the following Exclusion rules hold:*

1) $A \,!!\, B \subseteq A, A \,!!\, \emptyset = A$
2) $A \subseteq B \Leftrightarrow A \,!!\, B = \emptyset$
3) $A \neq B \Rightarrow A \,!!\, B \neq B \,!!\, A$
4) $C \,!!\, (C \,!!\, A) \subseteq A$
5) $C \,!!\, (A \,||\, B) = (C \,!!\, A) \,\&\&\, (C \,!!\, B)$
6) $C \,!!\, (A \,\&\&\, B) = (C \,!!\, A) \,||\, (C \,!!\, B)$

New E-UCM Operands. In order to increase the level of business context expressiveness, we formally define two additional operands, namely Symmetric Exclusion and Complement. Furthermore, negation (complement) belongs to the group of the fundamental operations introduced and widely exploited by computational logic. Therefore, the lack of the Complement operand significantly limits the reasoning capabilities of UCM, which, as it will be shown in Chapter 6, can be successfully applied for deriving new knowledge from the already existing contextual knowledge.

Definition 4.2.11 *The Symmetric Exclusion of BC clauses A and B, represented by $A \triangle B$, is the set of all BC nodes which are resolved from clauses A or B and are not resolved from both A and B.*

$$A \triangle B \overset{def}{\Longleftrightarrow} (A \,!!\, B) \,||\, (B \,!!\, A)$$

Rule 4.2.12 *For BC clauses A, B and C the following Symmetric Exclusion rules hold:*

1) $A \triangle A = \emptyset$
2) $A \triangle \emptyset = A$
3) $A \triangle B = B \triangle A$
4) $(A \triangle B) \triangle C = A \triangle (B \triangle C)$
5) $A \,\&\&\, (B \triangle C) = (A \,\&\&\, B) \triangle (A \,||\, C)$

Definition 4.2.13 *The Complement operation of a BC clause A, represented by $\overline{A}$, is the set of all BC nodes which belong to the BCG and are not resolved from A. In contrary to the other E-UCM operands, complement is a unary operand.*

$$\overline{A} \overset{def}{\Longleftrightarrow} \{x \mid x \notin A\}$$

Rule 4.2.14 *If A and B are BC clauses applied on the BCG, the following Complement rules hold:*

1) $\overline{\overline{A}} = A$
2) $A \subseteq B \Leftrightarrow \overline{B} \subseteq \overline{A}$
3) $\overline{A \,||\, B} = \overline{A} \,\&\&\, \overline{B}$
4) $\overline{A \,\&\&\, B} = \overline{A} \,||\, \overline{B}$

E-UCM Metamodel, Enhancement II. The main prerequisites to model and to exploit business context are the formal definition of the E-UCM operands, unconditional expressiveness of the arbitrary business context, and foundations for reasoning. We implement these tenets by extending the standard UCM BC expression metamodel, as illustrated in Figure 4.5. Compared to the original metamodel, the operands Symmetric Exclusion and Complement have been added (Figure 4.5, Mark 2). The complete explanations of the E-UCM predicates and E-UCM operands are summarized in Figure 4.6 and Figure4.7, respectively.

As explained in Section 4.1.2, UCM defines a Compound BC clause as a BC clause which consists of two ordered BC clauses connected by a UCM operand. Therefore, the standard UCM operands are only binary. However, we have enhanced the methodology by the additional unary operand Complement. Thus, a Compound BC clause can now consist of two ordered BC clauses connected by the binary operand, or it can consist of only one BC clause with the assigned Complement operand.

We have implemented the described requirements by changing the multiplicity of 2 to 1..2 of the aggregation which relates the Compound BC clause and the BC clause in the UCM BC

Syntax	Name	Semantics
=	Equal to	Resolves the specified BC node from the E-UCM BCG
!=	Not equal to	Resolves all BC nodes which belong to the same primary BC context category as the specified BC node from the corresponding E-UCM BC Subgraph except the specified BC node
>	Greater than	Resolves the predecessor BC nodes of the specified BC node from the E-UCM BCG
<	Less than	Resolves the successor BC nodes of the specified BC node from the E-UCM BCG
≥	Greater than or equal to	Resolves the predecessor BC nodes of the specified BC node from the E-UCM BCG including the specified BC node itself
≤	Less than or equal to	Resolves the successor BC nodes of the specified BC node from the E-UCM BCG including the specified BC node itself
~	Less than greater than or equal to	Resolves the predecessor and successor BC nodes of the specified BC node from the E-UCM BCG including the specified BC node itself

Figure 4.6: Enhanced UCM predicates

Syntax	Name	Semantics
||	Union	(A || B) resolves the BC nodes from the E-UCM BCG which are resolved from the left hand side BC clause (A), or from the right hand side BC clause (B), or from both.
&&	Intersection	(A && B) resolves the BC nodes from the E-UCM BCG which are resolved from both BC clauses (A and B).
!!	Exclusion	(A !! B) resolves the BC nodes from the E-UCM BCG which are resolved from the left hand side BC clause (A) and are not resolved from the right hand side BC clause (B).
Δ	Symmetric Exclusion	(A Δ B) resolves the BC nodes from the E-UCM BCG which are resolved from the left hand side BC clause (A) or from the right hand side BC clause (B) and are not resolved from both of these BC clauses.
-	Complement	($\overline{A}$) resolves the BC nodes from the E-UCM BCG which are not resolved from the specified BC clause (A) and which belong to the same primary BC category as the BC nodes resolved from A.

Figure 4.7: Enhanced UCM operands

expression metamodel (Figure 4.5, Mark 3). Furthermore, as shown in Figure 4.5, Mark 4, we have introduced the new constraint that the multiplicity of 1 is only valid for the unary operand. The multiplicity of 2 is valid in case that two ordered BC clauses are connected by a binary operand. In both situations, a BC clause can be either a Simple BC clause or a new, recursive Compound BC clause. Thus, our enhanced metamodel preserves a mechanism to express business context of an arbitrary complexity.

In correspondence with the described extensions, a new lexical notation of a Compound BC clause must follow the schema:

$$<CompoundBCClause>= [<unaryOperand> \sqcup (<SimpleBCClause| CompoundBCClause>)] \mid [(<SimpleBCClause|CompoundBCClause>) \sqcup <binaryOperand> \sqcup (<SimpleBCClause|CompoundBCClause>)].$$

The examples of the E-UCM expressions are presented in Figure 4.8. The underlying E-UCM BCG is illustrated in Figure 4.4.

	BC Clause	Type	Resolved BC Nodes
1.	(= Europe)	Simple	Europe
2.	(!= Europe)	Simple	BCG1_World, America, EU, NonEU
3.	(> Automotive)	Simple	BCG1_Industry
4.	(< BCG3_World)	Simple	Oceania, NewZeal.
5.	(≥ Paper)	Simple	Paper, BCG3_Industry
6.	(≤ Asia)	Simple	Asia, Japan
7.	(~ Europe)	Simple	BCG1_World, Europe, EU, NonEU
8.	(> EU) || (= America)	Compound	Europe, BCG1_World, America
9.	(< BCG1_World) && (≥ EU)	Compound	EU, Europe
10.	(≤ BCG3_Industry) !! (!= Paper)	Compound	Paper
11.	(< Europe) Δ (≥ NonEU)	Compound	EU, Europe, BCG1_World
12.	$\overline{(\geq \text{BCG1\_Metal})}$	Compound	Automotive, BCG2_Industry, BCG2_Metal, BCG3_Industry, Paper
13.	$\overline{(((\leq \text{Asia}) \text{ !! } (= \text{Japan}))}$ && ((< BCG1_World) || (!= Oceania)))	Compound	America, Europe, EU, NonEU, BCG3_World, NewZeal.
14.	((≥ EU) && (< BCG1_World)) || (≥ Paper) || (= Purchase)	Compound	Europe, EU, Paper, BCG3_Industry, Purchase

Figure 4.8: Examples - enhanced UCM business context expression

Finally, it is important that the original UCM BC expression metamodel completely conforms to our enhanced version. In other words, the already existing solutions developed in correspondence to the original UCM metamodel are valid and directly applicable in regards to the E-UCM metamodel.

4.2.3 Business Context Reasoning

In contrary to the standard UCM approach, the E-UCM model provides the foundations for business context reasoning. In a nutshell, it is possible to distinguish between two main reasoning

tenets: (i) *DAG based* reasoning and (ii) *rule based* reasoning. We elaborate on each of them in the following.

The core of the first reasoning tenet is the organization of business context in the form of the Directed Acyclic Graph (DAG), as described in Section 4.1.1. Accordingly, an edge in the E-UCM BCG restricts the business context assigned to its source node to the business context assigned to its target node. Thereby, not only the contextual knowledge that originates from the particular BC node, but also the knowledge indicated by its children nodes could be utilized for the characterization of the particular electronic business document.

The core of the second reasoning tenet is the existence of the reasoning rules explicitly provided by a user. Accordingly, a new knowledge can be derived from the existing contextual knowledge by following the reasoning rules. A business context is represented by an E-UCM BC expression, as explained in Section 4.2.2. Relations between different BC expressions can be formulated using the First-order logic syntax [108]. An example of the reasoning rule is:

$$(\forall BC_x)\ ((= BC_x) \subset (\leq BC_{France})) \Longrightarrow ((= BC_x) \subset (\leq BC_{Germany})).$$

In respect to this rule, the information specified by BC_x which holds in the geopolitical region denoted by France holds in the geopolitical region denoted by Germany as well.

The capabilities of business context reasoning can be exploited to model the structures of new electronic business documents by learning from existing document models. The corresponding approach is explained in Chapter 6.

4.3 Contextualizing UN/CEFACT's Core Components

In this Section we explain how electronic business documents based on CCTS can be contextualized by means of the E-UCM business context model. The used generic Core Components are introduced in our previous example described in Figure 2.8 in Chapter 2.3.4. For reasons of simplicity, in the following example we discard the activity BC category and consider only the location and industry BC categories. The corresponding BC nodes are organized in the form of the decentralized E-UCM BCG shown in Figure 4.4. The particular BC values are specified using the enhanced BC expressions, as described in Section 4.1.2.

Contextualizing Basic Core Components. As illustrated in Figure 4.9, Mark 1, two Business Information Entities (BBIE1 and BBIE2) are given. The BBIE1 is a piece of information which refers to the type of a tire valid only in the countries which are members of the European Union. The BBIE2 is a piece of information which refers to the size of a tire valid in Japan. Thus, the BBIE1 has the assigned business context which can be expressed as: $(\leq EU)\ ||\ (\sim Automotive)$ and the BBIE2 has the assigned business context which can be expressed as: $(\sim Japan)\ ||\ (\sim Automotive)$. As explained in Chapter 2.4.2, these assigned business contexts also represent the overall business contexts of the corresponding Basic Business Information Entities.

Contextualizing Aggregate Core Components. In the next step, Figure 4.9, Mark 2, the BBIE1 and the BBIE2 are covered by the Aggregate Business Information Entity denoted by ABIE1. The ABIE1 comprises the pieces of information which specify a tire product. As explained in Chapter 2.4.2, an Aggregate Business Information Entity does not have an assigned

❶ Basic Business Information Entities (BBIEs)

BBIE1 = BBIE_EU_Tire_Type

$BBIE1_{assignedBC}$ = (≤ EU) || (~ Automomotive) = $BBIE1_{overallBC}$

BBIE2 = BBIE_Japan_Tire_Size

$BBIE2_{assignedBC}$ = (~ Japan) || (~ Automotive) = $BBIE2_{overallBC}$

---

❷ Aggregate Business Information Entity (ABIE)

ABIE1 = ABIE_Tire_Product

$ABIE1_{overallBC}$ = $BBIE1_{overallBC}$ || $BBIE2_{overallBC}$
= ((≤ EU) || (~ Japan)) || (~ Automotive)

ABIE1	ABIE_ Tire_Product	
	(overall) BC = ((≤ EU) || (~ Japan)) || (~ Automotive)	
	BBIE1	BBIE_EU_Tire_Type
		(assigned) BC = (≤ EU) || (~ Automotive)
	BBIE2	BBIE_Japan_Tire_Size
		(assigned) BC = (~ Japan) || (~ Automotive)

---

❸ Association Business Information Entities (ASBIEs)

ASBIE1 = ASBIE_Europe_Tire_ProductGroupMember

$ASBIE1_{assignedBC}$ = (≤ Europe) || (~ Automotive)

$ASBIE1_{overallBC}$ = $ASBIE1_{assignedBC}$ && $ABIE1_{overallBC}$ = (≤ EU) || (~ Automotive)

ASBIE2 = ASBIE_OutsideEurope_Tire_ProductGroupMember

$ASBIE2_{assignedBC}$ = $\overline{(\leq \text{Europe})}$ || (~ Automotive)

$ASBIE2_{overallBC}$ = $ASBIE2_{assignedBC}$ && $ABIE1_{overallBC}$ = (~ Japan) || (~ Automotive)

ASBIE1	ASBIE_Europe_Tire_ProductGroupMember		
	(assigned) BC = (≤ Europe) || (~ Automotive)		
	(overall) BC = (≤ EU) || (~ Automotive)		
	ABIE1a	ABIE_ Tire_Product	
		(overall) BC = ((≤ EU) || (~ Japan)) || (~ Automotive)	
		(effective) BC = (≤ EU) || (~ Automotive) ❹	
		BBIE1a	BBIE_EU_Tire_Type
			(assigned) BC = (≤ EU) || (~ Automotive)
			(effective) BC = (≤ EU) || (~ Automotive) ❹
		BBIE2a	BBIE_Japan_Tire_Size
			(assigned) BC = (~ Japan) || (~ Automotive)
			(effective) BC = Ø ❹
ASBIE2	ASBIE_OutsideEurope_Tire_ProductGroupMember		
	(assigned) BC = $\overline{(\leq \text{Europe})}$ || (~ Automotive)		
	(overall) BC = (~ Japan) || (~ Automotive)		
	ABIE1b	ABIE_ Tire_Product	
		(overall) BC = ((≤ EU) || (~ Japan)) || (~ Automotive)	
		(effective) BC = (~ Japan) || (~ Automotive) ❹	
		BBIE1b	BBIE_EU_Tire_Type
			(assigned) BC = (≤ EU) || (~ Automotive)
			(effective) BC = Ø ❹
		BBIE2b	BBIE_Japan_Tire_Size
			(assigned) BC = (~ Japan) || (~ Automotive)
			(effective) BC = (~ Japan) || (~ Automotive) ❹

Figure 4.9: Example - Enhanced UCM, application on the UN/CEFACT's Core Components

business context. The overall business context of an Aggregate Business Information Entity is dependent, and, thus, calculated based on the union of the overall business contexts of its included Basic Business Information Entities and Association Business Information Entities.

Considering the E-UCM BC expression syntax we introduced in Section 4.2.2, this can be expressed by the following Formula:

$$BC\_ABIE_{overall} = (||_{i=0}^{k} BC\_BBIE_{overall}) \; || \; (||_{i=0}^{l} BC\_ASBIE_{overall}) \; , \tag{4.2}$$

where k and l represent the numbers of the included Basic Business Information Entities and Association Business Information Entities, respectively. Hence, the overall business context of the ABIE1 is expressed as: $((\leq EU) \; || \; (\sim Japan)) \; || \; (\sim Automotive)$.

Contextualizing Association Core Components. As illustrated in Figure 4.9, Mark 3, the ABIE1 is associated by the Association Business Information Entities denoted by ASBIE1 and ASBIE2. These Association Business Information Entities are *derived by restriction* from the same Association Core Component, and they relate the group of tire products with the specific tire products. However, in our case the ASBIE1 and the ASBIE2 are valid in different geopolitical regions (Europe and the World except Europe, respectively). Thereby, the ASBIE1 has the assigned business context $(\leq Europe) \; || \; (\sim Automotive)$ and the ASBIE2 has the assigned business context $(\overline{\leq Europe}) \sqcup (\sim Automotive)$. (Note that the geopolitical subdomain of the business context assigned to the ASBIE2 is expressed using our new Complement operand.)

As explained in Chapter 2.4.2, the overall business context of an Association Business Information Entity is dependent, and, thus, calculated based on the intersection of its assigned business context and the overall business context of its associated Aggregate Business Information Entity. Considering the E-UCM BC expression syntax we introduced in Section 4.2.2, this can be expressed by the following Formula:

$$BC\_ASBIE_{overall} = BC\_ASBIE_{assigned} \; \&\& \; BC\_AssociatedABIE_{overall} \; . \tag{4.3}$$

Hence, the ASBIE1 is valid in the overall business context $(\leq EU)) \; || \; (\sim Automotive)$ and the ASBIE2 is valid in the overall business context $(\sim Japan)) \; || \; (\sim Automotive)$. Consequently, the overall business context of the ABIE1 and the overall business contexts of its Basic Business Information Entities are effectively narrowed. This is illustrated by the *effective business contexts* shown in Figure 4.9, Mark 4.

Furthermore, note that the effective business contexts of the same Aggregate Business Information Entity can be different depending on the overall business context of its associating Association Business Information Entity. For instance, in case the ABIE1 is associated by the ASBIE1, its effective business context is $(\leq EU) \; || \; (\sim Automotive)$. However, in case the ABIE1 is associated by the ASBIE2, its effective business context is $(\sim Japan) \; || \; (\sim Automotive)$. The same conclusion holds for the effective business contexts of the BBIE1 and the BBIE2 which are contained by the ABIE1. In particular, the effective business context of some Basic Business Information Entity may be null (denoted by $\emptyset$ in the example). Thereby, these Business Information Entities are not relevant in the particular business scenario, and, thus, should be excluded from the corresponding business documents.

Contextualizing Association Message Assembly Components. Finally, we now consider the associating Association Business Information Entity which represents the first level Association Business Information Entity (Association Message Assembly Component defined by the UN/CEFACT Business Message Model presented in Chapter 2.6). In this case, the Association Business Information Entity is valid in its assigned business context which is specified by the

business context assigned to the electronic business document as a single unit. Considering our previous discussion, we can conclude that a business context assigned to a business document *effectively narrows* the overall business contexts of the Business Information Entities which are comprised by this document. It means that this electronic document might contain the contextual knowledge which is actually wider than it is described by its assigned business context. This conclusion is meaningful for developing our business context aware Core Components modeling approach in Chapter 6.

4.4 Final Assessments

The work presented in this Chapter provides the following main contributions: (i) the applicability analysis of the UN/CEFACT Context Methodology [4], (ii) the Logic based Business Context Model to express business context of arbitrary complexity, and (iii) the approach to contextualize already existing electronic business documents. We briefly summarize them in the following.

The Enhanced UN/CEFACT Business Context Model defines our logic based approach to represent business context of electronic business documents. Our research foundations are underpinned by the already existing UN/CEFACT Business Context Model [4]. This model is designed to manage representations and applications of business context under the scope of the UN/CEFACT's standardization efforts, such as the UN/CEFACT's Core Components based business document standard described in Chapter 2.

This Chapter has stipulated that the standard UN/CEFACT's business context modeling approach is not complete and that it is not directly applicable to real-world scenarios. Therefore, we have provided two main extensions of the corresponding methodology and developed our Enhanced UN/CEFACT Business Context Model. It is important that the standard model completely conforms to our enhanced version. In other words, the already existing solutions developed in correspondence to the original UN/CEFACT Business Context Model are valid and directly applicable in regards to our Enhanced UN/CEFACT Business Context Model.

Our *first main enhancement* of the standard UN/CEFACT Business Context Model represents a transition from a centralized to a decentralized business context graph approach. Consequently, the total number of nodes and edges in the business context graph is significantly reduced, which in turn decreases the cost of graph initialization and maintenance operations. Furthermore, this allows re-using existing subgraphs and embedding them into the decentralized business context graph. Hence, in contrary to the original model, the Enhanced UN/CEFACT Business Context Model addresses the norms, such as interoperability, scalability, consistency and flexibility.

Our *second main enhancement* of the standard UN/CEFACT Business Context Model provides a formal definition of its already existing operands, introduces two additional operands (Symmetric Exclusion and Complement), and provides a formal expression of the operand rules. Thus, the expressiveness of the original business context modeling approach is enriched and the foundations for the business contextual reasoning are established.

Finally, in this Chapter we have explained how the Enhanced UN/CEFACT Business Context Model can be used to contextualize electronic business documents built upon the UN/CEFACT's

Core Components based business document standard. The presented approach is demonstrated by the corresponding example.

CHAPTER 5

Business Context Ontology

The previous Chapters have concluded that that (i) the logic based context modeling techniques and (ii) the ontology based context modeling techniques are most suitable for modeling the business context of electronic business documents. As explained in Chapter 2, we especially consider documents which are built upon the Core Components Technical Specification (CCTS).

This Chapter describes our Ontology based Model to represent business context. Likewise, it explains how electronic business documents can be contextualized by means of our model. The main conclusions of this phase of our research are discussed in [15, 109, 110] and exploited in [111].

The remainder of this Chapter is structured as follows. First, in Section 5.1 we describe our Ontology based Model to represent business context. Furthermore, we highlight the benefits of the application of our model and explain how these can streamline business context modeling. In Section 5.2 we demonstrate the usability of the Ontology based Model on the concepts defined by CCTS. Finally, the concluding remarks of this phase of our research are outlined in Section 5.3.

Our concurrent Logic based Business Context Model is explained in the previous Chapter 4. In order to ease the understanding and comparisons between our context modeling approaches, we try, when it is possible, to describe the Ontology based Model by adapting the corresponding terminology and relevant examples used in Chapter 4 for the explanation of the Logic based Context Model.

5.1 Ontology Based Business Context Model

As stipulated in Chapter 3.3.6, ontology based modeling is a widely applied approach used to model different scopes of context. This is usually the context restricted to some particular domain, such as the context restricted to pervasive systems [95], smart environments [97], ubiquitous robots [98], driver assistance systems [99], home health monitoring [100] and mobile devices [101]. In a nutshell, an ontology describes the concepts (entities, classes) in some

specific domain and the relationships that hold between them. A concept is presented as a set of individuals (facts) which share common properties. More complex concepts are defined by derivation from simpler concepts.

In the following we describe how the ontology based modeling approach can be utilized to model the business context (BC) of electronic business documents particularly. We present our Business Context Ontology (BCOnt) and explain its beneficial characteristics, such as: the high degree of formalism, knowledge sharing, reasoning mechanism and capabilities of dynamic interrelations with external ontologies. This serves as a basis to contextualize the business document modeling concepts built upon the Core Components Technical Specification (CCTS) (explained in Chapter 2).

5.1.1 Business Context Ontology

The Business Context Ontology (BCOnt) is an OWL DL based model [92–94] which represents the circumstances in which electronic business documents are or are not valid. The corresponding model is presented in Figure 5.1. Accordingly, this is the three level ontology model which comprises the upper, middle and lower level.

Each of the BCOnt levels is composed by the following elements: classes, individuals and properties. Classes are concrete representations of concepts or groups of concepts with similar characteristics. They are organized in a superclass-subclass hierarchy (taxonomy). Individuals are instances of classes. Relations between individuals are established by properties.

Every individual has an assigned BC value. A *BC value* is an atomic piece of knowledge that represents one aspect of the business context (e.g., geopolitical region, industry or activity). If an individual *A* belongs the class *ClassA*, and if an individual *B* belongs to the class *ClassB* which is the subclass of the *ClassA*, the BC value assigned to the individual *A* is restricted to the BC value assigned to the individual *B*. The *restriction weight (rw)* of the business contextual derivation represents the minimal number of the superclass-subclass restriction steps between two BC values which are resolved from two related individuals defined in the BCOnt hierarchy. For instance, in case that the business context resolved from the aforementioned individual *A* is directly restricted to the business context resolved from the aforementioned individual B (B is the direct child of A), the value of the corresponding *restriction weight* is $rw(A, B) = 1$. The application of the *restriction weight* is meaningful for calculating the levels of different business context matchings (described in the following Chapter 6).

The upper level of BCOnt is a high level ontology which refers to the general concepts of business context. As stipulated in Chapter 3.2, each concept of business context can be grouped into one of the primary BC categories. In the following Chapters we present an approach that is in principle independent on the number of context categories. When explaining it, we cover the geopolitical region, industry and activity primary BC categories which we consider to be most important for the characterization of business context. However, the proposed solutions can be easily applied to process the knowledge which originates from other business context categories as well. Therefore, the upper level of BCOnt is implemented by the following classes: *GeopoliticalOrganization*, *IndustryClassification* and *Activity* which encapsulate domains restricted by the location, industry and activity primary BC categories, respectively.

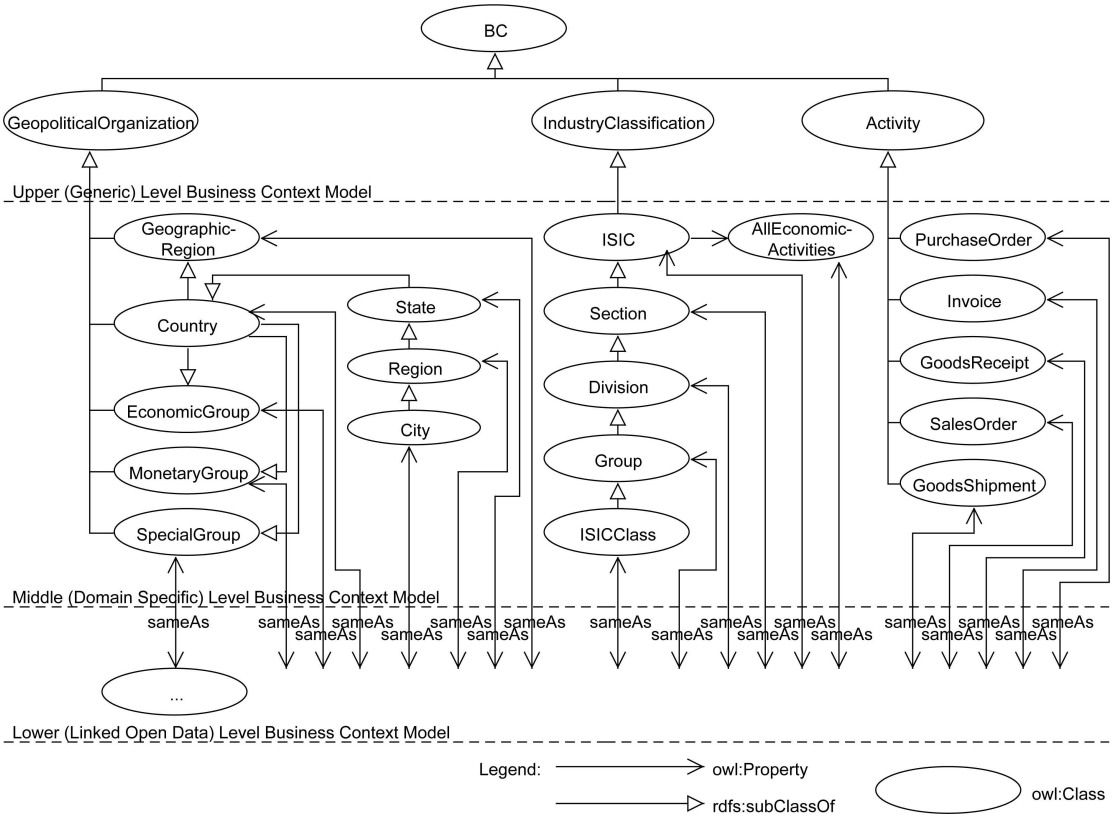


Figure 5.1: Business context ontology model

The middle level of the Ontology based Business Context Model covers more domain specific subontologies which refine concepts introduced by the upper level of the model. It consists of three main subontologies, namely *BCFAO*, *BCISIC* and *BCActivity*. *BCFAO* is the middle level subontology of our model which refines the geopolitical domain of business context. It is based on the geopolitical classification introduced by the Food and Agriculture Organization of the United Nations (FAO) [112]. *BCISIC* is the middle level subontology of our business context model which refines the industry domain of business context. It is built in respect to the International Standard Industrial Classification of All Economic Activities (ISIC) [105], proposed by the United Nations Statistics Division. We have decided to use the FAO and ISIC foundations to develop the middle level subontologies of BCOnt due to the following reasons: (i) both of these approaches belong to the group of the most complete and today worldwide applied classifications of their corresponding domains, and (ii) FAO, ISIC and UN/CEFACT are all standardized and propagated by the same institution, the United Nations. Finally, as shown in Figure 5.1, *BCActivity* is the third middle level subontology of the BCOnt model. It refines the activity domain of business context by providing a classification of all possible user activities, such as invoicing or purchasing.

The lower level of the BCOnt model is the collection of the subontologies which refer to the

```
<owl:NamedIndividual rdf:about="http://bcont.owl#Austria">
    <rdf:type rdf:resource="http://bcont.owl#Europe" />
    <owl:sameAs rdf:resource="http://dbpedia.org/page/Austria" />
    <foaf:homepage rdf:resource="http://www.bundeskanzleramt.at/" />
    <foaf:name>Republic of Austria</foaf:name>
    ...
</owl:NamedIndividual>
```

Figure 5.2: Business Context Ontology and Linked Open Data

more specific details of the more general concepts implemented in the upper levels of the model. It is essential that this level has plug in/unplug capabilities. Therefore, additional subontologies can be dynamically plugged in or unplugged from the model depending on the current business scenarios. Finally, the lower level of the BCOnt model is an extension point to the external ontologies located in the scope of Linked Open Data (LOD) [73]. Thus, in case that some concept is not defined in the model, BCOnt can be interrelated to some external ontology, such as DBpedia [73, 80], Geonames [73] and FOAF [73], where the missing concept is defined. This is implemented either by using the LOD interconnection techniques included by the semantics of the OWL DL language (property *owl:sameAs*, as shown in the example in Figure 5.2) or by applying a direct linking (property *foaf:name*, as shown in the example in Figure 5.2).

5.1.2 Business Context Expression

A particular business context presented by the BCOnt model is unambiguously expressed using the Description Logic (DL) based syntax [92]. Classes and individuals are uniquely identified by their names whose lexical notation starts with a capital letter, does not contain space characters and the first letter of each subsequent concatenated word is capitalized (CamelBack notation). For instance, in Figure 5.3 we present an excerpt of an exemplary BCOnt. It comprises the individual uniquely identified by its name *UnitedKingdom* (belongs to the class uniquely identified by its name *Country*), the individual uniquely identified by its name *Automotive* (belongs to the class uniquely identified by its name *Section*), etc. Generally speaking, each individual refers to the particular BC value. For instance, the aforementioned individual *UnitedKingdom* represents the business context which refers to the United Kingdom and the aforementioned individual *Automotive* represents the business context which refers to the Automotive industry.

Every property in the BCOnt model is uniquely identified by its name whose lexical notation starts with a lower case, does not have space characters and the first letter of each subsequent concatenated word is capitalized (CamelCase notation). For instance, in our exemplary ontology shown in Figure 5.3 the superclass-subclass relation is implemented along the property uniquely identified by its name *hasMember* and the subclass-superclass relation is implemented along the property uniquely identified by its name *isMemberOf*. Therefore, the individuals *Austria*, *Germany*, *France* and *UnitedKingdom* (all belong to the class *Country* defined by the model presented in Figure 5.1) refer to the BC values which additionally restrict the BC value assigned to the individual *EU* (belongs to the class *EconomicGroup*, the superclass of the class *Country*). As defined earlier, the minimal number of the superclass-subclass restriction steps along

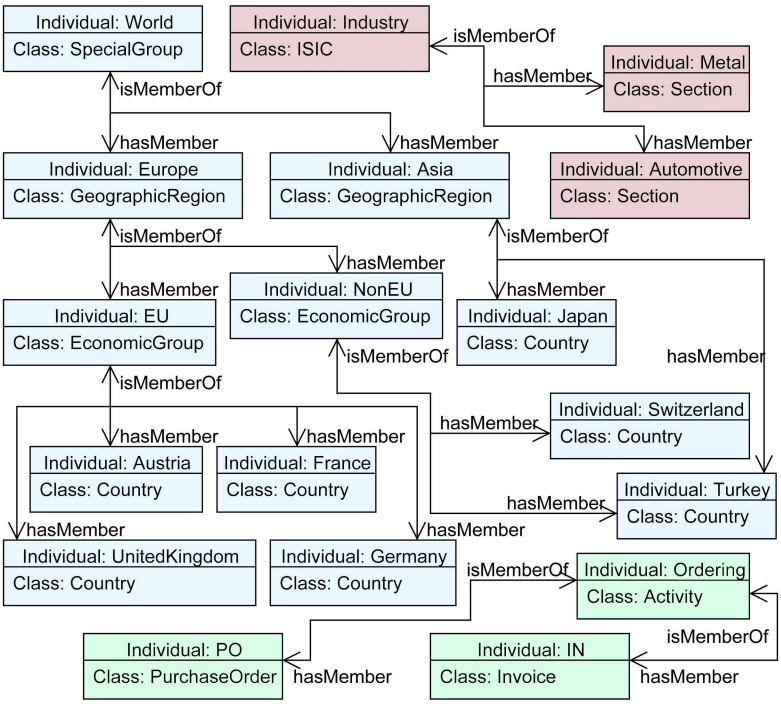


Figure 5.3: Exemplary Business Context Ontology (excerpt)

the business contextual derivation represents the value of the corresponding *restriction weight*. Thus, $rw(EU, Germany) = 1$, $rw(Europe, Germany) = 2$, $rw(World, Turkey) = 2$, etc.

A BC expression identifies a set of individuals from the BCOnt ontology that can be resolved to concrete BC values. These BC values combined form a unique business context. For instance, if the business context is assumed to be relevant in the scope of Asia, the corresponding BC expression is used for selecting the relevant individuals from BCOnt (the individuals *Japan* and *Turkey* from the exemplary ontology shown in Figure 5.3) which are then resolved to the particular BC values. The underlying model of BC expression is presented in Figure 5.4. Ac-

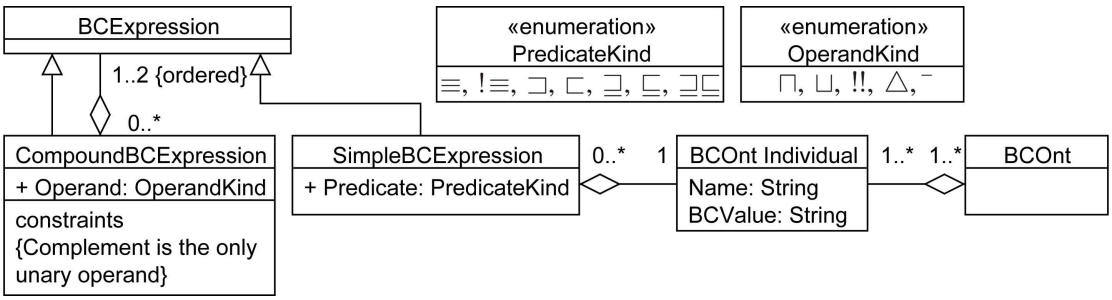


Figure 5.4: Business context expression model

cordingly, there are two types of BC expressions: (i) Simple BC expression and (ii) Compound BC expression.

A Simple BC expression is a BC expression which consists of the specified individual with an assigned predicate. Therefore, the lexical notation of a Simple BC expression follows the schema:

$$<SimpleBCExpression>=<predicate>\sqcup<IndividualName>\ .$$

The list of the allowed predicates is: (i) Equivalent to ($\equiv$), (ii) Not equivalent to ($!\equiv$), (iii) Superset of ($\sqsupset$), (iv) Subset of ($\sqsubset$), (v) Superset of or equivalent to ($\sqsupseteq$), (vi) Subset of or equivalent to ($\sqsubseteq$), and (vii) Subset of superset of or equivalent to ($\sqsupseteq\sqsubseteq$).

The predicate $\sqsubseteq$ identifies the individuals from the BCOnt model that are related to the specified individual along the subclass-superclass properties including the specified individual itself. The predicate $\sqsupseteq\sqsubseteq$ identifies the individuals from the BCOnt model that are related to the specified individual along the superclass-subclass and subclass-superclass properties including the specified individual itself. The other predicates can be described analogously. For instance, the business context applied on the exemplary ontology shown in Figure 5.3 and expressed as: ($\sqsubset\ EU$) identifies the following individuals: *Austria*, *Germany*, *France* and *UnitedKingdom*. The complete explanations of the BCOnt predicates are summarized in Figure 5.5.

A Compound BC expression is a BC expression which can consist of two ordered BC expressions connected by a binary operand, or it can consist of only one BC expression with an assigned unary operand. Thus, it is possible to express business contexts of arbitrary complexities. The corresponding lexical notation is:

$$<CompoundBCExpress>=[<unaryOperand>\sqcup(<SimpleBCExpress|CompoundBCExpress>)]\mid[(<SimpleBCExpress|CompoundBCExpress>)\sqcup<binaryOperand>\ \sqcup(<SimpleBCExpress|CompoundBCExpress>)].$$

Syntax	Name	Semantics
$\equiv$	Equivalent to	Resolves the specified individual from the BC ontology
$!\equiv$	Not equivalent to	Resolves all individuals which belong to the same primary BC context category as the specified individual from the BC ontology except the specified individual
$\sqsupset$	Superset of	Resolves individuals from the BC ontology that are related to the specified individual along the superclass-subclass properties
$\sqsubset$	Subset of	Resolves individuals from the BC ontology that are related to the specified individual along the subclass-superclass properties
$\sqsupseteq$	Superset of or equivalent to	Resolves individuals from the BC ontology that are related to the specified individual along the superclass-subclass properties including the specified individual itself
$\sqsubseteq$	Subset of or equivalent to	Resolves individuals from the BC ontology that are related to the specified individual along the subclass-superclass properties including the specified individual itself
$\sqsupseteq\sqsubseteq$	Subset of superset of or equivalent to	Resolves the individuals from the BC ontology that are related to the specified individual along the superclass-subclass and subclass-superclass properties including the specified individual itself

Figure 5.5: Business Context Ontology predicates

The list of the allowed operands is: Intersection ($\sqcap$), Union ($\sqcup$), Exclusion (!!), Symmetric Exclusion ($\triangle$) and Complement ($^{-}$).

The operand $\sqcap$ identifies the individuals from the BCOnt model which are resolved by both related BC expressions. The other operands can be described analogously. For instance, the business context applied on the exemplary ontology shown in Figure 5.3 and expressed as $(\equiv Austria) \sqcup (\sqsubset Asia)$ identifies the following individuals: *Austria*, *Japan* and *Turkey*. Finally, a BC expression can be negated using the unary operand Complement. For instance, the BC expression $\overline{(\sqsubset Europe)}$ identifies only those individuals from the geopolitical subontology of our exemplary Business Context Ontology (shown in Figure 5.3) which are not identified by the BC expression $(\sqsubset Europe)$. The complete explanations of the BCOnt operands are sum-

Syntax	Name	Semantics
$\sqcup$	Union	(A $\sqcup$ B) resolves the individuals from the BC ontology which are resolved from the left hand side BC expression (A), or from the right hand side BC expression (B), or from both of them.
$\sqcap$	Intersection	(A $\sqcap$ B) resolves the individuals from the BC ontology which are resolved from both BC expressions (A and B).
!!	Exclusion	(A !! B) the individuals from the BC ontology which are resolved from the left hand side BC expression (A) and are not resolved from the right hand side BC expression (B).
Δ	Symmetric Exclusion	(A Δ B) resolves the individuals from the BC ontology which are resolved from the left hand side BC expression (A) or from the right hand side BC expression (B) and are not resolved from both of these BC expressions.
$^{-}$	Complement	$\overline{(A)}$ resolves the individuals from the BC ontology which are not resolved from the specified BC expression (A) and which belong to the same primary BC context category as the individuals resolved from A.

Figure 5.6: Business Context Ontology operands

	BC Clause	Type	Resolved BC Nodes
1.	($\equiv$ Automotive)	Simple	Automotive
2.	(!$\equiv$ Automotive)	Simple	Industry, Metal
3.	($\sqsupset$ EU)	Simple	World, Europe
4.	($\sqsubset$ EU)	Simple	Austria, France, UnitedKingdom, Germany
5.	($\sqsupseteq$ EU)	Simple	World, Europe, EU
6.	($\sqsubseteq$ EU)	Simple	EU, Austria, France, UnitedKingdom, Germany
7.	($\sqsupseteq\sqsubseteq$ EU)	Simple	World, Europe, EU, Austria, France, UnitedKingdom, Germany
8.	($\sqsupset$ Europe) $\sqcup$ ($\equiv$ Japan)	Compound	World, Japan
9.	($\sqsubseteq$ Europe) $\sqcap$ ($\sqsupseteq$ France)	Compound	Europe, EU, France
10.	($\sqsubseteq$ Industry) !! (!$\equiv$ Automotive)	Compound	Automotive
11.	($\sqsubset$ Europe) Δ ($\sqsupseteq$ NonEU)	Compound	EU, Austria, France, UnitedKingdom, Germany, World, Europe, Switzerland, Turkey
12.	$\overline{(\sqsupseteq \text{Metal})}$	Compound	Automotive
13.	$(\overline{(\sqsubseteq \text{Asia}) \text{ !! } (\equiv \text{Japan})}) \sqcap$ ($\sqsubset$ World) && ($\sqsupseteq$ Germany)))	Compound	Europe, EU, Germany
14.	(($\sqsupseteq$ Switzerland) $\sqcap$ ($\sqsubset$ World)) $\sqcup$ ($\sqsupseteq$ Automotive) $\sqcup$ ($\equiv$ Ordering)	Compound	Europe, NonEU, Switzerland, Industry, Automotive, Ordering

Figure 5.7: Examples - Business Context Ontology business context expression

marized in Figure 5.6. The examples of the BCOnt expressions are presented in Figure 5.7. The underlying ontology model is illustrated in Figure 5.1.

5.1.3 Business Context Reasoning

As highlighted in Chapter 3, the reasoning capabilities are essential benefits of the ontology based modeling approach. Thereby, in our research we try to utilize them in order to derive new implicit business contextual knowledge. The BCOnt reasoning is based on two main reasoning tenets: (i) *ontology based* reasoning and (ii) *rule based* reasoning. Both techniques are implemented by the reasoning rules which are expressed by the DL based syntax [92–94].

The *ontology based* reasoning mechanism is applied to acquire an implicit business contextual knowledge by following the existing reasoning rules. These rules are integrated in respect to the semantics of the used OWL DL language [92–94], for example: subclass relation (*rdfs:subClassOf*), equality relation (*owl:sameAs*), and functional property (*owl:FunctionalProperty*). In our work we use ontology based reasoning to build class taxonomy and check consistency of the concepts. For example, (i) if the *ClassA* is the subclass of the *ClassB*, and (ii) if the *ClassB* is the subclass of the *ClassC*, the ontology based reasoning mechanism can infer that the *ClassA* is also the subclass of the *ClassC*. This can be formally expressed by the following rule:

$$(?A\ rdfs{:}subClassOf\ ?B) \sqcap (?B\ rdfs{:}subClassOf\ ?C) \Longrightarrow (?A\ rdfs{:}subClassOf\ ?C).$$

The *rule based* reasoning follows the reasoning rules which are not included by the OWL DL semantics. These rules are explicitly defined by users. In our work we use this approach to infer high level information from the low level information which holds in a specific business context. For example: (i) Two different electronic business documents ($BDoc_1$ and $BDoc_2$) are given. Both documents conform to CCTS described in Chapter 2; (ii) The introduced documents are valid in two different countries; (iii) These countries are members of the same economic organization; and (iv) One of the introduced documents ($BDoc_1$) contains Business Information Entity BIE_1 which is *derived by restriction* from some particular Core Component. In this case, the reasoning mechanism can infer that the other document ($BDoc_2$) contains Business Information Entity BIE_2 which is *derived by restriction* from the same Core Component as the BIE_1. This can be formally expressed as:

$$(?A\ bcont{:}hasMember\ ?B) \sqcap (?A\ bcont{:}hasMember\ ?C) \sqcap (?B\ bcont{:}hasBIE\ ?D) \sqcap (?D\ bcont{:}DbR\ ?E) \Longrightarrow (?C\ bcont{:}hasBIE\ ?F) \sqcap (?F\ bcont{:}DbR\ ?E).$$

The capabilities of business context reasoning can be exploited to model the structures of new electronic business documents by learning from already existing document models. The corresponding approach is explained in Chapter 6.

5.2 Contextualizing UN/CEFACTS's Core Components

In this Section we explain how electronic business documents based on CCTS can be contextualized by means of the BCOnt model. The used generic Core Components are introduced in our previous example described in Figure 2.8 in Chapter 2.3.4. Every Business Information Entity is

valid in a business context which is presented using our Ontology based Model. The particular BC values are specified using the BC expressions, as described in Section 5.1.2. For reasons of simplicity, in the following example we discard the activity BC category and consider only the location and industry BC categories presented by our exemplary BCOnt shown in Figure 5.3.

Contextualizing Basic Core Components. As illustrated in Figure 5.8, Mark 1, two Business Information Entities (BBIE1 and BBIE2) are given. The BBIE1 is a piece of information which refers to the type of a tire valid in the European Union. The BBIE2 is a piece of information which refers to the size of a tire valid in Japan. Thus, the BBIE1 has the assigned business context $(\sqsupseteq\sqsubseteq\ EU) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$ and the BBIE2 has the assigned business context $(\sqsupseteq\sqsubseteq\ Japan) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$. As explained in Chapter 2.4.2, these assigned business contexts also represent the overall business contexts of the corresponding Basic Business Information Entities.

Contextualizing Aggregate Core Components. In the next step, Figure 5.8, Mark 2, the BBIE1 and the BBIE2 are parts of the Aggregate Business Information Entity ABIE1. The ABIE1 comprises the pieces of information which specify a tire product. As explained in Chapter 2.4.2, an Aggregate Business Information Entity does not have an assigned business context. The overall business context of an Aggregate Business Information Entity is dependent, and, thus, calculated based on the union of the overall business contexts of its included Basic Business Information Entities and Association Business Information Entities. Considering the BCOnt BC expression syntax we introduced in Section 5.1.2, this can be expressed by the following Formula:

$$BC\_ABIE_{overall} = (\sqcup_{i=0}^{k} BC\_BBIE_{overall})\ \sqcup\ (\sqcup_{i=0}^{l} BC\_ASBIE_{overall})\ , \qquad (5.1)$$

where k and l represent the numbers of the included Basic Business Information Entities and Association Business Information Entities, respectively. Hence, the overall business context of the ABIE1 is expressed as: $((\sqsupseteq\sqsubseteq\ EU) \sqcup (\sqsupseteq\sqsubseteq\ Japan)) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$.

Contextualizing Association Core Components. As illustrated in Figure 5.8, Mark 3, the ABIE1 is associated by the Association Business Information Entities denoted by ASBIE1 and ASBIE2. These Association Business Information Entities are derived by restriction from the same Association Core Component, and they relate the group of tire products with the specific tire products. However, in our case the ASBIE1 and the ASBIE2 are valid in different geopolitical regions (Europe and Asia, respectively). Thereby, the ASBIE1 is valid in the assigned business context $(\sqsubset\ Europe) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$ and the ASBIE2 is valid in the assigned business $(\sqsubset\ Asia) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$.

As explained in Chapter 2.4.2, the overall business context of an Association Business Information Entity is dependent, and, thus, calculated based on the intersection of its assigned business context and the overall business context of its associated Aggregate Business Information Entity. Considering the BCOnt BC expression syntax we introduced in Section 5.1.2, this can be expressed by the following Formula:

$$BC\_ASBIE_{overall} = BC\_ASBIE_{assigned}\ \sqcap\ BC\_AssociatedABIE_{overall}\ . \qquad (5.2)$$

Hence, the ASBIE1 has the overall BC $(\sqsubseteq\ EU) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$ and the ASBIE2 has the overall BC $(\sqsubseteq\ Japan) \sqcup (\sqsupseteq\sqsubseteq\ Automotive)$. As an essential consequence, the overall business context of the ABIE1 and the overall business contexts of its Basic Business Information

❶ Basic Business Information Entities (BBIEs)

BBIE1 = BBIE_EU_Tire_Type

$BBIE1_{assignedBC}$ = (⊇⊑ EU) ⊔ (⊇⊑ Automotive) = $BBIE1_{overallBC}$

BBIE2 = BBIE_Japan_Tire_Size

$BBIE2_{assignedBC}$ = (⊇⊑ Japan) ⊔ (⊇⊑ Automotive) = $BBIE2_{overallBC}$

---

❷ Aggregate Business Information Entity (ABIE)

ABIE1 = ABIE_Tire_Product

$ABIE1_{overallBC}$ = $BBIE1_{overallBC}$ ⊔ $BBIE2_{overallBC}$

= ((⊇⊑ EU) ⊔ (⊇⊑ Japan)) ⊔ (⊇⊑ Automotive)

ABIE1	ABIE_ Tire_Product	
	(overall) BC = ((⊇⊑ EU) ⊔ (⊇⊑ Japan)) ⊔ (⊇⊑ Automotive)	
	BBIE1	BBIE_EU_Tire_Type
		(assigned) BC = (⊇⊑ EU) ⊔ (⊇⊑ Automotive)
	BBIE2	BBIE_Japan_Tire_Size
		(assigned) BC = (⊇⊑ Japan) ⊔ (⊇⊑ Automotive)

---

❸ Association Business Information Entities (ASBIEs)

ASBIE1 = ASBIE_Europe_Tire_ProductGroupMember

$ASBIE1_{assignedBC}$ = (⊏ Europe) ⊔ (⊇⊑ Automotive)

$ASBIE1_{overallBC}$ = $ASBIE1_{assignedBC}$ ⊓ $ABIE1_{overallBC}$ = (⊑ EU) ⊔ (⊇⊑ Automotive)

ASBIE2 = ASBIE_Asia_Tire_ProductGroupMember

$ASBIE2_{assignedBC}$ = (⊏ Asia) ⊔ (⊇⊑ Automotive)

$ASBIE2_{overallBC}$ = $ASBIE2_{assignedBC}$ ⊓ $ABIE1_{overallBC}$ = (⊑ Japan) ⊔ (⊇⊑ Automotive)

ASBIE1	ASBIE_Europe_Tire_ProductGroupMember		
	(assigned) BC = (⊏ Europe) ⊔ (⊇⊑ Automotive)		
	(overall) BC = (⊑ EU) ⊔ (⊇⊑ Automotive)		
	ABIE1a	ABIE_ Tire_Product	
		(overall) BC = ((⊇⊑ EU) ⊔ (⊇⊑ Japan)) ⊔ (⊇⊑ Automotive)	
		(effective) BC = (⊑ EU) ⊔ (⊇⊑ Automotive) ❹	
		BBIE1a	BBIE_EU_Tire_Type
			(assigned) BC = (⊇⊑ EU) ⊔ (⊇⊑ Automotive)
			(effective) BC = (⊑ EU) ⊔ (⊇⊑ Automotive) ❹
		BBIE2a	BBIE_Japan_Tire_Size
			(assigned) BC = (⊇⊑ Japan) ⊔ (⊇⊑ Automotive)
			(effective) BC = ⊥ ❹
ASBIE2	ASBIE_Asia_Tire_ProductGroupMember		
	(assigned) BC = (⊏ Asia) ⊔ (⊇⊑ Automotive)		
	(overall) BC = (⊑ Japan) ⊔ (⊇⊑ Automotive)		
	ABIE1b	ABIE_ Tire_Product	
		(overall) BC = ((⊇⊑ EU) ⊔ (⊇⊑ Japan)) ⊔ (⊇⊑ Automotive)	
		(effective) BC = (⊑ Japan) ⊔ (⊇⊑ Automotive) ❹	
		BBIE1b	BBIE_EU_Tire_Type
			(assigned) BC = (⊇⊑ EU) ⊔ (⊇⊑ Automotive)
			(effective) BC = ⊥ ❹
		BBIE2b	BBIE_Japan_Tire_Size
			(assigned) BC = (⊇⊑ Japan) ⊔ (⊇⊑ Automotive)
			(effective) BC = (⊑ Japan) ⊔ (⊇⊑ Automotive) ❹

Figure 5.8: Example - Business Context Ontology Model, application on the UN/CEFACT's Core Components

Entities are effectively narrowed. This is illustrated by the *effective business contexts* shown in Figure 5.8, Mark 4.

Furthermore, note that the effective business contexts of the same Aggregate Business In-

formation Entity can be different depending on the overall business context of its associating Association Business Information Entity. For instance, in case the ABIE1 is associated by the ASBIE1, its effective business context is $(\sqsubseteq EU) \sqcup (\sqsupseteq\sqsubseteq Automotive)$. However, in case the ABIE1 is associated by the ASBIE2, its effective business context is $(\sqsubseteq Japan) \sqcup (\sqsupseteq\sqsubseteq Automotive)$. The same conclusion holds for the effective business contexts of the BBIE1 and the BBIE2 which are contained by the ABIE1. In particular, the effective business context of some Basic Business Information Entity may be null (denoted by $\perp$ in the example). Thereby, these Basic Business Information Entities are not relevant in the specified business scenario, and, thus, they should be excluded from the corresponding electronic business documents.

Contextualizing Association Message Assembly Components. Finally, in case that the associating Association Business Information Entity represents the first level Association Business Information Entity (Association Message Assembly Component defined by the UN/CEFACT Business Message Model presented in Chapter 2.6), its assigned business context is specified by the business context assigned to the electronic business document as a single unit. Considering our previous discussion, we can conclude that a business context assigned to a business document *effectively narrows* the overall business contexts of the Business Information Entities which are comprised by this document. It means that this electronic document might contain the contextual knowledge which is actually wider than it is described by its assigned business context. This conclusion is meaningful for developing our business context aware Core Components modeling approach in Chapter 6.

5.3 Final Assessments

The Business Context Ontology Model (BCOnt) defines our ontology based approach to represent business context in which CCTS based electronic business documents are valid. This is the extensible OWL based [92–94] business context model enabling the automatic classification of concepts, reasoning (ontology based reasoning and rule based reasoning) and knowledge sharing. It is built upon the three level subontology structure. Thus, it is possible to extend the model with external pluggable concepts (e.g., ontologies from the scope of Linked Open Data [73]) depending on the current business scenario.

Furthermore, this Chapter has shown how our Business Context Ontology Model can be used to contextualize already existing electronic business documents. The presented approach is demonstrated by the corresponding example.

The work presented in this Chapter provides the following main contributions: (i) the Ontology based Business Context Model to express business context of arbitrary complexity, and (ii) the approach to contextualize already existing electronic business documents.

The previous Chapter 4 described our concurrent Logic based Business Context Model. Both Logic based and Ontology based Business Context Models serve as a basis for developing our approach to model data building blocks of new electronic business documents, which is explained in the following Chapters of this Doctoral Thesis.

CHAPTER 6

Business Context Aware Core Components Modeling

The previous Chapters have defined our Logic based and Ontology based Business Context Models to represent business context in which electronic business documents are valid. We especially consider documents which conform to the Core Components Technical Specification (CCTS) explained in Chapter 2. This Chapter describes our approach to generate the Core Component contents of new electronic business documents by learning from the already existing documents. The main conclusions of this phase of our research are discussed in [15, 102, 109].

The remainder of this Chapter is structured as follows. First, Section 6.1 highlights the main differences between our and other relevant document modeling approaches. It gives the basic idea of our approach, describes its starting assumptions, preconditions and postconditions without the details of the approach itself. All the details are explained in the rest of the Chapter. Different types of business context matchings are defined in Section 6.2. Section 6.3 introduces our first Core Components modeling scenario and discusses the corresponding conceptual solution. Different types of mappings between contextualized Core Components are explained in Section 6.4. These mapping techniques are exploited in the following Section 6.5. This Section introduces our second Core Components modeling scenario and elaborates on the corresponding conceptual solution. Finally, the concluding remarks of this phase of our research are summarized in Section 6.6.

6.1 Problem Statement

Business transactions between different enterprises are more and more executed by a flow of well-defined electronic business documents. The concrete structures of these documents significantly differ depending on industry branches and geopolitical regions in which the underlying inter-organizational business processes are executed. However, business document standards are generic, and, thus, must be adapted to a specific *business context (BC)*.

In order to use a business document standard in a specific context, user groups develop business document implementation guidelines. A *business document implementation guideline (BDocIG)* represents a context specific constraint of the underlying generic document standard. This constraint takes only a small subset (3-5%) of the standard. However, defining a new implementation guideline for a specific business context always starts from scratch, which is time-consuming and often leads to heterogeneous interpretations of the standard. This Chapter describes our approach to speed up this development process and to create more homogeneous implementation guidelines by learning from already existing models.

6.1.1 Overview on Business Documents Engineering

As highlighted in [113], the essential issue for inter-organizational business document exchange is to maintain semantic consistency of the exchanged data. Likewise, in [114] the documents are described as public interfaces to their respective business processes. Similarly, in [115] the management of data content standards is seen as a cornerstone issue of successful electronic commerce. Finally, in [116] the seamless and automatic exchange of electronic documents is considered to be the most important aim in today's business.

All these approaches provide their own solutions which address business document design and information interoperability issues. For instance, these solutions are based on separation of a business document into layers of document structure and document semantic [113], linking data elements with the common semantic meanings [115], and transformations of document standards [116]. Therefore, the information interoperability issues are tackled by developing special schemas representing semantic mappings between different data contents. However, none of these approaches deals with the variability of the business semantics in different business contexts. Accordingly, these approaches deal with the semantic mappings, whereas we address which semantics are needed in different business environments.

6.1.2 Core Components Modeling - Basic Concepts

Before we can apply our business context aware Core Components modeling approach, the already existing Business Document Implementation Guidelines (ExistBDocIGs) must be contextualized by means of our Enhanced UN/CEFACT Business Context Model (E-UCM) or Business Context Ontology Model (BCOnt). The corresponding techniques are described in Chapters 4.3 and 5.2, respectively.

Afterwards, the Core Components which are valid in the requested business context are extracted from the ExistBDocIGs and re-used for the generation of a missing guideline. As explained in Chapter 4, an edge in the E-UCM model restricts the business context resolved from its source BC node to the business context resolved from its target BC node. As explained in Chapter 5, a superclass-subclass property in the BCOnt model restricts the business context resolved from its source individual to the business context resolved from its target individual. Thereby, not only the contextual knowledge that originates from the particular concept (from the BC node in the E-UCM model or from the individual in the BCOnt model), but also the knowledge indicated by its subconcepts (by children BC nodes in the E-UCM model or by target individuals in the BCOnt model) could be utilized for the generation of a new guideline.

Therefore, we define different degrees of BC matchings based on the number of the restriction steps between the corresponding concepts (BC nodes in the E-UCM model and individuals in the BCOnt model). Our business context aware Core Components modeling approach compares the requested business context with the business contexts in which the already existing Core Components are valid considering different degrees of the BC match. Clearly, the generated guidelines may differ depending on the acceptable degrees of the BC matchings which are specified by a user.

Furthermore, we have detected that the Core Component contents of related documents (such as Purchase Order and Invoice documents) may overlap. However, we have noticed that the mapping between the Core Components contained by the related documents which are valid in one business context does not necessarily mean that the analogous mapping exists in other business contexts. Our business context aware Core Components modeling approach correlates the to-be-developed guideline with the ExistBDocIGs. It tries to predict the overlapping Core Component contents measuring the matching degrees between the requested business context and the business contexts in which the certain Core Component mappings occur or do not occur. Based on these calculations, our approach makes a decision whether the Core Component overlappings between the to-be-developed guideline and its related guidelines exist or do not exist.

The generated guideline can be additionally tailored by applying BC reasoning rules (explained in Chapters 4.2.3 and 5.1.3, when the E-UCM model and the BCOnt model are used, respectively). The reasoning rules can infer that either additional Core Components can be added into the new guideline or that the Core Components generated in the previous steps of the approach have to be excluded from it.

6.1.3 Starting Assumptions

The main assumptions which hold in our research on business context aware Core Components modeling are defined by: (i) business document standard, (ii) business context representation and (iii) main use case scenarios.

Business Document Standard. We consider electronic business documents which are based on the business document standard defined by CCTS. The most important foundations of this standard are already explained in Chapter 2. Accordingly, each Business Information Entity (BIE) is *derived by restriction* from the corresponding Core Component (CC) in respect to the particular business context (BC). In the following we represent this business context using our Enhanced UN/CEFACT Business Context Model (E-UCM) or using our Business Context Ontology Model (BCOnt).

For generality reasons, we denote both BC nodes (defined by the E-UCM model) and individuals (defined by the BCOnt model) by a common terminology - *concepts*. The proposed solutions are demonstrated on the examples which use either E-UCM model or BCOnt model. However, the analogous solutions hold in cases when the concurrent business context model is used. The solutions which significantly differ depending on the used business context model are especially emphasized and demonstrated on separated examples.

Finally, as introduced in Chapters 4.2.1 and 5.1.2, the *restriction weight (rw)* of the business contextual derivation represents the minimal number of the concepts (restriction steps) between two BC values which are resolved from two related concepts defined by the used business context

model. The role of the *restriction weight* is meaningful for calculating the business context matching levels (described in the following Sections).

Business Context. Both the E-UCM model and the BCOnt model define business context as a set of attributes in which some Business Information Entity is or is not valid. As explained in Chapter 3, each of these attributes can be grouped into one of the primary BC categories. Thus, we represent business context as $BC_1 = (ctg_1, ctg_2, ..., ctg_c)$, where ctg_j is the primary BC category identified by its index j. c specifies the total number of the primary BC categories.

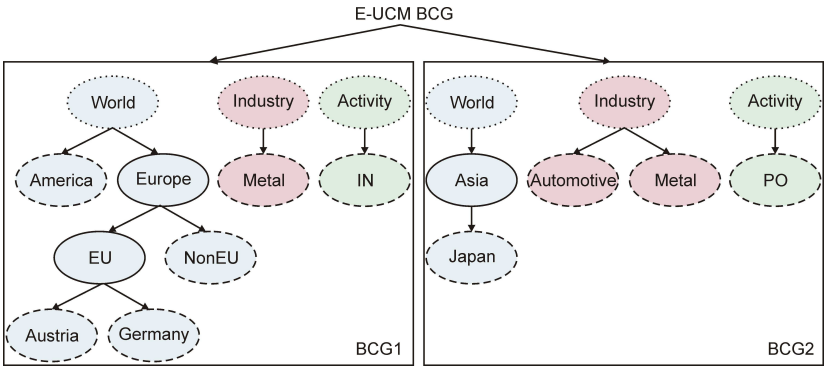


Figure 6.1: Enhanced UCM business context graph

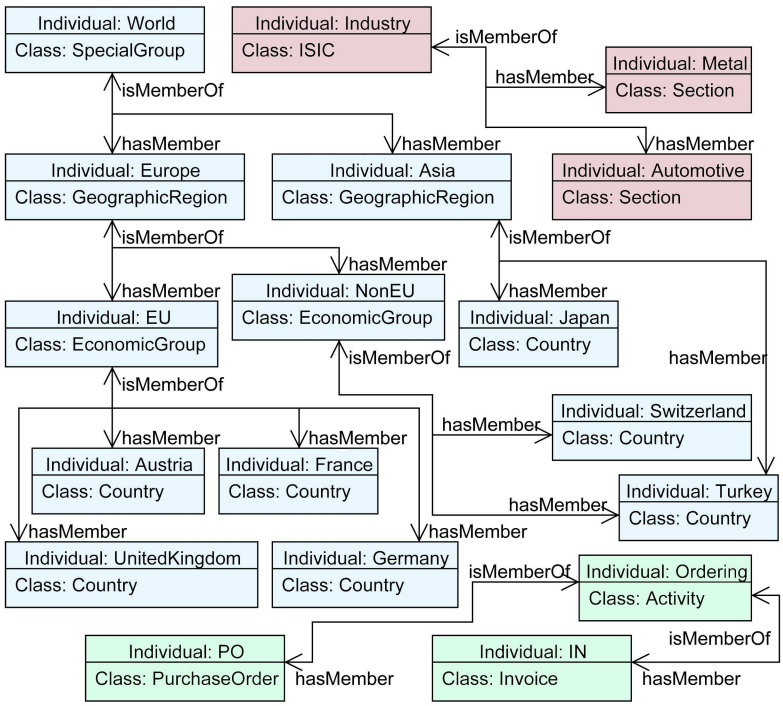


Figure 6.2: Exemplary Business Context Ontology (excerpt)

Our business context aware Core Components modeling approach is in principle independent on the number of the BC categories. When explaining our approach, we use the following categories: (i) geopolitical BC category, (ii) industry BC category and (iii) activity BC category. The E-UCM business context graph and the excerpt of the BCOnt ontology which are used in the corresponding examples are presented in Figures 6.1 and 6.2, respectively.

Scenarios of Business Context Aware Core Components Modeling. The following Sections of this Chapter describe our approach to apply the business contextual knowledge for (semi-) automatically generating semantically interoperable Core Components. We consider two main scenarios. The first is defined by the set of the *non-paired business document implementation guidelines*, while the second is defined by the set of the *paired business document implementation guidelines*.

6.1.4 Scenario A - Non-paired Business Document Implementation Guidelines

The problem statement of Scenario A is illustrated in Figure 6.3. Accordingly, there is the set of $n-1$ ($n > 1, n \in N$) already existing contextualized business document implementation guidelines (*ExistBDocIGs*). In Figure 6.3, we present these guidelines by $BDocIG_1$, $BDocIG_2$, ..., $BDocIG_{n-1}$. Each of them is valid in its assigned business context (BC_{as_1}, BC_{as_2}, ..., $BC_{as\_(n-1)}$). However, the implementation guideline $BDocIG_n$ - valid in the requested business context ($BC_{req} = BC_{as\_n}$) - is missing. Our main task is to provide an approach to apply the already existing business contextual knowledge contained by the *ExistBDocIGs* in order to (semi-) automatically generate the new $BDocIG_n$ which is valid in the business context requested by a user (BC_{req}). The described problem can be expressed by the following Formula:

$$BDocIG_n = AlgA(BDocIG_1, BDocIG_2, \ldots, BDocIG_{n-1}, BC_{req}), \tag{6.1}$$

where $AlgA$ represents the algorithm we have developed to calculate the BIE structure of the missing business document implementation guideline $BDocIG_n$.

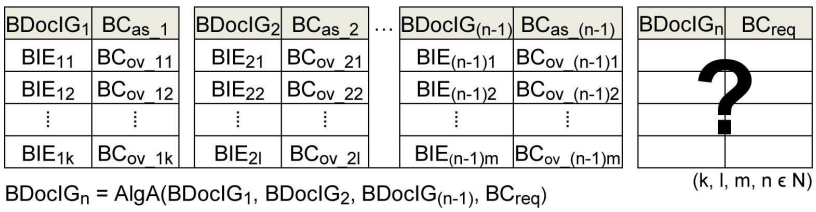


Figure 6.3: Scenario A - problem statement

6.1.5 Scenario B - Paired Business Document Implementation Guidelines

Business ecosystems often comprise the business activities, such as purchasing, which executions directly imply the executions of the consequent business activities, such as invoicing. Therefore, the exchange of one document type, such as Purchase Order, usually invokes the exchange of its subsequent document types, such as Invoice. In the following we refer to the

subsequent electronic business documents as the *paired documents*. We denote the corresponding relationship by $P(BDoc_1, BDoc_2)$, where the *BDoc*$_1$ and the *BDoc*$_2$ represent the paired documents. The real-world business solutions usually involve the exchanges of the subsequent Purchase Order (PO) and Invoice (IN) documents, thus, in the following we will especially consider these document types.

The problem statement of Scenario B is illustrated in Figure 6.4. Accordingly, there is the set of $(n-1)$ pairs of the already existing, contextualized business document implementation guidelines. We denote these pairs by $P(POIG_1, INIG_1)$, $P(POIG_2, INIG_2)$, ..., $P(POIG_{n-1}, INIG_{n-1})$. The implementation guidelines are valid in their assigned business contexts (BC_{as_p1}, BC_{as_i1}, BC_{as_p2}, ..., $BC_{as\_i(n-1)}$). The $POIG_n$ is the Purchase Order implementation guideline which paired Invoice implementation guideline, denoted by $INIG_n$, is unknown. Our main task is to provide an approach to apply the already existing business contextual knowledge contained by the pairs of the *ExistBDocIGs* in order to (semi-) automatically generate the missing document implementation guideline $INIG_n$. The described problem can be expressed by the following Formula:

$$INIG_n = AlgB(P(POIG_1, INIG_1), P(POIG_2, INIG_2), ..., P(POIG_{n-1}, INIG_{n-1}), POIG_n, BC_{req}), \quad (6.2)$$

where $AlgB$ represents the algorithm we have developed to calculate the BIE structure of the missing $INIG_n$. BC_{req} is the business context requested by a user in which the generated guideline $INIG_n$ will be used ($BC_{req} = BC_{as\_in}$).

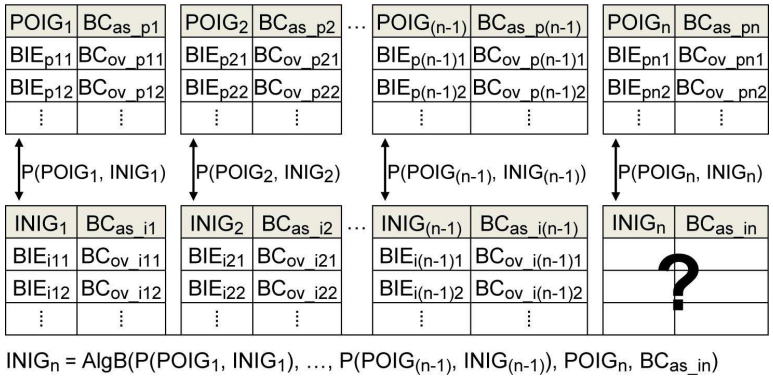


Figure 6.4: Scenario B - problem statement

Note that Scenario B addresses the main problem statement of this Doctoral Thesis which is introduced in Chapter 1.3. The solutions described in the following are independent on the number of the related business document types. Therefore, they can be utilized in cases when more than two subsequent business document types are correlated as well.

6.2 Business Context Match

Before we can explain our conceptual solution of the scenarios described by Formula 6.1 and Formula 6.2, we first must define the *BC match*. Therefore, in the following we distinguish between four types of the *BC match*: (i) *Direct BC match*, (ii) *Indirect BC match*, (iii) *Partial BC match* and (iv) *BC Non-match*.

Definition 6.2.1 : Direct BC Match. The business context BC_1 is *directly matched* to the business context BC_2, represented by $BC_1 \subseteq BC_2$, iff every concept resolved from the BC_1 can also be resolved from the BC_2.

For instance, there is the *direct match* between the $BC_1 = (= EU) \mid\mid (= NonEU)$ and the $BC_2 = (< Europe)$ presented using our E-UCM business context model. The corresponding E-UCM BC graph is shown in Figure 6.1.

Definition 6.2.2 : Indirect BC Match. The business context BC_1 is *indirectly matched* to the business context BC_2, represented by $BC_1 \subseteq^i BC_2$, iff every concept resolved from the BC_1 which can not be resolved from the BC_2 belongs to the set of the children concepts resolved from the BC_2.

For instance, there is the *indirect BC match* between the $BC_1 = (\sqsubseteq Austria) \sqcup (\sqsubset France)$ and the $BC_2 = (\equiv Europe)$. The business context in this example is presented using our BCOnt model which corresponding instance is shown in Figure 6.2.

Definition 6.2.3 : Partial BC Match. The business context BC_1 is *partially matched* to the business context BC_2, represented by $BC_1 \subseteq^{p_1,p_2} BC_2$ ($p_1, p_2 \in N \wedge p_1, p_2 > 0$), iff every concept resolved from the BC_1 which neither can be resolved from the BC_2 nor belongs to the set of the children concepts of the concepts resolved from the BC_2 can be identified as the concept resolved from the BC_{ref} such that $BC_1 \subseteq^i BC_{ref} \wedge BC_2 \subseteq^i BC_{ref}$. We refer to the BC_{ref} as the *Reference BC*, while p_1 and p_2 are the maximal allowed values of the *Partial BC matching indexes*. The *Partial BC matching index* represents the *restriction weight* between the concept identified by the *Reference BC* and the corresponding partially matched concept. Thus, the following holds: $p_1 = maxAllowed(rw(BC_{ref}, BC_1))$ and $p_2 = maxAllowed(rw(BC_{ref}, BC_2))$. In case that $p_1 = p_2 = p$, we denote the *partial BC match* between the BC_1 and the BC_2 by $BC_1 \subseteq^p BC_2$.

For instance, there is the *partial BC match* between the $BC_1 = (\leq Austria)$ and the $BC_2 = (\leq Germany)$ for $p = 1$ ($BC_{ref} = (= EU)$), where the business context is presented using our E-UCM model. The corresponding E-UCM business context graph is shown in Figure 6.1.

Definition 6.2.4 : BC Non-match. The business context BC_1 is *unmatched* to the business context BC_2, represented by $BC_1 \not\subseteq^{n_1,n_2} BC_2$, iff there is at least one concept resolved from the BC_1 which neither can be resolved from the BC_2 nor belongs to the set of the children concepts of the concepts resolved from the BC_2 nor can be resolved from some BC_{ref} such that $BC_1 \subseteq^i BC_{ref} \wedge BC_2 \subseteq^i BC_{ref}$. We refer to the BC_{ref} as the *Reference BC*, while

n_1 and n_2 are the maximal allowed values of the *BC non-matching indexes*. The *BC non-matching index* represents the *restriction weight* between the concept identified by the *Reference BC* and the corresponding non-matched concept. Thus, the following holds: $n_1 = maxAllowed(rw(BC_{ref}, BC_1))$ and $n_2 = maxAllowed(rw(BC_{ref}, BC_2))$. In case that $n_1 = n_2 = n$, we denote the *BC non-match* between the BC_1 and the BC_2 by $BC_1 \not\sqsubseteq^n BC_2$.

For instance, the $BC_1 = (\sqsubseteq France)$ can not be matched to the $BC_2 = (\sqsubseteq Japan)$ for $n = 2$, where the business context is presented using our exemplary Business Context Ontology shown in Figure 6.2.

6.3 Scenario A - Conceptual Solution

In the following we explain the conceptual solution of Scenario A described by Formula 6.1 and illustrated in Figure 6.3. $AlgA$ consists of five main processing steps: (i) extraction of the already existing BIEs, (ii) application of the *direct BC match*, (iii) application of the *indirect BC match*, (iv) application of the *partial BC match* and (v) application of the *BC reasoning rules*. We present $AlgA$ by the pseudo-code given in Algorithm 6.1. The realized conceptual solution is illustrated in Figure 6.5.

```
Input: ExistBDocIGs, BC_req, p_1, p_2
Output: CustBDocIG
for each BDocIG : ExistBDocIGs do
    BIEList = BDocIG.extractBIEs();
    GenBDocIG.addBIEs(BIEList);
end for
GenBDocIG.calculateOverallBC(); {Formulas 4.2, 4.3, 5.1 and 5.2}
for each bie : GenBDocIG do
    if directBCMatch(BC_req, bie.overallBC) then
        CustBDocIG.add(bie);
    end if
end for
for each bie : GenBDocIG do
    if !directBCMatch(BC_req, bie.overallBC) and
    indirectBCMatch(BC_req, bie.overallBC) then
        CustBDocIG.add(bie);
    end if
end for
for each bie : GenBDocIG do
    if !directBCMatch(BC_req, bie.overallBC) and
    !indirectBCMatch(BC_req, bie.overallBC) and
    partialBCMatch(BC_req, bie.overallBC, p_1, p_2) then
        CustBDocIG.add(bie);
    end if
end for
CustBDocIG.applyRuleBasedBCReasoning(rules);
return CustBDocIG;
```

Algorithm 6.1: AlgA

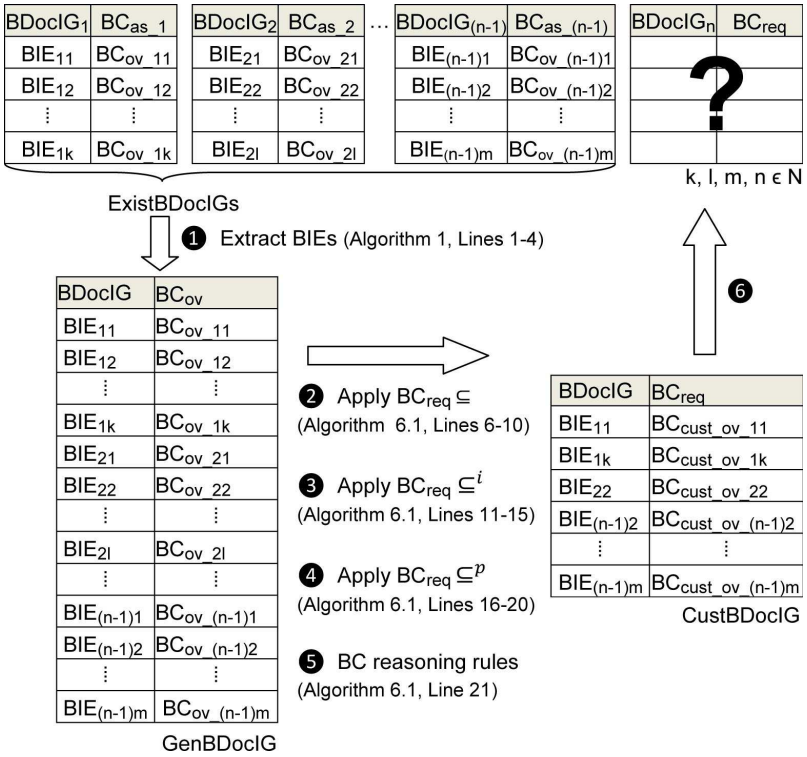


Figure 6.5: Scenario A - conceptual solution

6.3.1 Algorithm A - Generic Business Document Implementation Guideline

As shown in Figure 6.5, the business contextual knowledge is contained by the implementation guidelines of the already existing business documents ($BDocIG_1$ valid in BC_{as_1}, $BDocIG_2$ valid in BC_{as_2}, ..., $BDocIG_{n-1}$ valid in $BC_{as\_(n-1)}$). According to CCTS (Chapter 2), the implementation guidelines consist of the semantically interoperable BIE blocks, where each BIE is valid in its assigned and overall business context (BC_{as} and BC_{ov}, respectively).

In Chapters 4.3 and 5.2 we have underlined that a business context assigned to a business document effectively narrows the overall business contexts of the BIEs which are comprised by the aforementioned document. It means that this document might comprise the contextual knowledge which is actually wider than it is described by its assigned business context. Thus, in order not to lose segments of the contextual knowledge, in our further calculations we take into account the overall business contexts of BIEs, and we do not consider the assigned business contexts of the *BDocIGs* (BC_{as_1}, BC_{as_2}, ..., $BC_{as\_(n-1)}$ shown in Figure 6.5) where these BIEs are located.

In the first step of $AlgA$ (Figure 6.5, Mark 1), all BIEs are extracted from the original *ExistBDocIGs* and added into a Generic Business Document Implementation Guideline (*GenBDocIG*). Therefore, the *GenBDocIG* is the implementation guideline which contains the whole business contextual knowledge collected from the already existing business document implementation

guidelines.

6.3.2 Algorithm A - Customized Business Document Implementation Guideline

In the second step of $AlgA$, only those BIEs - which are valid in the BC *directly matched* with the BC requested by a user ($BC_{req} \subseteq BC_{BIE}$) - are extracted from the *GenBDocIG* (Figure 6.5, Mark 2) and are added into a new Customized Business Document Implementation Guideline (*CustBDocIG*). Therefore, the *CustBDocIG* is the new implementation guideline which is relevant in the business context explicitly requested by a user.

6.3.3 Algorithm A - Business Context Reasoning

We can streamline the conceptual solution described in the previous Subsection by introducing reasoning capabilities. The reasoning capabilities are underpinned by the reasoning techniques provided by the used business context model. The reasoning techniques supported by the E-UCM business context model are explained in Chapter 4.2.3, while the reasoning techniques supported by the BCOnt business context model are explained in Chapter 5.1.3. In a nutshell, the BC reasoning can be achieved by the two main tenets: (i) learning from a business context model and (ii) learning from a business contextual knowledge database.

The core of the first BC reasoning tenet is either (i) the business context organization in the form of the Directed Acyclic Graph (DAG) (when our E-UCM business context model is used), or (ii) the business context organization in the form of the ontology (when our BCOnt business context model is used). According to the E-UCM business context model, an edge in the business context graph restricts the BC assigned to its source node to the BC assigned to its target node. According to the BCOnt business context model, the superclass-subclass property restricts the BC assigned to its source individual to the BC assigned to its target individual. Thereby, not only the contextual knowledge that originates from the particular concept (from the BC node in the E-UCM business context model or from the individual in the BCOnt business context model), but also the knowledge indicated by its subconcepts (by children BC nodes in the E-UCM business context model or by target individuals in the BCOnt business context model) could be utilized for the further customization of the *GenBDocIG*. Therefore, in the third step of our algorithm $AlgA$ (Figure 6.5, Mark 3), only those BIEs which are valid in the BC *indirectly matched* with the BC requested by a user ($BC_{req} \subseteq^{i} BC_{BIE}$) are extracted from the *GenBDocIG* and added into the *CustBDocIG*. Analogously, in the fourth step of $AlgA$ (Figure 6.5, Mark 4), only those BIEs - which are valid in the BC *partially matched* with the BC requested by a user ($BC_{req} \subseteq^{p} BC_{BIE}$) - are extracted from the *GenBDocIG* and added into the *CustBDocIG*.

The second BC reasoning tenet comes as a direct consequence of the application of the *rule based* reasoning technique provided by the used business context model. The new knowledge, thus, can be derived from the existing business contextual knowledge by following the explicitly provided BC reasoning rules. This is illustrated as the fifth execution step of *AlgA* in Figure 6.5, Mark 5. The more complete explanations and the corresponding examples are already presented in Chapters 4.2.3 and 5.1.3 when our E-UCM and BCOnt business context models are used, respectively.

6.3.4 Scenario A - Example

In the following we show how our conceptual solution can be applied in practice to generate a new business document implementation guideline valid in a requested business context. We provide two analogous examples. In the first example business context is represented using our E-UCM business context model (explained in Chapter 4), while in the second example business context is represented using our BCOnt business context model (explained in Chapter 5). We especially highlight the situations when the final outcomes of algorithm A significantly differ depending on the used business context model.

Example - Application of Scenario A Using the E-UCM Model. The following example shows how our conceptual solution can be applied in practice to generate a new BDocIG valid in a requested BC. The business contextual knowledge used in the example is represented by our E-UCM business context model (see Figure 6.1). The particular BC values are specified using the BC expressions, as already described in Chapter 4.2.2. The requested business context is expressed as $BC_{req} = (= EU) \mid\mid (= BCG2\_Metal) \mid\mid (= PO)$. The *Partial BC matching indexes* are: $p = p_1 = p_2 = 1$. The *BC non-matching indexes* are: $n = n_1 = n_2 = 1$.

The BIEs of the already existing BDocIGs ($BDocIG_1$, $BDocIG_2$, ... and $BDocIG_{(n-1)}$) are presented in Figure 6.6. Each of these guidelines is valid in its assigned BC. The included BIEs are valid in their overall BCs which are calculated following the Formulas 4.2 and 4.3. These pieces of information are shown in Figure 6.6 in the corresponding fields.

As we have already emphasized in Chapters 4.3 and 5.2, the overall BC of a single BIE can be wider than the assigned BC of its containing electronic business document. This situation is illustrated in the example. For instance, the *BDocIG*$_1$ is valid in the $BC_{as\_1} = (\leq Germany) \mid\mid (= BCG2\_Metal) \mid\mid (= PO)$. It narrows the $BC_{ov\_11} = (\leq World) \mid\mid (\leq BCG2\_Industry) \mid\mid (= PO)$ previously calculated for the included *ABIE*$_{11}$. Thereby, the $BC_{req} = (= EU) \mid\mid (= BCG2\_Metal) \mid\mid (= PO)$ is valid in the scope of the *ABIE*$_{11}$ (*direct BC match*: $BC_{req} \subseteq BC_{ov\_11}$), but not in the scope of the BC assigned to the guideline *BDocIG*$_1$ (*BC non-match*: $BC_{req} \not\subseteq^{n=1} BC_{as\_11}$). However, this contextual knowledge is meaningful and we do not exclude it from the further steps of the customization.

According to *AlgA*, all BIEs located in the *ExistBDocIGs* are extracted and added into a *GenBDocIG*. The excerpt of the composed guideline is illustrated in Figure 6.7, Mark 1. Afterwards, in order to select only those BIEs which are relevant in the specified geopolitical region, industry and activity, we apply the requested BC under the *GenBDocIG*. The excerpt of the resulting *CustBDocIG* is presented in Figure 6.7, Mark 2.

For instance, the *BBIE*$_{15}$ located in the *GenBDocIG* is relevant in the requested BC (*direct BC match*: $BC_{req} \subseteq BC_{ov\_15}$)). Hence, the corresponding BIEs (*BBIE*$_{15}$ and *ABIE*$_{11}$) are re-used to generate the missing *CustBDocIG*. Likewise, the geopolitical domain of the BC_{req} is partially matched to the geopolitical domain of the overall BC in which the *BBIE*$_{16}$ is valid. The corresponding industry and activity BC domains are directly matched. Hence, the following holds: $BC_{req} \subseteq^{p=1} BC_{ov\_16}$. Thus, the *BBIE*$_{16}$ should be included by the *CustBDocIG* as well. However, the *BBIE*$_{14}$ - located in the same ABIE as the previously re-used *BBIE*$_{15}$ and *BBIE*$_{16}$ - is not valid in the industry domain specified by the requested BC (*BC non-match*: $BC_{req} \not\subseteq^{n=1} BC_{ov\_14}$). Consequently, it is omitted from the *CustBDocIG*.

BDocIG$_1$	BC$_{assigned}$ = (≤ Germany) || (= BCG2_Metal) || (= PO)	
ABIE$_{11}$	Name: Product BC$_{overall}$ = (≤ World) || (≤ BCG2_Industry) || (= PO)	
	BBIE$_{14}$	Name: Type BC$_{overall}$ = (≤ World) || (< Automotive) || (= PO)
	BBIE$_{15}$	Name: Status BC$_{overall}$ = (≤ World) || $\overline{(> \text{Automotive})}$ || (= PO)
	BBIE$_{16}$	Name: Color BC$_{overall}$ = ((≤ Germany) || (= NonEU)) || (≤ BCG2_Industry) || (= PO)
	⋮	
ABIE$_{12}$	Name: Address BC$_{overall}$ = (< World) || (≤ BCG2_Metal) || (= PO)	
	BBIE$_{17}$	Name: Postcode BC$_{overall}$ = (≤ EU) || (= BCG2_Metal) || (= PO)
	ASBIE$_{18}$	Name: Country BC$_{overall}$ = (≤ Germany) || (≤ BCG2_Metal) || (= PO)
	⋮	
ABIE$_{13}$	Name: Transaction BC$_{overall}$ = (≤ Germany) || (≥ BCG2_Metal) || (= PO)	
	⋮	
⋮		

,

BDocIG$_2$	BC$_{assigned}$ = (< EU) || (< Automotive) || (= PO)	
ABIE$_{21}$	Name: Address BC$_{overall}$ = (≤ Europe) || (≤ BCG2_Industry) || (= PO)	
	BBIE$_{24}$	Name: City BC$_{overall}$ = (≤ EU) || $\overline{(> \text{Automotive})}$ || (= PO)
	BBIE$_{25}$	Name: Postcode BC$_{overall}$ = (= EU) || (< Automotive) || (= PO)
	ASBIE$_{26}$	Name: Country BC$_{overall}$ = (≤ Italy) || (≤ BCG2_Industry) || (= PO)
	⋮	
ABIE$_{22}$	Name: Person BC$_{overall}$ = (≤ Europe) || ((≤ BCG1_Industry) || (≤ BCG2_Industry)) || (= PO)	
	BBIE$_{27}$	Name: Passport BC$_{overall}$ = (≤ Europe) || (= Automotive) || (= PO)
	ASBIE$_{28}$	Name: Country BC$_{overall}$ = (< Europe) || (≤ BCG2_Industry) || (= PO)
	⋮	
ABIE$_{23}$	Name: Product BC$_{overall}$ = (< EU) || (≤ Automotive) || (= PO)	
	⋮	
⋮		

, …,

BDocIG$_{(n-1)}$	BC$_{assigned}$ = (≤ EU) || (= BCG2_Industry) || (= PO)	
ABIE$_{(n-1)1}$	Name: Metrics BC$_{overall}$ = (≤ EU) || ((≤ BCG1_Industry) || (≤BCG2_Industry)) || (= PO)	
	⋮	
ABIE$_{(n-1)2}$	Name: Product BC$_{overall}$ = (< Europe) || (= BCG2_Industry) || (= PO)	
	⋮	
⋮		

Figure 6.6: Example - existing business document implementation guidelines (business contextualization by means of the Enhanced UN/CEFACT Model)

The *BDocIG*$_1$ comprises the *ABIE*$_{12}$ and the *BDocIG*$_2$ comprises the *ABIE*$_{21}$. These ABIEs originate from the same ACC (named *Address. Details*), but they are *derived by restrictions* based on different BCs. According to our approach, the *ABIE*$_{12}$ and *BBIE*$_{17}$ are re-used and plugged into the *CustBDocIG*. However, the *BBIE*$_{24}$ covered by the *ABIE*$_{21}$ from the guideline *BDocIG*$_2$ is also relevant in the requested BC (*direct BC match*: $BC_{req} \subseteq BC_{ov\_24}$). Thus, it is added into the *CustBDocIG* in the scope of the re-used *ABIE*$_{12}$ which originates from the previously processed implementation guideline *BDocIG*$_1$.

Finally, the *ASBIE*$_{26}$ located in the *ABIE*$_{21}$ is valid in the geopolitical BC domain $(\leq Italy)$ which can not be resolved using our E-UCM business context graph presented in Figure 6.1. Therefore, in contrary to the situation explained in the following Subsection, the E-UCM business context model **can not be used** to resolve the BC node denoted by *Italy*. Hence, as shown in Figure 6.7, Mark 2, the ASBIE$_{26}$ is omitted from the customized BDocIG.

Generic-BDocIG	$BC_{assigned}$ = (≤ BCG) ❶	
$ABIE_{11}$	Name: Product $BC_{overall}$ = (≤ World) || (≤ BCG2_Industry) || (= PO)	
	$BBIE_{14}$	Name: Type $BC_{overall}$ = (≤ World) || (< Automotive) || (= PO)
	$BBIE_{15}$	Name: Status $BC_{overall}$ = (≤ World) || $\overline{(> \text{Automotive})}$ || (= PO)
	$BBIE_{16}$	Name: Color $BC_{overall}$ = ((≤ Germany) || (= NonEU)) || (≤ BCG2_Industry) || (= PO)
	⋮	
$ABIE_{12}$	Name: Address $BC_{overall}$ = (< World) || ((= BCG1_Metal) || (= BCG2_Metal)) || (= PO)	
	$BBIE_{17}$	Name: Postcode $BC_{overall}$ = (≤ EU) || ((= BCG1_Metal) || (= BCG2_Metal)) || (= PO)
	$ASBIE_{18}$	Name: Country $BC_{overall}$ = (≤ Germany) || (≤ BCG2_Metal) || (= PO)
	⋮	
$ABIE_{13}$	Name: Transaction $BC_{overall}$ = (≤ Germany) || (≥ BCG2_Metal) || (= PO)	
	⋮	
⋮		
$ABIE_{21}$	Name: Address $BC_{overall}$ = (≤ Europe) || (≤ BCG2_Industry) || (= PO)	
	$BBIE_{24}$	Name: City $BC_{overall}$ = (≤ EU) || $\overline{(> \text{Automotive})}$ || (= PO)
	$BBIE_{25}$	Name: Postcode $BC_{overall}$ = (= EU) || (< Automotive) || (= PO)
	$ASBIE_{26}$	Name: Country $BC_{overall}$ = (≤ Italy) || (≤ BCG2_Industry) || (= PO)
	⋮	
$ABIE_{22}$	Name: Person $BC_{overall}$ = (≤ Europe) || ((≤ BCG1_Industry) || (≤ BCG2_Industry)) || (= PO)	
	$BBIE_{27}$	Name: Passport $BC_{overall}$ = (≤ Europe) || (= Automotive) || (= PO)
	$BBIE_{28}$	Name: Country $BC_{overall}$ = (< Europe) || (≤ BCG2_Industry) || (= PO)
	⋮	
$ABIE_{23}$	Name: Product $BC_{overall}$ = (< EU) || (≤ Automotive) || (= PO)	
⋮		
$ABIE_{(n-1)1}$	Name: Metrics $BC_{overall}$ = (≤ EU) || ((≤ BCG1_Industry) || (≤BCG2_Industry)) || (= PO)	
	⋮	
$ABIE_{(n-1)2}$	Name: Product $BC_{overall}$ = (< Europe) || (= BCG2_Industry) || (= PO)	
	⋮	
⋮		

Cust-BDocIG	BC_{req} = (= EU) || (= BCG2_Metal) || (= PO) ❷	
$ABIE_{11}$	Name: Product $BC_{overall}$ = (≤ World) || (≤ BCG2_Industry) || (= PO)	
	$BBIE_{15}$	Name: Status $BC_{overall}$ = (≤ World) || $\overline{(> \text{Automotive})}$ || (= PO)
	$BBIE_{16}$	Name: Color $BC_{overall}$ = ((≤ Germany) || (= NonEU)) || (≤ BCG2_Industry) || (= PO)
	⋮	
$ABIE_{12}$	Name: Address $BC_{overall}$ = (< World) || (≤ BCG2_Metal) || (= PO)	
	$BBIE_{17}$	Name: Postcode $BC_{overall}$ = (≤ EU) || ((= BCG1_Metal) || (= BCG2_Metal)) || (= PO)
	$BBIE_{24}$	Name: City $BC_{overall}$ = (≤ EU) || $\overline{(> \text{Automotive})}$ || (= PO)
	⋮	
⋮		
$ABIE_{22}$	Name: Person $BC_{overall}$ = (≤ Europe) || ((≤ BCG1_Industry) || (≤ BCG2_Industry)) || (= PO)	
	$BBIE_{27}$	Name: Passport $BC_{overall}$ = (≤ Europe) || (= Automotive) || (= PO)
	$BBIE_{28}$	Name: Country $BC_{overall}$ = (< Europe) || (≤ BCG2_Industry) || (= PO)
	⋮	
⋮		
$ABIE_{(n-1)1}$	Name: Metrics $BC_{overall}$ = (≤ EU) || ((≤ BCG1_Industry) || (≤BCG2_Industry)) || (= PO)	
	⋮	
⋮		

Figure 6.7: Example - (i) generic business document implementation guideline and (ii) customized business document implementation guideline; (business contextualization by means of the Enhanced UN/CEFACT Model)

Example - Application of Scenario A Using the BCOnt Model. The following example shows how our conceptual solution can be invoked in practice to generate a new BDocIG valid in a requested BC. A business contextual knowledge used in the example is represented by our exemplary BCOnt shown in Figure 6.2. The particular BC values are specified using the BC expressions, as already described in Chapter 5.1.2. The requested business context is expressed as $BC_{req} = (\equiv EU) \sqcup (\equiv Metal) \sqcup (\equiv PO)$. The *Partial BC matching indexes* are: $p = p_1 = p_2 = 1$. The *BC non-matching indexes* are: $n = n_1 = n_2 = 1$.

The BIE elements of the already existing $BDocIG_1$, $BDocIG_2$ and $BDocIG_{(n-1)}$ are presented in Figure 6.8. These guidelines are valid in their assigned business contexts ($BC_{assigned\_1}$, $BC_{assigned\_2}$ and $BC_{assigned\_(n-1)}$, respectively). The included BIEs are valid in their overall BCs which are calculated following the Formulas 5.1 and 5.2. These pieces of information are shown in Figure 6.8 in the corresponding fields.

According to *AlgA*, all BIEs located in the *ExistBDocIGs* are extracted and added into a *GenBDocIG*. The excerpt of the composed guideline is illustrated in Figure 6.9, Mark 1. Afterwards, in order to select only those BIEs which are relevant in the specified region, indus-

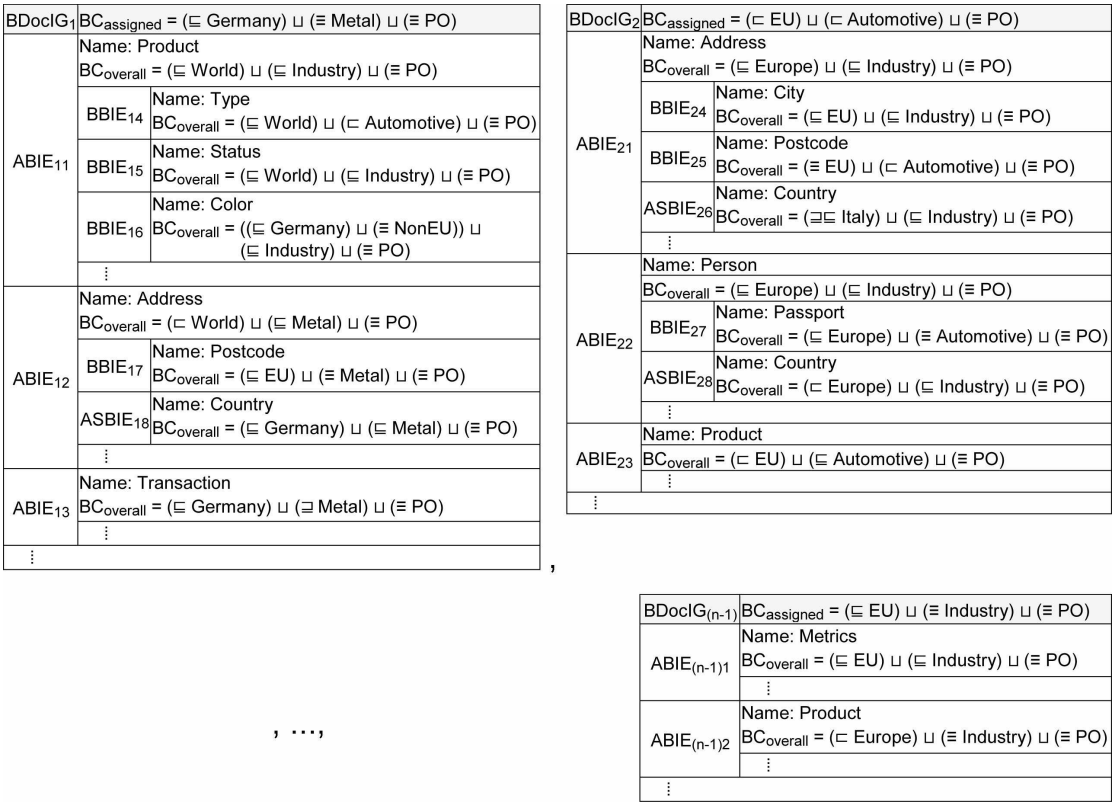


Figure 6.8: Example - existing business document implementation guidelines (business contextualization by means of the Business Context Ontology Model)

try and activity, we apply the requested BC under the *GenBDocIG*. The excerpt of the resulting *CustBDocIG* is presented in Figure 6.9, Mark 2.

For instance, the $BBIE_{15}$ located in the *GenBDocIG* is relevant in the requested BC (*direct BC match*: $BC_{req} \subseteq BC_{ov\_15}$). Hence, the corresponding BIEs ($BBIE_{15}$ and $ABIE_{11}$) are reused to generate the missing *CustBDocIG*. The explanations related to the $BBIE_{16}$, $BBIE_{14}$,

Generic-BDocIG	$BC_{assigned}$ = (⊑ World) ⊔ (⊑ Industry) ⊔ (⊑ Activity) ❶	
$ABIE_{11}$	Name: Product $BC_{overall}$ = (⊑ World) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	$BBIE_{14}$	Name: Type $BC_{overall}$ = (⊑ World) ⊔ (⊏ Automotive) ⊔ (≡ PO)
	$BBIE_{15}$	Name: Status $BC_{overall}$ = (⊑ World) ⊔ (⊑ Industry) ⊔ (≡ PO)
	$BBIE_{16}$	Name: Color $BC_{overall}$ = ((⊑ Germany) ⊔ (≡ NonEU)) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
$ABIE_{12}$	Name: Address $BC_{overall}$ = (⊏ World) ⊔ (⊑ Metal) ⊔ (≡ PO)	
	$BBIE_{17}$	Name: Postcode $BC_{overall}$ = (⊑ EU) ⊔ (≡ Metal) ⊔ (≡ PO)
	$ASBIE_{18}$	Name: Country $BC_{overall}$ = (⊑ Germany) ⊔ (⊑ Metal) ⊔ (≡ PO)
	⋮	
$ABIE_{13}$	Name: Transaction $BC_{overall}$ = (⊑ Germany) ⊔ (⊒ Metal) ⊔ (≡ PO)	
	⋮	
⋮		
$ABIE_{21}$	Name: Address $BC_{overall}$ = (⊑ Europe) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	$BBIE_{24}$	Name: City $BC_{overall}$ = (⊑ EU) ⊔ (⊑ Industry) ⊔ (≡ PO)
	$BBIE_{25}$	Name: Postcode $BC_{overall}$ = (≡ EU) ⊔ (⊏ Automotive) ⊔ (≡ PO)
	$ASBIE_{26}$	Name: Country $BC_{overall}$ = (⊒⊑ Italy) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
$ABIE_{22}$	Name: Person $BC_{overall}$ = (⊑ Europe) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	$BBIE_{27}$	Name: Passport $BC_{overall}$ = (⊑ Europe) ⊔ (≡ Automotive) ⊔ (≡ PO)
	$BBIE_{28}$	Name: Country $BC_{overall}$ = (⊏ Europe) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
$ABIE_{23}$	Name: Product $BC_{overall}$ = (⊏ EU) ⊔ (⊑ Automotive) ⊔ (≡ PO)	
⋮		
$ABIE_{(n-1)1}$	Name: Metrics $BC_{overall}$ = (⊑ EU) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	⋮	
$ABIE_{(n-1)2}$	Name: Product $BC_{overall}$ = (⊏ Europe) ⊔ (≡ Industry) ⊔ (≡ PO)	
	⋮	
⋮		

Cust-BDocIG	BC_{req} = (≡ EU) ⊔ (≡ Metal) ⊔ (≡ PO) ❷	
$ABIE_{11}$	Name: Product $BC_{overall}$ = (⊑ World) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	$BBIE_{15}$	Name: Status $BC_{overall}$ = (⊑ World) ⊔ (⊑ Industry) ⊔ (≡ PO)
	$BBIE_{16}$	Name: Color $BC_{overall}$ = ((⊑ Germany) ⊔ (≡ NonEU)) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
$ABIE_{12}$	Name: Address $BC_{overall}$ = (⊏ World) ⊔ (⊑ Metal) ⊔ (≡ PO)	
	$BBIE_{17}$	Name: Postcode $BC_{overall}$ = (⊑ EU) ⊔ (≡ Metal) ⊔ (≡ PO)
	$BBIE_{24}$	Name: City $BC_{overall}$ = (⊑ EU) ⊔ (⊑ Industry) ⊔ (≡ PO)
	$ASBIE_{26}$	Name: Country $BC_{overall}$ = (⊒⊑ Italy) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
⋮		
$ABIE_{22}$	Name: Person $BC_{overall}$ = (⊑ Europe) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	$BBIE_{27}$	Name: Passport $BC_{overall}$ = (⊑ Europe) ⊔ (≡ Automotive) ⊔ (≡ PO)
	$BBIE_{28}$	Name: Country $BC_{overall}$ = (⊏ Europe) ⊔ (⊑ Industry) ⊔ (≡ PO)
	⋮	
⋮		
$ABIE_{(n-1)1}$	Name: Metrics $BC_{overall}$ = (⊑ EU) ⊔ (⊑ Industry) ⊔ (≡ PO)	
	⋮	
⋮		

Figure 6.9: Example - (i) generic business document implementation guideline and (ii) customized business document implementation guideline; (business contextualization by means of the Business Context Ontology Model)

ABIE$_{12}$, *ABIE*$_{21}$, *BBIE*$_{17}$ and *BBIE*$_{24}$ are analogous to the explanations which are related to the corresponding BIEs discussed in our previous example (when the E-UCM business context model is used).

Finally, the *ASBIE*$_{26}$ is valid in the BC where its location subdomain ($\sqsupseteq\sqsubseteq$ *Italy*) can not be resolved using our exemplary BCOnt. However, as already described in Chapter 5, BCOnt can be interrelated with the external ontologies where a missing concept is defined. For instance, the external ontology named DBpedia [73, 80], in contrary to our exemplary BCOnt, defines the concept marked *Italy* which participates in the following property: (*dbpedia:Italy*, *rdf:type*, *yago:EuropeanUnionMemberState*). Therefore, DBpedia provides the business contextual knowledge that the concept marked *Italy* (not defined in our exemplary BCOnt) is a member of the concept marked *European Union* (equivalent to the concept marked *EU* in BCOnt). Thereby, in contrary to the E-UCM business context model (shown in the previous example), our BCOnt business context model **can be used** to resolve the missing concept (denoted by *Italy* in this example). Consequently, as shown in Figure 6.9, Mark 2, the *ASBIE*$_{26}$ is valid in our requested BC (*direct BC match*: $BC_{req} \subseteq BC_{ov\_26}$), and, thus, can be re-used for the generation of the missing BDocIG.

6.4 Mapping of Business Information Entities

Before we can explain our conceptual solution of Scenario B described by Formula 6.2, we first must define the *BIEs mapping*. Therefore, in the following we distinguish between two types of the BIEs mapping: (i) *Vertical mapping* and (ii) *Horizontal mapping*.

We consider the Business Information Entities: the BIE_1 located in the $BDocIG_1$, the BIE_2 located in the $BDocIG_2$ and the BIE_3 located in the $BDocIG_3$. They are valid in the following business contexts: $BC_1 = (ctg_{11}, ctg_{12}, ..., ctg_{1c})$, $BC_2 = (ctg_{21}, ctg_{22}, ..., ctg_{2c})$ and $BC_3 = (ctg_{31}, ctg_{32}, ..., ctg_{3c})$, respectively. c represents the total number of the primary business context categories. As already explained, in our particular work we distinguish between location, industry and activity primary BC categories. Thus, in the following examples we consider that $c = 3$. ctg_{ml} is the BC expression (explained in Chapter 4.2.2 and in Chapter 5.1.2 for cases using our E-UCM model and our BCOnt model, respectively) which identifies the domain of the corresponding primary business context category where the BIE_m is valid ($1 \leq m \leq 3$ and $1 \leq l \leq c$).

Definition 6.4.1 : Vertical Mapping. There is the *vertical mapping in respect to the primary BC category identified by its index l* between the BIE_1 and the BIE_2, represented by $VM^l(BIE_1, BIE_2)$, iff (i) the $BDocIG_1$ and the $BDocIG_2$ are paired: $P(BDocIG_1, BDocIG_2)$; (ii) the BIE_1 and the BIE_2 are *derived by restriction* from the same Core Component which is denoted by C: $DbR(BIE_1, C) \wedge DbR(BIE_2, C)$; and (iii) the ctg_{1l} and the ctg_{2l} are not matched: $ctg_{1l} \not\subseteq ctg_{2l}$. In the following we especially consider the *vertical mapping in respect to the activity primary BC category*, and, thus, we refer to this type of mapping only as the *vertical mapping*. It is represented by $VM(BIE_1, BIE_2)$.

For instance, the example of the *vertical mapping* between the BIE_{p11} and the BIE_{i11} is shown in Figure 6.10, Mark 1. The used BC is represented by our BCOnt model where the exemplary ontology is illustrated in Figure 6.2. (i) Both BIEs belong to the paired BDocIGs ($POIG_1$ and $INIG_1$, respectively); (ii) both BIEs are *derived by restriction* from the same CC named *Address. Details*; and (iii) the activity BC domains - in which the BIE_{p11} and the BIE_{i11} are valid - are non-matched.

Definition 6.4.2 : Horizontal Mapping. There is the *horizontal mapping in respect to the primary BC category identified by its index l* between the BIE_1 and the BIE_2, represented by $HM^l(BIE_1, BIE_2)$, iff (i) the $BDocIG_1$ and the $BDocIG_2$ are not paired: $\neg P(BDocIG_1, BDocIG_2)$; (ii) the BIE_1 and the BIE_2 are *derived by restriction* from the same Core Component which is denoted by C: $DbR(BIE_1, C) \wedge DbR(BIE_2, C)$; and (iii) the ctg_{1l} and the ctg_{2l} are matched: $(ctg_{1l} \subseteq ctg_{2l}) \vee (ctg_{1l} \subseteq^i ctg_{2l}) \vee (ctg_{1l} \subseteq^p ctg_{2l})$. In the following we especially consider the *horizontal mapping in respect to the activity primary BC category*, and, thus, we refer to this type of mapping only as the *horizontal mapping*. It is represented by $HM(BIE_1, BIE_2)$.

For instance, the example of the *horizontal mapping* between the BIE_{p21} and the BIE_{p11} is shown in Figure 6.10, Mark 2. The used BC is represented by our BCOnt model where the exemplary ontology is illustrated in Figure 6.2. (i) Both BIEs belong to the non-paired guidelines ($POIG_1$ and $POIG_2$, respectively); (ii) both BIEs are *derived by restriction* from the same CC named *Address. Details*; and (iii) the activity BC domains - in which the BIE_{p21} and the BIE_{p11} are valid - are directly matched.

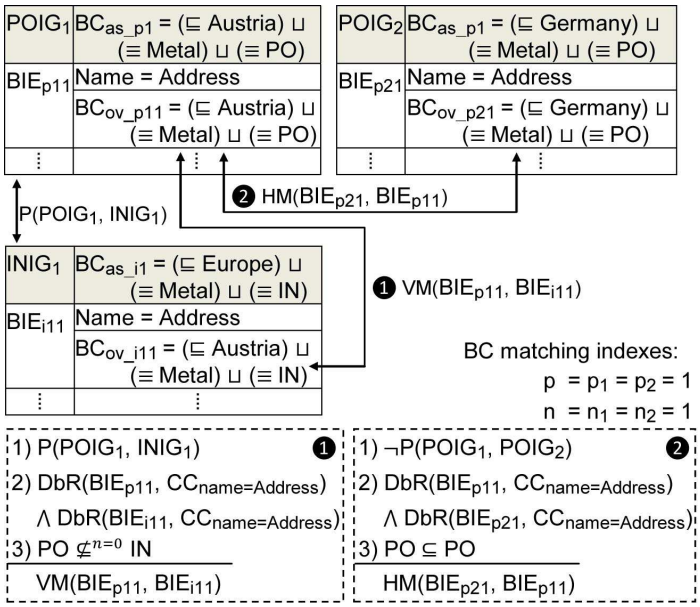


Figure 6.10: Vertical and horizontal mapping

6.5 Scenario B - Conceptual Solution

In the following we explain the conceptual solution of Scenario B described by Formula 6.2 and illustrated in Figure 6.4. $AlgB$ consists of four main execution steps: (i) application of *AlgA*, (ii) application of the *1:1 BIEs mapping*, (iii) application of the *1:m BIEs mapping* and (iv) application of the *additional BC reasoning rules*.

We show the application of the executions steps of *AlgB* using the excerpt of the real-world example presented in Figure 6.11. Accordingly, there is the set of the paired business document implementation guidelines: $(P(POIG_1, INIG_1)$, $P(POIG_2, INIG_2)$ and $P(POIG_3, INIG_3))$. The $POIG_4$ is the guideline which paired $INIG_4$ is missing. The guidelines and their included BIEs are valid in the business contexts which are shown in Figure 6.11 in the corresponding fields. The business context is represented in the example using our BCOnt model. The application of our E-UCM business context model is completely analogous. The *Partial BC matching indexes* are: $p = p_1 = p_2 = 1$. The *BC non-matching indexes* are: $n = n_1 = n_2 = 1$.

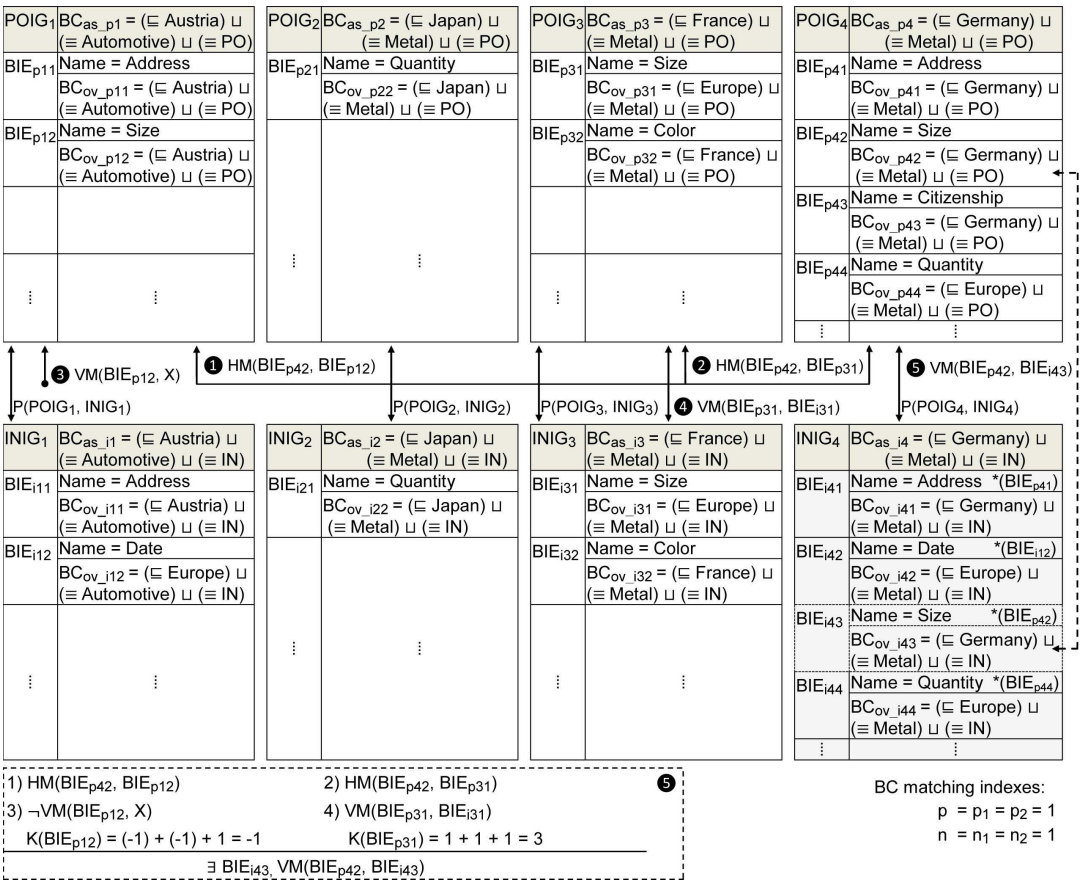


Figure 6.11: Example - scenario B

6.5.1 Algorithm B - Application of Algorithm A

Application of AlgA is the processing step of *AlgB* where the knowledge which originates from the paired relationships established between the corresponding POIGs and INIGs is not considered for the generation of the missing INIG. In contrary to other processing steps of *AlgB*, this processing step transforms Scenario B (Figure 6.12, Mark 1) to Scenario A (Figure 6.12, Mark 2). Therefore, it processes the paired guidelines independently involving *AlgA* (explained earlier in this Chapter) to generate BIE contents of the new INIG. The developed INIG can be additionally tailored by utilizing the business contextual knowledge which originates from the paired relationships typical only for Scenario B. This is performed by the upcoming execution steps of *AlgB* which consider the original Scenario B and do not transform it to Scenario A.

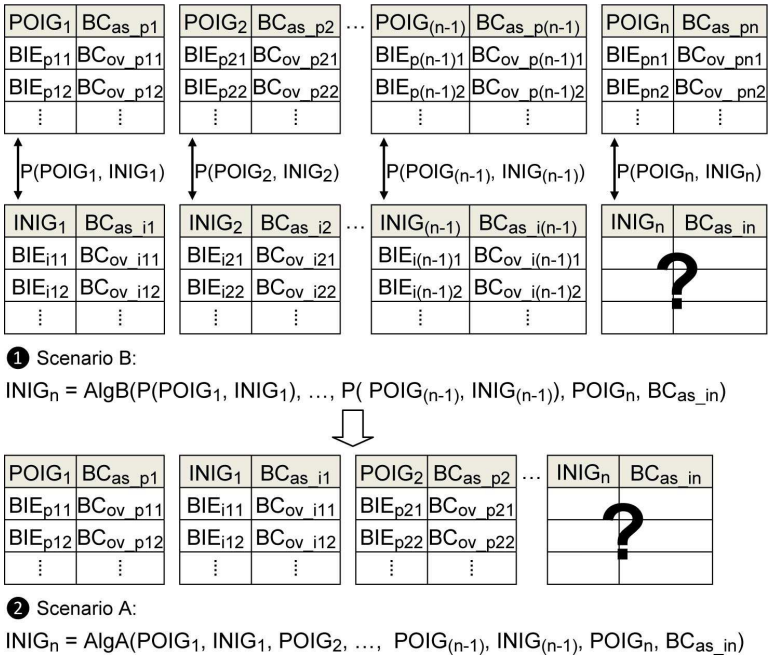


Figure 6.12: Example - transformation to scenario A

Application of Algorithm A - Example. This processing step of *AlgB* is illustrated in our already introduced example shown in Figure 6.11. For instance, let us consider the BIE_{i12} located in the $INIG_1$. The geopolitical and activity BC domains in which the $INIG_4$ should be valid (($\sqsubseteq Germany$) and ($\equiv IN$), respectively) are *directly matched* to the corresponding BC domains in which the BIE_{i12} is valid (($\sqsubseteq Europe$) and ($\equiv IN$), respectively). Furthermore, the industry BC domain in which the $INIG_4$ should be valid ($\equiv Metal$) is *partially matched* ($p = 1$) to the industry BC domain in which the BIE_{i12} is valid ($\equiv Automotive$). Therefore, according to *AlgA*, the BIE_{i12} should be added into the missing guideline $INIG_4$. This is represented by the BIE_{i42} in the example.

6.5.2 Algorithm B - 1:1 Business Information Entities Mapping

1:1 BIEs Mapping is the processing step of *AlgB* where the knowledge which originates from the paired relationships established between the corresponding POIGs and INIGs is used to generate the BIE contents of the missing INIG. In the following the BIE_{pn} represents the BIE located in the $POIG_n$. The $POIG_n$ is the existing guideline which paired $INIG_n$ is missing (shown in Figure 6.4). The BIE_{pi} and the BIE_{ii} are the BIEs which belong to the paired $POIG_i$ and $INIG_i$, respectively. The pseudo-code which implements the *1:1 BIEs Mapping* is shown in Algorithm 6.2.

We define the *1:1 BIEs Mapping* in the following. (i) If there exists the *horizontal mapping* between the BIE_{pn} and the BIE_{pi}: $HM_i = HM(BIE_{pn}, BIE_{pi})$ (Algorithm 6.2, Line 4); (ii) if the BIE_{pn} participates in exactly one *horizontal mapping*: $HM(BIE_{pn}, BIE_{pi}) \wedge HM(BIE_{pn}, BIE_{pv}) \Rightarrow i = v$ (Algorithm 6.2, Line 10); and (iii) if there exists the *verti-*

Input: $set(P(POIG, INIG)), POIG_n, BC_{req}, p_1, p_2, n_1, n_2$
Output: $IOIG_n$

```
1:  for each bie_pn : POIG_n do
2:      for each P_i(POIG_i, INIG_i) : set(P(POIG, INIG)) do
3:          for each bie_pi : POIG_i do
4:              if HM(bie_pn, bie_pi, p_1, p_2) then
5:                  hmPairs.add(bie_pi, P_i(POIG_i, INIG_i));
6:                  break;
7:              end if
8:          end for
9:      end for
10:     if hmPairs.length() == 1 then
11:         for each bie_ii : hmPairs[0].INIG_i do
12:             if VM(hmPairs[0].bie_pi, bie_ii, p_1, p_2) then
13:                 BIE bie_in = new BIE(VM(bie_pn, p_1, p_2));
14:                 INIG_n.addBIE(bie_in);
15:                 break;
16:             end if
17:         end for
18:     else if hmPairs.length() > 1 then
19:         set(bie_pq) = hmPairs.getAllBIE_pi();
20:         fVM = false;
21:         for each bie_pq : set(bie_pq) do
22:             if VM(bie_pq, bie_iq) then
23:                 fVM = true;
24:                 break;
25:             end if
26:         end for
27:         if fVM then
28:             goto: Algorithm 6.3, Line 2
29:         end if
30:     end if
31: end for
32: return INIG_n;
```

Algorithm 6.2: AlgB - 1:1 Business Information Entities Mapping

cal mapping between the BIE_{pi} and some BIE_{ii} (Algorithm 6.2, Line 12); then there exists one BIE_{in} such that the BIE_{pn} is *vertically mapped* to this BIE_{in}: $VM(BIE_{pi}, BIE_{ii}) \Rightarrow \exists(BIE_{in}), VM(BIE_{pn}, BIE_{in})$ (Algorithm 6.2, Lines 13-14).

1:1 Business Information Entities Mapping - Example. The example of the *1:1 BIEs Mapping* is shown in Figure 6.13. Accordingly, the business contextual knowledge which originates from the $P(POIG_1, INIG_1)$ and the $POIG_2$ is used to generate the missing content of the $INIG_2$ (paired with the $POIG_2$). The business context is presented using our E-UCM business context model where the corresponding E-UCM business context graph is illustrated in Figure 6.1. In particular, (i) there exists the *horizontal mapping* between the BIE_{p21} and the BIE_{p11}; (ii) the BIE_{p21} participates in exactly one *horizontal mapping*; and (iii) there exists the *vertical mapping* between the BIE_{p11} and the BIE_{i11}. Therefore, the $INIG_2$ contains the BIE_{i21} such that the BIE_{p21} is vertically mapped to the BIE_{i21}.

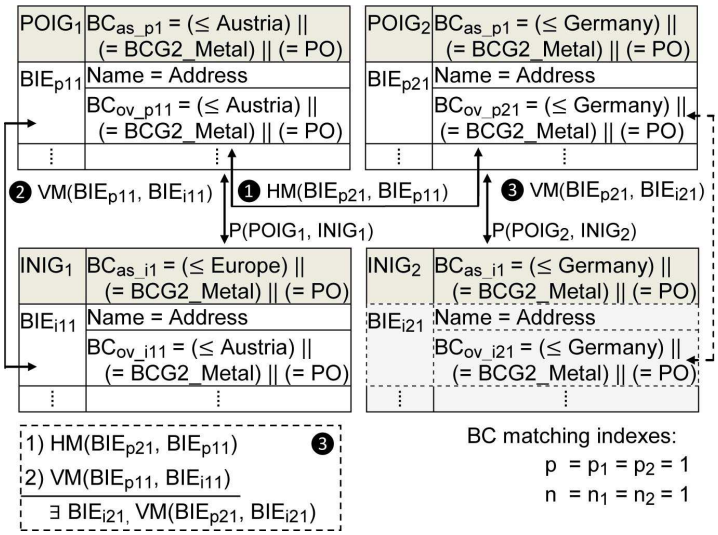


Figure 6.13: Example - 1:1 Business Information Entities mapping

In case that our concurrent BCOnt business context model is used, the application of the *1:1 BIEs Mapping* is analogous. For instance, the BIE_{i41} and the BIE_{i44} (contextualized by means of the BCOnt model) in the already introduced example shown in Figure 6.11 are generated following this processing step of *AlgB*.

6.5.3 Algorithm B - 1:m Business Information Entities Mapping

1:m BIEs Mapping is the processing step of *AlgB* where the knowledge which originates from the paired relationships established between the corresponding POIGs and INIGs is used to generate the unknown document implementation guideline. In the following BIE_{pn} represents the Business Information Entity located in the $POIG_n$. The $POIG_n$ is the existing guideline which paired $INIG_n$ is missing (shown in Figure 6.4). The BIE_{pi} and the BIE_{ii} are the Business Information Entities which belong to the paired $POIG_i$ and $INIG_i$, respectively.

The BIE_{pv} is the Business Information Entity located in the already existing $POIG_v$ (paired with the already existing $INIG_v$). The pieces of the pseudo-code which implement the *1:m BIEs Mapping* are shown in Algorithm 6.2 and Algorithm 6.3.

We define the *1:m BIEs Mapping* in the following. (i) if there exists the *horizontal mapping* between the BIE_{pn} and the BIE_{pi}: $HM_i = HM(BIE_{pn}, BIE_{pi})$ (Algorithm 6.2, Line 4); (ii) if the BIE_{pn} participates in more than one *horizontal mappings*, where the $set(BIE_{pq})$ represents the set of all BIEs which are *horizontally mapped* with the BIE_{pn}: $HM(BIE_{pn}, BIE_{pi}) \wedge HM(BIE_{pn}, BIE_{pv}) \not\Rightarrow i = v$ (Algorithm 6.2, Line 18); and (iii) if there exists at least one BIE (denoted by BIE_{pq}) from the $set(BIE_{pq})$ which is *vertically mapped* to some other BIE (denoted by BIE_{iq}) (Algorithm 6.2, Line 27); then there might exist one BIE_{in} such that the BIE_{pn} is *vertically mapped* to this BIE_{in}: $\exists(BIE_{pq}), HM(BIE_{pn}, BIE_{pq}) \wedge VM(BIE_{pq}, BIE_{iq}) \rightarrow \exists(BIE_{in}), VM(BIE_{pn}, BIE_{in})$ (Algorithm 6.2, Line 28).

There are two options for the following execution steps of *AlgB*. (i) In case that each BIE from the $set(BIE_{pq})$ participates in the *vertical mapping*, there exists one BIE_{in} such that the BIE_{pn} is *vertically mapped* to this BIE_{in}: $\forall BIE_{pq} \wedge BIE_{pq} \in set(BIE_{pq}), \exists BIE_{iq}, HM(BIE_{pn}, BIE_{pq}) \wedge VM(BIE_{pq}, BIE_{iq}) \rightarrow \exists(BIE_{in}), VM(BIE_{pn}, BIE_{in})$ (Algorithm 6.3, Lines 9-11). However, (ii) in case that some of the BIEs (at least one) which belong to the $set(BIE_{pq})$ participate while other BIEs (at least one) from the same set do not participate in the *vertical mappings*, the further calculation is requested. Before we can determinate whether the BIE_{in} exists or does not exist, first we must introduce: (i) *BC matching index*, (ii) *Partitive*

Input: $set(P(POIG, INIG)), POIG_n, BC_{req}, p_1, p_2, n_1, n_2$
Output: $IOIG_n$
1: ... Algorithm 6.2, Lines 1-28
2: $fVM = true$;
3: **for each** $bie_{pq} : set(bie_{pq})$ **do**
4: **if** $!VM(bie_{pq}, bie_{iq}, p_1, p_2)$ **then**
5: $fVM = false$;
6: $break$;
7: **end if**
8: **end for**
9: **if** fVM **then**
10: $BIE\ bie_{in} = new\ BIE(VM(bie_{pn}, p_1, p_2))$;
11: $INIG_n.add(bie_{in})$;
12: **else**
13: **for each** $bie_{pq} : set(bie_{pq})$ **do**
14: $k_1, k_2, k_3, = Formula\ 6.4(bie_{pq}, p_1, p_2, n_1, n_2)$
15: $set(K).add(k_1 + k_2 + k_3)$; {Formula 6.6}
16: **end for**
17: $bie_{pKmax} = max(set(K))$; {Formula 6.7}
18: **if** $VM(bie_{pKmax}, bie_{iKmax}, p_1, p_2)$ **then**
19: $BIE\ bie_{in} = new\ BIE(VM(bie_{pn}, p_1, p_2))$;
20: $INIG_n.add(bie_{in})$;
21: **end if**
22: **end if**
23: **return** $INIG_n$;

Algorithm 6.3: AlgB - 1:m Business Information Entities Mapping

BC matching index and (iii) *Maximal BC matching index*.

Business Context Matching Index. The *BC matching index* of the Business Information Entity BIE_1 in respect to the Business Information Entity BIE_2, denoted by $K(BIE_1, BIE_2)$, is defined by the following Formula:

$$K(BIE_1, BIE_2) = \sum_{j=1}^{c} K_j(BIE_1, BIE_2). \tag{6.3}$$

Therefore, the *BC matching index* is equal to the sum of the corresponding *Partitive BC matching indexes*. For instance, $K_j(BIE_1, BIE_2)$ represents the *Partitive BC matching index* of the BIE_1 in respect to the BIE_2 along the BC category identified by its index j. c represents the total number of the primary BC categories (c = 3 in our approach).

The *Partitive BC matching index* can be calculated by the following Formula:

$$K_j(BIE_1, BIE_2) = \begin{cases} 1, & BC_1^j \subseteq BC_2^j \\ -rw, & BC_1^j \subseteq^i BC_2^j \\ -max(p), & BC_1^j \subseteq^{p_1,p_2} BC_2^j \\ -\infty, & BC_1^j \not\subseteq^{n_1,n_2} BC_2^j \end{cases}. \tag{6.4}$$

The BC_1^j and the BC_2^j are domains of the BCs along the primary BC category identified by its index j in which the BIE_1 and the BIE_2 are valid, respectively. According to Formula 6.4, the *Partitive BC matching index* of the BIE_1 in respect to the BIE_2 along the BC category identified by its index j has the value 1 if the BC_1^j is *directly matched* to the BC_2^j. In case that the BC_1^j is *indirectly matched* to the BC_2^j, the corresponding *Partitive BC matching index* is equal to the negation of the *restriction weight* value (introduced earlier in this Chapter in Section 6.1.3) which is calculated between the indirectly matched concepts. In case that the BC_1^j is *partially matched* to the BC_2^j, the corresponding *Partitive BC matching index* is equal to the negation of the maximal *Partial BC matching index* (introduced earlier in this Chapter in Section 6.2). Finally, in case that the BC_1^j *can not be matched* to the BC_2^j, the *Partitive BC matching index* of the BIE_1 in respect to the BIE_2 along the BC category identified by its index j has the negative infinity value.

In particular, we distinguish between 3 primary BC categories ($c = 3$). Furthermore, the BIE_1 introduced by Formula 6.3 belongs to the POIG which paired INIG is missing. In our specific case, thus, the following holds: $BIE_1 = BIE_{pn}$. Hence, the *BC matching index* of the BIE_{pn} in respect to some BIE is calculated as the sum of the *Partitive BC matching indexes* of the BIE_{pn} in respect to the aforementioned BIE along the relevant primary BC categories. Therefore, the following Formulas can be derived from Formula 6.3:

$$K(BIE_{pn}, BIE) = K_1(BIE_{pn}, BIE) + K_2(BIE_{pn}, BIE) + K_3(BIE_{pn}, BIE), \tag{6.5}$$

or

$$K(BIE) = K_1(BIE) + K_2(BIE) + K_3(BIE). \tag{6.6}$$

Maximal Business Context Matching Index. In contrary to the step of *AlgB* where the *1:1 BIEs mapping* is processed, the *1:m BIEs mapping* correlates the BIE_{pn} to at least two BIEs located in different POIGs. Furthermore, these already existing BIEs can either be *vertically mapped* to the BIEs which are located in the paired INIGs, or they do not participate in any *vertical mapping*. In both cases, the *BC matching indexes* of the BIE_{pn} in respect to its *horizontally mapped* BIEs can be calculated using Formula 6.6. The *Maximal BC matching index*, denoted by K_{max}, is defined as the maximal of these calculated *BC matching indexes*. This can be specified by the following Formula:

$$K_{max} = max(set(K)), \tag{6.7}$$

where the $set(K)$ represents the set of the *BC matching indexes* which are calculated for the BIE_{pn} in respect to its *horizontally mapped* BIEs.

The *Maximal BC Matching Index* identifies exactly one BIE (denoted by BIE_{pKmax}). There are two options for the following execution steps of *AlgB*: (i) In case that the BIE_{pKmax} is *vertically mapped* to some BIE_{iKmax}, there exists one BIE_{in} such that the BIE_{pn} is *vertically mapped* to it: $VM(BIE_{pKmax}, BIE_{iKmax}) \Rightarrow \exists(BIE_{in}), VM(BIE_{pn}, BIE_{in})$ (Algorithm 6.3, Lines 18-21). (ii) In case that the BIE_{pKmax} does not participate in any *vertical mapping*, there does not exist the BIE_{in} such that the BIE_{pn} is *vertically mapped* to it.

1:m Business Information Entities Mapping - Example. The *1:m BIEs mapping* is illustrated in our example in Figure 6.11. (i) The BIE_{p42} is *horizontally mapped* to the BIE_{p12} (Figure 6.11, Mark 1). (ii) The BIE_{p42} is *horizontally mapped* to the BIE_{p31} (Figure 6.11, Mark 2). (iii) The BIE_{p12} does not participate in any *vertical mapping* (Figure 6.11, Mark 3). (iv) The BIE_{p31} is *vertically mapped* to the BIE_{i31} (Figure 6.11, Mark 4). Therefore, in respect to *AlgB*, the missing $INIG_4$ might contain the BIE_{i42} such that BIE_{p42} is *vertically mapped* to it.

The *BC matching index* is calculated by Formula 6.6. Hence, $K(BIE_{p12}) = K_1(BIE_{p12}) + K_2(BIE_{p12}) + K_3(BIE_{p12})$. In particular, the BIE_{p42} is valid in its geopolitical BC domain ($\sqsubseteq Germany$) which is *partially matched* to the geopolitical BC domain where the BIE_{p12} is valid ($\sqsubseteq Austria$). The corresponding maximal *Partial BC matching index* is equal to 1 ($max(1,1) = 1$, $BC_{ref} = EU$). Formula 6.4, thus, implies that $K_1(BIE_{p12}) = -1$. Likewise, the BIE_{p42} is valid in its industry BC domain ($\equiv Metal$) which is *partially matched* to the industry BC domain where the BIE_{p12} is valid ($\equiv Automotive$). The corresponding maximal *Partial BC matching index* is equal to 1 ($max(1,1) = 1$, $BC_{ref} = Industry$). Thus, based on Formula 6.4, $K_2(BIE_{p12}) = -1$. Furthermore, the BIE_{p42} is valid in its activity BC domain ($\equiv PO$) which is *directly matched* to the activity BC domain where the BIE_{p12} is valid ($\equiv PO$). Consequently, according to Formula 6.4, $K_3(BIE_{p12}) = +1$. Finally, the *BC matching index* of the BIE_{p42} in respect to the BIE_{p12} can be calculated as: $K(BIE_{p12}) = (-1) + (-1) + 1 = -1$.

Similarly, Formula 6.6 implies the following: $K(BIE_{p31}) = K_1(BIE_{p31}) + K_2(BIE_{p31}) + K_3(BIE_{p31})$. The BIE_{p42} is valid in its geopolitical BC domain ($\sqsubseteq Germany$) which is *directly matched* to the geopolitical BC domain where the BIE_{p31} is valid ($\sqsubseteq Europe$). Thus, based on Formula 6.4, $K_1(BIE_{p31}) = +1$. Likewise, the BIE_{p42} is valid in its industry BC domain ($\equiv Metal$) which is *directly matched* to the industry BC domain where the BIE_{p31}

is valid ($\equiv Metal$). Consequently, according to Formula 6.4, $K_2(BIE_{p12}) = +1$. Furthermore, the BIE_{p31} is valid in its activity BC domain ($\equiv PO$) which is *directly matched* to the activity BC domain where the BIE_{p31} is valid ($\equiv PO$). Therefore, Formula 6.4 implies that $K_3(BIE_{p31}) = +1$. Finally, the *BC matching index* of the BIE_{p42} in respect to the BIE_{p31} is: $K(BIE_{p31}) = 1 + 1 + 1 = 3$.

According to Formula 6.7, the corresponding *Maximal BC matching* index can be calculated as: $K_{max} = max(-1, 3) = 3$. Hence, the BIE_{p31} is identified ($BIE_{pKmax} = BIE_{p31}$). As shown in Figure 6.11, the BIE_{p31} is *vertically mapped* to the BIE_{i31}. Therefore, in respect to our *AlgB*, the missing implementation guideline $INIG_4$ contains the BIE_{i42} such that the BIE_{p42} is *vertically mapped* to it.

6.5.4 Algorithm B - Additional Business Context Reasoning Rules

Additional BC reasoning rules represent the processing step of *AlgB* where the *rule based reasoning* capabilities (explained earlier in Section 6.3.3) are enhanced by the *additional BC reasoning rules*. These rules are explicitly defined by a user following the DL based syntax [92–94]. However, in contrary to the rules introduced earlier in Section 6.5.4, the *additional BC reasoning rules* can invoke the relationships which are specific only for Scenario B, such as the *paired relationship*, *vertical mapping* and *horizontal mapping*.

For example, if some POIG contains the Business Information Entity denoted by *B*, the reasoning mechanism can infer that its paired INIG also contains the Business Information Entity which is derived by restriction from the same Core Component as *B*. This can be expressed by the following *Additional BC reasoning rule*: *(?A bcont:hasMember ?B)* $\Longrightarrow$ *(?B bcont:hasVerticallMapping ?E)*.

Note that this Doctoral Thesis establishes only the foundations for business context reasoning. However, it neither discusses the performances of these techniques nor develops new approaches for their improvements. This is the scope of the future work.

6.6 Final Assessments

In this Chapter we presented our approach to calculate the content model of the business context aware electronic business document implementation guidelines. The implementation guidelines represent the business context aware restrictions of the underlying business document standard. Generally speaking, they are used to define the structures of the electronic business documents which are exchanged between business partners when executing inter-organizational business processes. More precisely, our research considers the electronic business documents which conform to CCTS (explained in Chapter 4). Accordingly, every business document comprises the set of interoperable data building blocks, so-called Core Components.

The main contribution of the presented work is the approach to model new electronic business document implementation guidelines by learning from the already existing contextualized models. Our approach utilizes the business contextual knowledge which comprises the circumstances (geopolitical region, industry and activity) where the electronic business documents are

or are not valid. The applied business context is represented using our concurrent business context models (the Enhanced UN/CEFACT Business Context Model developed in Chapter 4 and the Business Context Ontology Model developed in Chapter 5). The existing business document implementation guidelines - based on the CCTS document standard - can be contextualized by means of these models. In order to simplify the steps of our research, we distinguish between two main scenarios (Scenario A and Scenario B).

Scenario A consists of the set of the already existing contextualized documents which are valid in different business contexts. However, the business document - which should be valid in the business context specified by a user - is missing. The proposed algorithm applies the business contextual knowledge which is comprised by the existing documents. It detects from these documents only those Core Components which are valid in the requested business context, and, thus, can be re-used to generate the implementation guideline of the missing document.

Scenario B comprises the main problem statement of this Doctoral Thesis introduced in Chapter 1.3. Therefore, it consists of at least two business domains specified by different business contexts. In the first domain there are m ($m>1$) documents. In the second domain there are $m-1$ business documents, while one document is missing. For instance, let us assume that the first domain consists of the Purchase Order and Invoice documents, while the second domain consists only of the Purchase Order document. The proposed algorithm performs the horizontal and vertical mappings between the corresponding Core Components which are located in the existing documents. Based on the outcomes of these processing steps, the implementation guideline of the missing business document is generated (Invoice document implementation guideline in our example).

Furthermore, in the more complex case of Scenario B, the correlated documents might exist in more than one business context. In contrary to the previous situation, the processing of the horizontal and vertical mappings does not guarantee that some Core Components exists or does not exist in the missing guideline. Therefore, our algorithm applies the business context matching criteria for making the final decision.

The feasibility of our approach is demonstrated by a prototype implementation which is described in the succeeding Chapter 7. The evaluation is done following the *design science research methodology* [5]. The corresponding outcomes are presented in Chapter 8.

CHAPTER 7

Implementation

The previous Chapter 6 has explained our approach to apply the business contextual knowledge for (semi-) automatically generating implementation guidelines of electronic business documents. An implementation guideline represents a context specific constraint of the underlying generic document standard. As already stipulated, this Doctoral Thesis considers the electronic business documents built upon the Core Components Technical Specification (CCTS) (discussed in Chapter 2).

This Chapter presents the implementation of our business context aware Core Components modeling approach. In the following we additionally tailor this approach and demonstrate that it does not hold only in theory, but also in practice. Therefore, we implement the corresponding algorithms and show how these algorithms can be integrated in a prototype system. The proposed prototype exploits business contextual information to re-use the already existing Core Components during the development of new business context aware business document implementation guidelines. The main conclusions of this phase of our research are discussed in [15, 104, 111].

The remainder of this Chapter is structured as follows. First, we describe the XML based representation of contextualized business document implementation guidelines in Section 7.1. Afterwards, in Section 7.2 we present the overview of the prototype of our business context aware Core Components modeling approach. We elucidate clearly the key features of the most important services provided by the underlying architecture and describe how these services can be implemented by our proposed algorithms. Finally, the concluding remarks of this phase of our research are outlined in Section 7.3.

7.1 Representation of Business Context Aware Business Document Implementation Guidelines

In Chapter 2.5 we have described the XML Naming and Design Rules (NDR) [14]. This is a specification defined by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). It formulates the set of the rules necessary to develop XML schemas and

XML schema based documents which conform to the Core Components Technical Specification (CCTS) document standard. Thereby, in the following we present business context (BC) aware electronic business document implementation guidelines (BDocIGs) following the principles defined by the NDR specification.

We contextualize BC aware BDocIGs using our concurrent Enhanced UN/CEFACT Model (E-UCM) and Business Context Ontology Model (BCOnt) which are developed in Chapters 4 and 5, respectively. However, the standard NDR specification can not be directly applied to represent business documents which are contextualized by means of these models. Therefore, in order to provide an instrument to assign and to process business contextual information, the NDR specification must be enhanced. Our corresponding solution introduces the new XML schema element which is denoted as: *<ccts:BC>*. It is used to specify the concrete business context in which some specific Business Information Entity presented by the XML NDR schema is

```
<xsd:element name="PostCode" type="bdt:StringType">
  <xsd:annotation>
    <xsd:documentation>
      <ccts:AcronymCode>BBIE</ccts:AcronymCode>
      <ccts:UniqueID>f0e2b9ad-7d03</ccts:UniqueID>
        ...
    <xsd:documentation>
    <xsd:appInfo>
      <ccts:BC> ❶
        <ccts:GeoRegionBC>(~ Europe) Δ (≤ EU)</ccts:GeoRegionBC>
        <ccts:IndustryBC>(≤ Automotive)</ccts:IndustryClassificationBC>  ❷
        <ccts:ActivityBC>(= IN) || (= PO)</ccts:ActivityBC>
      </ccts:BC>
    </xsd:appInfo>
  </xsd:annotation>
</xsd:element>
```

Figure 7.1: Example - contextualized UN/CEFACT's XML Naming and Design Rules using the Enhanced UN/CEFACT Business Context Model

```
<xsd:element name="PostCode" type="bdt:StringType">
  <xsd:annotation>
    <xsd:documentation>
      <ccts:AcronymCode>BBIE</ccts:AcronymCode>
      <ccts:UniqueID>f0e2b9ad-7d05</ccts:UniqueID>
        ...
    <xsd:documentation>
    <xsd:appInfo>
      <ccts:BC> ❶
        <ccts:GeoRegionBC>(⊑ EU) ⊔ (≡ Switzerland)</ccts:GeoRegionBC>
        <ccts:IndustryBC>(⊒ Automotive) ⊓ (⊑ Industry)</ccts:IndustryClassificationBC>  ❷
        <ccts:ActivityBC>(≡ Invoice)</ccts:ActivityBC>
      </ccts:BC>
    </xsd:appInfo>
  </xsd:annotation>
</xsd:element>
```

Figure 7.2: Example - contextualized UN/CEFACT's XML Naming and Design Rules using the Business Context Ontology Model

valid. The introduced element is integrated in the scope of the application information element (*<xsd:appInfo>*) defined by the standard NDR. The relevant examples are shown in Figure 7.1, Mark 1 and in Figure 7.2, Mark 1.

Furthermore, the *<ccts:BC>* element comprises the following children elements: *<ccts: RegionBC>*, *<ccts:IndustryBC>* and *<ccts:ActivityBC>*. These are new XML schema elements which are correspondent to our primary BC categories geopolitical region, industry and activity, respectively. (The detailed specification of the primary BC categories is defined in Chapter 3.2). Thus, the subdomains of the business context - in which some specific Core Component is valid - can be presented by the BC expression indicated within the corresponding primary BC category tags. The case using our E-UCM buisness context model is presented in Figure 7.1, Mark 2 (the corresponding BC expression model is explained in Chapter 4.2.2), while the case using our BCOnt business context model is illustrated in Figure 7.2, Mark 2 (the corresponding BC expression model is explained in Chapter 5.1.2). Taking into account that Java supports Unicode identifier characters that are letters or digits, in our prototype we present the E-UCM and BCOnt predicates and operands using abbreviations of their names (such as LTE instead of the E-UCM symbol $\leq$ for the predicate Less than or equal to, and UNION instead of the BCOnt symbol $\sqcup$ for the operand Union).

7.2 Architecture

In the following we explain the simplified architecture which implements our approach to model business document implementation guidelines (BDocIGs) valid in the requested business context (BC). The corresponding blueprint and the explanation of its graphical notation are shown in Figure 7.3. All processing units and included libraries are developed using the Java programming language. The business document implementation guidelines conform to the UN/CEFACT

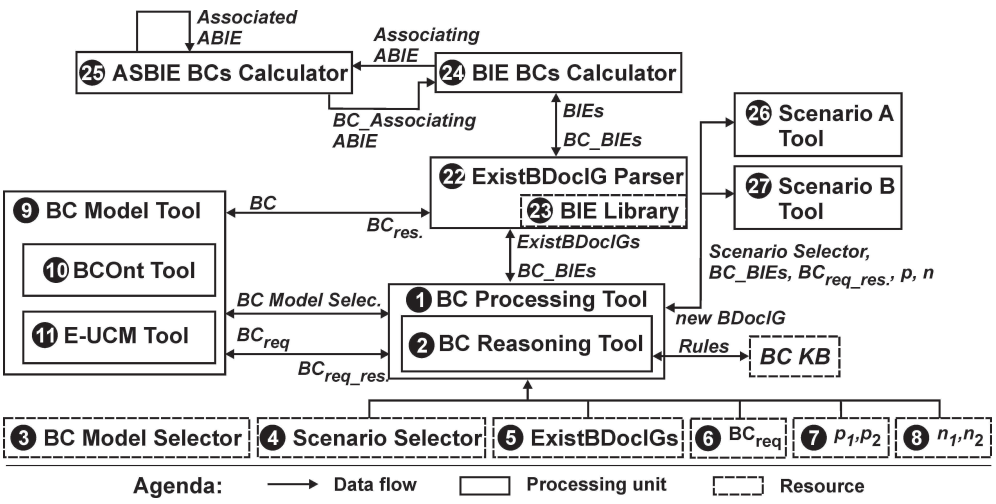


Figure 7.3: Business context aware Core Components modeling - overview of the architecture

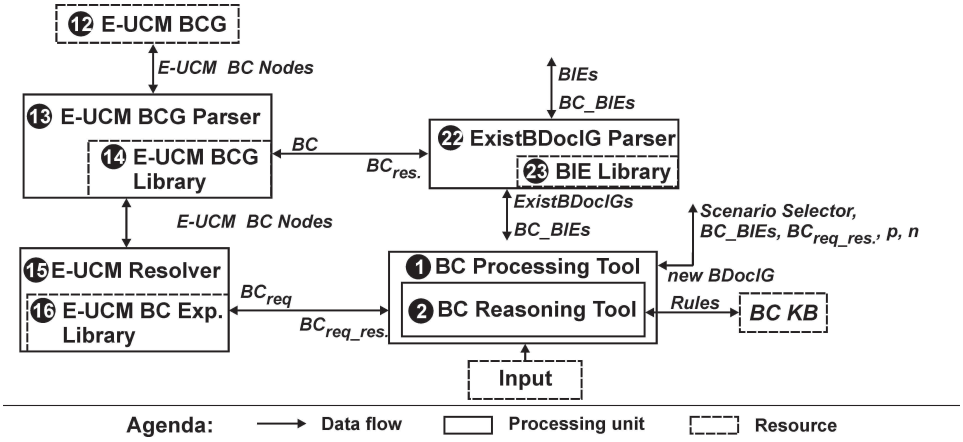


Figure 7.4: BC Model Tool - Enhanced UN/CEFACT Model processing elements

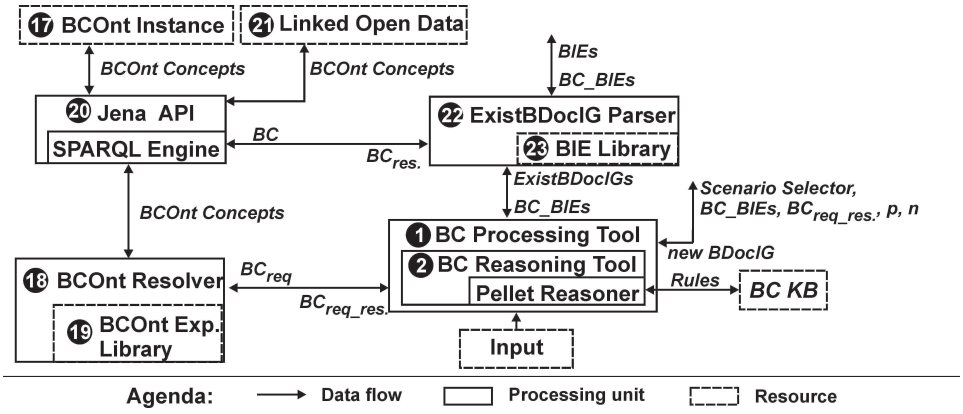


Figure 7.5: BC Model Tool - Business Context Ontology Model processing elements

Business Message Model explained in Chapter 2.6. They are represented using the enhanced NDR specification introduced in the previous Section.

The core of our prototypical architecture is the *Business Context Processing Tool* (Figure 7.3, Mark 1). It is the processing unit which initiates, controls and coordinates the executions of the other elements in the system. As shown in Figure 7.3, Mark 2, this tool comprises the *BC Reasoning Tool*. It is the processing unit which executes the BC reasoning techniques already introduced in Chapter 6 (Sections 6.3.3 and 6.5.4).

7.2.1 Input Processing

The following input parameters are processed by the proposed architecture: (i) *BC Model selector*, (ii) *instance* of the business context model, (iii) *Scenario selector*, (iv) already existing business document implementation guidelines (*ExistBDocIGs*), (v) business context in which the new guideline must be valid (BC_{req}), and (v) *BC matching indexes* (p_1, p_2, n_1 and n_2). In

the following we explain only those processing units which we consider to be most important for understanding our simplified architecture.

As already described in the previous Chapters, we have developed two concurrent models - Enhanced UN/CEFACT Model (E-UCM) and Business Context Ontology Model (BCOnt) - to represent business contextual information. The *BC Model selector* is the input parameter used to specify which of these models will be applied to process the business context. Both models are implemented within the *BC Model Tool* which is shown in Figure 7.3, Mark 9. The more detailed structure of this tool is presented in the Figures 7.4 and 7.5, and is explained in the following Subsections.

7.2.2 Enhanced UN/CEFACT Context Model Tool

The *E-UCM Tool* (Figure 7.4) is the unit of the *BC Model Tool* (Figure 7.3, Mark 9.) which processes the business contextual information represented by the E-UCM business context model

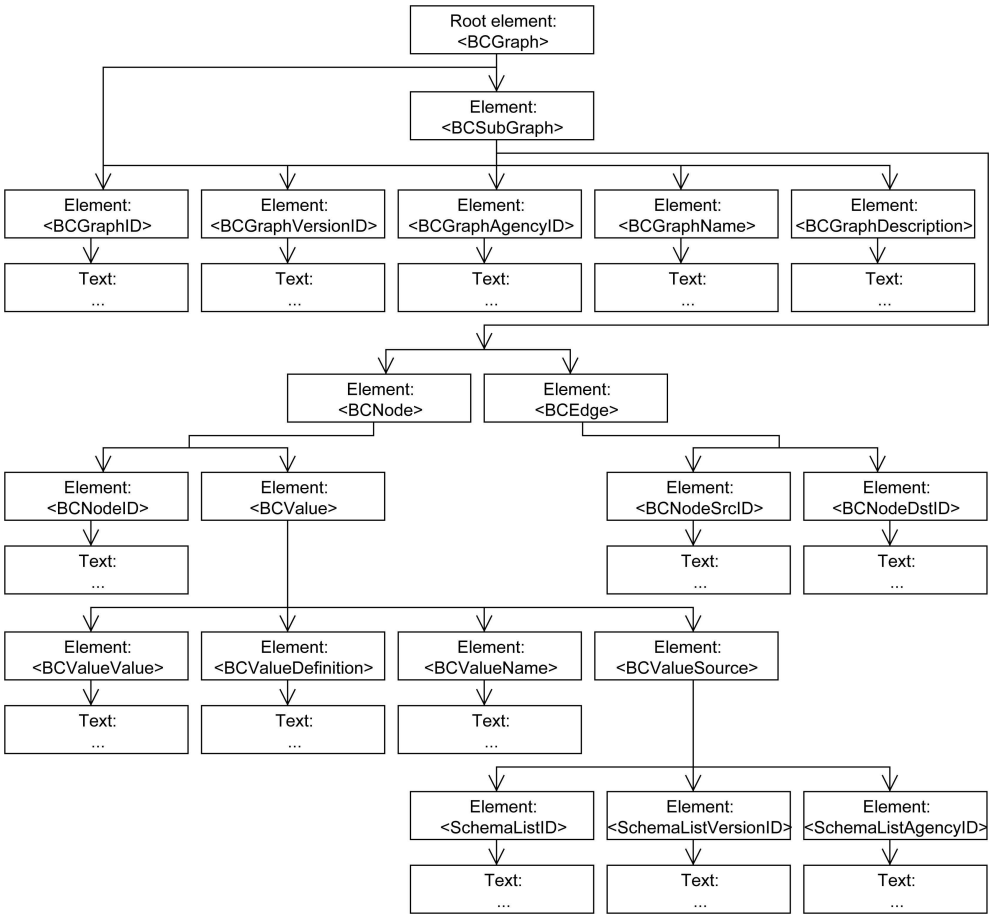


Figure 7.6: Enhanced UN/CEFACT business context graph - XML schema hierarchical structure

(explained in Chapter 4). The E-UCM BCG (Figure 7.4, Mark 12) denotes the Enhanced UN/CEFACT business context graph which is defined in Chapter 4.2.1. We represent the input E-UCM business context graph as the XML based document. The underlying XML schema tree-structure maps the elements of the UCM business context graph model which are explained in Chapter 4 and shown in Figure 4.1. The corresponding XML structure is illustrated in Figure 7.6. Furthermore, the particular E-UCM business context graph is resolved by the *E-UCM BCG Parser* (Figure 7.4, Mark 13). The included *E-UCM BCG Library* (Figure 7.4, Mark 14) represents our Java implementation of the E-UCM business context graph model.

The *E-UCM Resolver* (Figure 7.3, Mark 15) is the processing unit which checks the syntax correctness of the requested business context (BC_{req}) and identifies the specified BC nodes. The *E-UCM BC Expression Library* (Figure 7.4, Mark 16) represents our Java implementation of the E-UCM BC expression model explained in Chapter 4.2.2.

7.2.3 Business Context Ontology Tool

The *BCOnt Tool* (Figure 7.5) is the unit of the *BC Model Tool* (Figure 7.3, Mark 9.) which processes the business contextual information represented by the BCOnt model (explained in Chapter 5). BCOnt (Figure 7.5, Mark 17) is the OWL DL ontology [92–94] developed by the Protégé modeling tool [83] . Protégé is a free, open-source ontology editor and knowledge base framework. We have chosen Protégé due to its support of the OWL languages, plug in extension possibilities, built in reasoners, excellent documentation, user friendly interface and its ease of use.

As shown in Figure 7.5, Mark 2, the business context reasoning (explained in Chapter 5.1.3) is conducted by the *Pellet reasoner* [71]. It is an open source, sound and complete OWL DL reasoner written in Java and supported by the Protégé modeling tool. For instance, the *Pellet reasoner* can (i) be invoked to automatically classify business contextual knowledge, (ii) detect inconsistencies in the BCOnt ontology and (iii) derive new business contextual knowledge from the already existing knowledge by following the reasoning rules which are explicitly provided by the user. We highlight that the first and the second application of the BC reasoning are only possible in the case when the BCOnt model is used and **not** in the case when the concurrent E-UCM model is used.

The *BCOnt Resolver* (Figure 7.3, Mark 18) is the processing unit which checks the syntax correctness of the requested business context (BC_{req}) and identifies the specified concepts from BCOnt. As shown in Figure 7.5, Mark 19, the access to the ontology is realized by the *Jena Semantic Web Framework* [117]. It is a Java framework used for building Semantic Web applications. It is integrated with the *Pellet reasoner* and includes the *SPARQL Engine* [118, 119]. SPARQL is an RDF based query language which is applied to retrieve concepts specified by the BC_{req}. In case that some of these concepts are not defined by BCOnt, our ontology can be interconnected to external ontologies located in the scope of Linked Open Data (LOD) [73] where the missing concepts are defined (Figure 7.3, Mark 20). The corresponding piece of pseudo-code is presented in Algorithm 7.1 and explained in the following.

If the current processing concept can not be resolved directly from the BCOnt ontology (Algorithm 7.1, Line 2), a connection to the DBpedia ontology [73, 80] is established and the missing concept is queried through the SPARQL endpoint (Algorithm 7.1, Line 6). In case that

```
Input: BCOnt Concept names
Output: resolved BCOnt Concepts
 1: ...
 2: if !BCOnt.contains(conceptName) then
 3:     String service = "http://DBpedia.org/sparql";
 4:     String DBpediaUniqueID = "<http://Dbpedia.org/resource/" + conceptName+">";
 5:     String query = "SELECT ?X WHERE { " + DBpediaUniqueID +
        "<http//dbpedia.org/ontology/country> ?X}";
 6:     Result res = SPARQLService(service, query);
 7:     if res! = null then
 8:         OntConcept parentConcept = BCOnt.getConcept(res.getConceptName);
 9:         linking(parentConcept, DBpediaUniqueID);
10:     else
11:         print("Concept is not defined in DBpedia.");
12:     end if
13: end if
14: ...
```

Algorithm 7.1: Example: BCOnt - LOD interrelation

the missing concept can be refined invoking DBpedia (Algorithm 7.1, Line 7), linking between BCOnt and DBpedia is established (Algorithm 7.1, Line 9). More precisely, in the example implemented by Algorithm 7.1, the relationship between corresponding instances of the class *Country* defined by BCOnt (shown in Chapter 5, Figure 5.1) and the corresponding instances of the class *City* defined by DBpedia are set up. Thus, the business context which is expressed as ($\sqsupseteq Berlin$) and applied on the excerpt of BCOnt shown in Chapter 5, Figure 5.3 can now be resolved. It identifies the following concepts: *Berlin* (defined by DBpedia), *Germany*, *EU*, *Europe* and *World*. We highlight that the described procedure is only feasible in the case when the BCOnt model is used and **not** in the case when the concurrent E-UCM model is used.

7.2.4 Overall Business Context Calculation

ExistBDocIG Pareser. The *ExistBDocIG Pareser* (Figure 7.3, Mark 22) is the processing unit which extracts the already existing Business Information Entities (BIEs) located in the already existing business document implementation guidelines (ExistBDocIGs) (Figure 7.3, Mark 5). The included *BIE Library* (Figure 7.3, Mark 23) represents our Java implementation of the BIE models defined by CCTS (explained in Chapter 2 of this thesis).

BIE BCs Calculator. The *BIE BCs Calculator* (Figure 7.3, Mark 24) is the processing unit used to calculate the overall BCs in which the already existing (generic) Aggregate Business Information Entities (ABIEs) are valid. Therefore, it conducts the computations described by Chapter 4, Formula 4.2 (when the E-UCM model is used) and Chapter 5, Formula 5.1 (when the BCOnt model is used). The corresponding pseudo-code is presented in Algorithm 7.2.

The ABIEs contained in the list of the generic ABIEs are processed within the loop initiated in Algorithm 7.2, Line 1. According to Formulas 4.2 and 5.1, an ABIE is valid in the overall BC which is calculated as the union based on the following two components: (i) the union of the overall BCs in which its included BBIEs are valid, and (ii) the union of the overall BCs in which its included ASBIEs are valid. The first component of the overall BC of the currently processing

```
Input: ABIEList {BIE_overallBCs are not calculated.}
Output: ABIEList {BIE_overallBCs are calculated.}
 1: for each abie : ABIEList do
 2:     abie.overallBC = null;
 3:     for each bbie : abie.BBIEList do
 4:         bbie.overallBC = bbie.assignedBC;
 5:         abie.overallBC = Union(abie.overallBC, bbie.assignedBC);
 6:     end for
 7:     if abie.hasASBIEs then
 8:         for each asbie : abie.ASBIEList do
 9:             asbie.isOverallBCCalculated = false;
10:         end for
11:         abie.isOverallBCCalculated = false;
12:     else
13:         abie.isOverallBCCalculated = true;
14:     end if
15: end for
16: for each abie : ABIEList do
17:     if !abie.isOverallBCCalculated then
18:         abie.overallBC = Algorithm_73(abie);
19:         abie.isOverallBCCalculated = true;
20:     end if
21: end for
22: return ABIEList
```

Algorithm 7.2: BIE BCs Calculator

ABIE is calculated in Algorithm 7.2, Lines 3-6. If this ABIE does not contain any Association Business Information Entity (ASBIE), the second component of its overall BC is null. Thus, its previously calculated component of the BC is equal to its overall BC (Algorithm 7.2, Line 13). However, if the currently processing ABIE contains ASBIEs, the second component of its overall BC is not null, and it is calculated involving the *ASBIE BCs Calculator* (Algorithm 7.2, Line 18).

ASBIE BCs Calculator. The *ASBIE BCs Calculator* (Figure 7.3, Mark 25) is the unit used to calculate the overall BCs in which the ASBIEs contained by the generic ABIEs are valid. Therefore, it conducts the computations described by Chapter 4, Formula 4.3 (when the E-UCM model is used) and Chapter 5, Formula 5.2 (when the BCOnt model is used). It is implemented by the recursive algorithm which pseudo-code is presented in Algorithm 7.3.

The ABIE which comprises the currently processed ASBIEs (associating ABIE) is the input parameter of Algorithm 7.3. The ASBIEs contained by the input ABIE are handled within the loop initiated in Algorithm 7.3, Line 2. According to Formulas 4.3 and 5.2, the overall BC of an ASBIE is dependant, and, thus, calculated based on the intersection of its assigned BC and the overall BC of its associated ABIE. Therefore, there are two options (checked in Algorithm 7.3, Line 4) for the following execution steps of Algorithm 7.3: (i) the overall BC of the associated ABIE is still unknown, and (ii) the overall BC of the associated ABIE has already been calculated.

In this execution phase, the overall BC of the associated ABIE is unknown iff this ABIE contains at least one ASBIE which overall BC has not been processed yet. Therefore, Algorithm 7.3 is recursively called (Algorithm 7.3, Line 5) where the associated ABIE is indicated as the new

```
Input: abie {associating ABIE, overall BC is not calculated}
Output: abie {associating ABIE, overall BC is calculated}
 1: if !abie.isOvelrallBCCalculated then
 2:     for each asbie : abie.ASBIEList do
 3:         associatedABIE = asbie.associatedABIE();
 4:         if !associatedABIE.isOverallBCCalculated then
 5:             asbie.overallBC = Intersection(asbie.assignedBC, Algorithm_73(associatedABIE));
 6:         else
 7:             asbie.overallBC = Intersection(asbie.assignedBC, associatedABIE.overallBC);
 8:         end if
 9:         asbie.isOverallBCCalculated = true;
10:         abie.overallBC = Union(abie.overallBC, asbie.overallBC);
11:     end for
12:     abie.isOverallBCCalculated = true;
13:     ret = abie.overallBC;
14: else
15:     ret = abie.overallBC;
16: end if
17: return ret
```

Algorithm 7.3: ASBIE BCs Calculator

input parameter. In case that the overall BC of the associated ABIE is already known, the exit condition of the recursion is reached, and the overall BC of the currently processed ASBIE is calculated (Algorithm 7.3, Line 7). Finally, the previously calculated component of the BC in which the associating ABIE is valid is unionised with the overall BC in which the currently processing ASBIE is valid in Algorithm 7.3, Line 10.

7.2.5 Scenario Tools

The Scenario Tools process the already existing contextualized business document implementation guidelines and develop the new implementation guidelines which are valid in the BC_{req} (Figure 7.3, Mark 6). The *Scenario A Tool* (Figure 7.3, Mark 26) implements the algorithm *AlgA* explained in Chapter 6.3. Therefore, it addresses Scenario A of our BC aware Core Components modeling approach defined by Formula 6.1. Likewise, the *Scenario B Tool* (Figure 7.3, Mark 27) implements the algorithm *AlgB* explained in Chapter 6.5. Therefore, it addresses Scenario B of our BC aware Core Components modeling approach defined by Formula 6.2.

The processing steps of *AlgA* and *AlgB* including the most important pieces of their pseudo-codes are already explained in Chapter 6. In a nutshell, these algorithms conduct the calculations which differ depending on the particular relationships which hold between the overall BCs of the specified BIEs. For instance, it is often necessary to detect from the list of the generic BIEs only those which are valid in the BC_{req} (effective BIEs). The corresponding pseudo-code is shown in Algorithm 7.4.

The ABIEs contained in the list of the generic ABIEs are processed within the loop initiated in Algorithm 7.4, Line 1. As explained in Chapters 4.3 and 5.2, the overall BC of an ABIE is calculated based on the union of the overall BCs of its included BIEs (BBIEs and ASBIEs).

```
Input: genericABIEList, p1, p2
Output: effectiveABIEList
 1: for each abie : genericABIEList do
 2:     newBBIEList = null;
 3:     newASBIEList = null;
 4:     fReused = false;
 5:     if directBCMatch(requestedBC, abie.overallBC) or
        indirectBCMatch(requestedBC, abie.overallBC) or
        partialBCMatch(requestedBC, abie.overallBC, p1, p2) then
 6:         for each bbie : abie.BBIEList do
 7:             if directBCMatch(requestedBC, bbie.overallBC) or
                indirectBCMatch(requestedBC, bbie.overallBC) or
                partialBCMatch(requestedBC, bbie.overallBC, p1, p2) then
 8:                 newBBIEList.add(bbie);
 9:                 fReused = true;
10:             end if
11:         end for
12:         for each asbie : asbie.ASBIEList do
13:             if directBCMatch(requestedBC, asbie.overallBC) or
                indirectBCMatch(requestedBC, asbie.overallBC) or
                partialBCMatch(requestedBC, asbie.overallBC, p1, p2) then
14:                 newASBIEList.add(asbie);
15:                 fReused = true;
16:             end if
17:         end for
18:         if fReused then
19:             newABIE = abie;
20:             newABIE.set(newABIEList);
21:             newABIE.set(newASBIEList);
22:             effectiveABIEList.add(newABIE);
23:         end if
24:     end if
25: end for
26: return effectiveABIEList
```

Algorithm 7.4: Effective BIEs Extractor

Therefore, if the currently processing ABIE is valid in the BC_{req} (checked in Algorithm 7.4, Line 5), it is possible that it contains the BIEs which are valid in the BC_{req}. These BBIEs and ASBIEs are selected in Algorithm 7.4, Lines 9 and 15, respectively. The non-selected BIEs are not relevant in the current business scenario, and, thus, excluded from the further processing.

The new ABIE which contains only the previously selected BIEs is generated in Algorithm 7.4, Lines 19-22. Thus, this newly created ABIE originates from the same Aggregate Core Component (ACC) as the currently processing ABIE, but it is *derived by restriction* based on the different business context (BC_{req}). Finally, the whole list of the effective ABIEs can be processed depending on the current processing step of *AlgA* or *AlgB*. For instance, it can be plugged into the customized business document implementation guideline (CustBDocIG), as explained in the processing step of *AlgA* described in Chapter 6.3.2.

7.3 Final Assessments

This Chapter has presented the implementation of our approach to calculate the content model of business context aware business document implementation guidelines. An implementation guideline represents a context specific constraint of the underlying generic document standard. We have especially considered the guidelines which are built upon CCTS (discussed earlier in Chapter 2). They are contextualized using our concurrent Enhanced UN/CEFACT Model (described in Chapter 4) and Business Context Ontology Model (described in Chapter 5).

The main contributions of the work presented in this Chapter are: (i) the XML based representation of contextualized Core Components, (ii) the prototype of our Core Components modeling approach described in Chapter 6, and (iii) the demonstration that our Core Components modeling approach does not hold only in theory, but also in practice.

First, we have adapted the standard UN/CEFACT's XML Naming and Design Rules specification (explained in Chapter 2.5) to represent contextualized business document implementation guidelines using XML schema elements. Second, we have developed the processing units which implement our Core Components modeling solutions. The final outcomes of the underlying algorithms are new, more homogeneous business document implementation guidelines which are valid in the business context specified by the user. The corresponding evaluation is conducted following the *design science research methodology* [5]. The related outcomes are summarized and compared in the following Chapter 8.

CHAPTER 8

Methodology and Evaluation

This Doctoral Thesis has developed a new approach to calculate the Core Component based data building blocks of electronic business documents. This Chapter describes the scientific methodology followed during the corresponding research. It especially highlights the evaluation and contributions of this thesis.

The research on the Core Components modeling was conducted in respect to the design science research (DSR) methodology [5]. Therefore, it was divided into heuristic and iterative research phases. Each of these phases follows the design science research guidelines (design as an artifact, problem relevance, design evaluation, research contributions, research rigor, design as a search process, and communication of research). The fulfillments of these guidelines are discussed in the rest of this Chapter.

Section 8.1 introduces the design science research methodology. It presents our global research framework and elaborates on the main research phases. The main artifacts of the research are classified and explained in Section 8.2. Afterwards, the relevancy of the research is underlined in Section 8.3.

Section 8.4 describes the evaluation of the presented work and discusses the evaluation results. First, it introduces the main evaluation criteria. Second, it uses these criteria to evaluate the business context models against each other. Based on the corresponding results, it proposes the business context model choice. Afterwards, this Section evaluates the usability and functionality of our business context aware Core Components modeling approach. Finally, it highlights the role of our communication-on-research in the evaluation process. The communication-on-research is presented in Section 8.5.

Section 8.6 specifies and classifies the contributions of this Doctoral Thesis. The following Section 8.7 presents our work as an iterative and heuristic design science research process. The scientific theories and concepts applied within this research process are underlined and briefly explained in Section 8.8. Finally, the concluding remarks of this phase of our research are summarized in Section 8.9.

8.1 Research Methodology

The research described in this Doctoral Thesis has been conducted following the design science research (DSR) methodology [5]. It was organized in the form of coordinated research phases: (i) state of the art in business context, (ii) development of the Enhanced UN/CEFACT Business Context Model, (iii) development of the Business Context Ontology Business Context Model, (iv) contextualization of Core Components using the Enhanced UN/CEFACT Business Context Model, (v) contextualization of Core Components using the Business Context Ontology Business Context Model, (vi) development of the business context aware Core Components modeling, and (vii) implementation of the business context aware Core Components modeling. The outcomes of each research phase are processed as starting design foundations of the succeeding research phases. This is illustrated in our research framework shown in Figure 8.1.

Research Phase 1 - State of the Art in Business Context (BC). This research phase is illustrated in Figure 8.1, Mark 1. First, it summarizes and compares different definitions of context and context awareness applied in context theory. Second, it defines business context and business context awareness in the scope of electronic business documents which are exchanged between business partners when executing inter-organization business processes. Third, it summarizes and compares different context modeling techniques. Finally, it presents starting guidelines for modeling the previously defined business context. The corresponding research is explained in Chapter 3 and discussed in [15, 16, 102–104, 109–111].

Research Phase 2 - Development of the Enhanced UN/CEFACT Business Context Model (E-UCM). This research phase is illustrated in Figure 8.1, Mark 2. First, it analyzes the structure and usability of the UN/CEFACT Business Context Model (UCM) [4]. Afterwards, it defines the Enhanced UN/CEFACT Business Context Model to represent business context defined in the previous research phase (research phase 1, Figure 8.1, Mark 1). The corresponding research is explained in Chapter 4 and discussed in [102–104].

Research Phase 3 - Development of the Business Context Ontology Business Context Model (BCOnt). This research phase is illustrated in Figure 8.1, Mark 3. First, it analyzes the general ontology based context modeling techniques applied in context theory. Afterwards, it defines the Business Context Ontology to represent business context defined in the previous research phase (research phase 1, Figure 8.1, Mark 1). The corresponding research is explained in Chapter 5 and discussed in [15, 109–111].

Research Phase 4 - Contextualization of Core Components Using the Enhanced UN/CEFACT Business Context Model. This research phase is illustrated in Figure 8.1, Mark 4. It develops the approach to contextualize the electronic business documents which are exchanged between business partners when executing inter-organizational business process. The exchanged documents conform to the Core Components Technical Specification (CCTS) [3]. The business context is represented using the Enhanced UN/CEFACT Model developed in the preceding research phase (research phase 2, Figure 8.1, Mark 2). The corresponding research is explained in Chapter 4 and discussed in [102–104].

Research Phase 5 - Contextualization of Core Components Using the Business Context Ontology Business Context Model. This research phase is illustrated in Figure 8.1, Mark 5. Analogously to its concurrent research phase 4 (Figure 8.1, Mark 4), it develops the approach

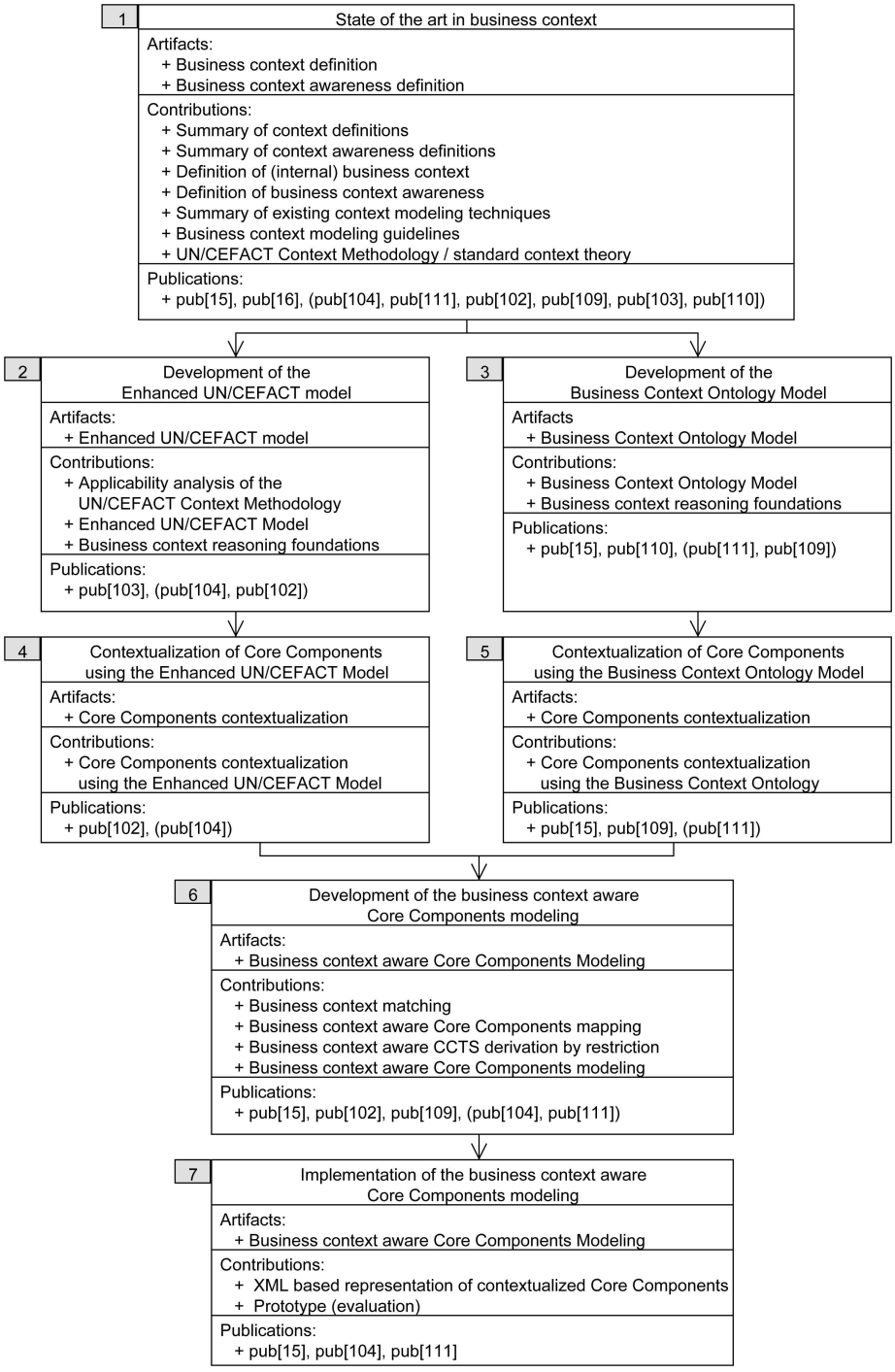


Figure 8.1: Research framework

to contextualize the Core Components. In contrary to the research phase 4, the business context is represented using the Business Context Ontology Model developed in the preceding research phase (research phase 3, Figure 8.1, Mark 3). The corresponding research is explained in Chapter 5 and discussed in [15, 109, 111].

Research Phase 6 - Development of the Business Context Aware Core Components Modeling. This research phase is illustrated in Figure 8.1, Mark 6. It designs the approach to (semi-) automatically develop new business context aware Core Component subsets, so called Business Document Implementation Guidelines (BDocIGs), of the CCTS document standard. The business context is represented using our concurrent Enhanced UN/CEFACT Model and Business Context Ontology Model which are designed in the previous research phases (research phases 2 and 3, respectively). The corresponding research is explained in Chapter 6 and discussed in [15, 102, 104, 109, 111].

Research Phase 7 - Implementation of the Business Context Aware Core Components Modeling. This research phase is illustrated in Figure 8.1, Mark 7. First, it develops the prototype of the business context aware Core Components modeling approach. This prototype is based on the algorithms which are developed in the previous research phases. Second, this research phase evaluates the results achieved by the concurrent application of the Enhanced UN/CEFACT and Business Context Ontology Business Context Models. The corresponding research is explained in Chapter 6 and in the following Sections of this Chapter. The relevant publications are [15, 104, 111].

8.2 Design Science Research Guideline - Design as an Artifact

The research presented in this Doctoral Thesis develops four main artifacts:

- Definition of the business context and business context awareness,
- Business context representation models (Enhanced UN/CEFACT Model and Business Context Ontology Model),
- Approach to contextualize Core Components by means of the developed business context models, and
- Approach to (semi-) automatically model electronic business document implementation guidelines.

Business Context Definition. We have defined business context in Chapter 3 (research phase 1). Accordingly, a business context represents the set of attributes which describe the situation of some electronic business document exchanged between business partners when executing inter-organizational business processes. The capability to sense, share and (partially) re-use this business contextual information represents a business context awareness.

Business Context Models. The Enhanced UN/CEFACT Model (E-UCM) and the Business Context Ontology Model (BCOnt) represent our concurrent business context models. They are used to express business context defined in Chapter 3, Definition 3.2.1.

The Enhanced UN/CEFACT Model is a Logic based Business Context Model which has been developed upon the already existing UN/CEFACT Model (UCM) [4]. The corresponding research is elaborated in Chapter 4 (research phase 2). In contrary, the Business Context Ontology Model is an ontology based business context modeling approach. It has been developed within the research elaborated in Chapter 5 (research phase 3).

Core Components Contextualization. Both the Enhanced UN/CEFACT and the Business Context Ontology Models can be used to contextualize the semantically interoperable Core Components. Generally speaking, every Core Component is valid in its assigned business context. However, this assigned business context can be subject of further contextualization depending on the business contexts in which other relevant Core Components are or are not valid. The corresponding calculation Formulas are described in Chapter 4.3 (when our E-UCM model is used) and in Chapter 5.2 (when our BCOnt model is used). The underlying algorithms are discussed in Chapter 7 (Algorithm 7.2 and Algorithm 7.3).

Business Context Aware Core Components Modeling. Our approach to (semi-) automatically model the subsets of the CCTS business document standard is described in Chapter 6 (research phase 6). We distinguish between two main application scenarios.

The first application scenario consists of the set of the already existing contextualized electronic documents which are valid in different business contexts. However, the electronic business document - which should be valid in the business context specified by a user - is missing. The corresponding problem statement is expressed by Formula 6.1. It is addressed by the algorithm *AlgA* described in Chapter 6.3.

The second application scenario consists of at least two business domains specified by different business contexts. In the first domain there are at least two related electronic documents (e.g., Purchase Order and Invoice documents). In the second domain one of the related documents is missing. The corresponding problem statement is expressed by Formula 6.2. It is addressed by the algorithm *AlgB* described in Chapter 6.3.

The final outcomes of *AlgA* and *AlgB* are the homogeneous subsets of the CCTS business document standard which are valid in the business contexts explicitly requested by a user. The business context is presented using either our Enhanced UN/CEFACT Model or our Business Context Ontology Model.

8.3 Design Science Research Guideline - Problem Relevance

The importance of context in the domain of business documents modeling has for the first time been detected within the work on the Electronic Business using eXtensible Markup Language (ebXML) [9, 120]. This is a family of XML based standards proposed by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) [7] and the Organization for the Advancement of Structured Information Standards (OASIS) [121]. It has been developed to provide an open, XML based infrastructure that enables the global use of electronic business information in an interoperable, secure, and consistent manner by all trading partners [9].

As described in [120], ebXML is structurally based on the subsequent layers. We briefly introduce them in the top-bottom order:

- *Core Components* are conceptual data model components for assembling electronic business documents (see Chapter 4);
- *Business Process* defines business documents/Core Components supporting a single step in the choreography of a business process;
- *Trading Partner Profile* specifies the Trading Partner Profile (TPP) and the Trading Partner Agreement (TPA). The TPP describes the capabilities of an individual business partner (commercial/business and technical). The TPP describes the agreed intersections between the profiles of two business partners;
- *Registry and Repository* provide a set of services for sharing of information between partners participating in an ebXML transaction (e.g., collaboration protocol profiles of trading partners, core libraries, business libraries, business process and business documents);
- *Transport and Routing* is a bottom ebXML layer which defines a wire format and protocol for exchanging electronic business documents at run-time.

The Core Component architecture has been developed by UN/CEFACT. In 2001, its key aspects were produced: (i) the information component model (Core Components/Business Information Entities) and (ii) the basic Context Methodology. However, the Context Methodology was defined only on an abstract level. On the one hand, it has been viewed by some stakeholders as *«inadequate and/or underspecified »* [3, 4]. On the other hand, it has been viewed by many as *«the cornerstone to reduce ambiguity and fully enable the CCTS component model»* [3, 4]. Therefore, the UN/CEFACT Unified Context Methodology (UCM) Project [4] was started in 2005. Its main goal is to develop a robust and flexible Context Methodology.

This Doctoral Thesis builds upon the UCM Project. First, it correlates UCM with the general context theory. Afterwards, it analyzes the applicability of the solutions proposed by UCM and develops the new solutions which overcome the detected shortcomings. Finally, it presents the business context aware Core Components modeling approach enabling the global use of electronic business information in an interoperable, secure, and consistent manner.

8.4 Design Science Research Guideline - Evaluation

The evaluation of our research has been conducted following the evaluation tenets defined by the DSR methodology. Therefore, it has been interpreted as the build and evaluate loop iterated a number of times before the final design artifacts were developed. The evaluation consists of three main interwoven phases:

- Evaluation of the business context models (UCM model, E-UCM model and BCOnt model),
- Evaluation of the business context aware Core Components modeling, and
- Expertise from the relevant scientific community.

8.4.1 Evaluation of the Business Context Models

The usability and functionality issues of the UCM model have already been proved by UN/CEFACT in [3,4,14]. As described in previous Chapters, these criteria are additionally streamlined by the E-UCM and BCOnt models. In our research we evaluate the UCM model, the E-UCM model and the BCOnt model against each other.

First, we analyze the evaluation criteria used in different scientific researches [4, 37–40, 73, 97, 122–125] to evaluate the context models used in different domains. Second, we select those evaluation criteria which we consider to be relevant for comparisons of our business context models. These criteria are briefly described in the following.

- **Type of Formalism [37–39, 123]** addresses data structures used to capture and represent business context. As described in Chapter 3, different data structures provide different features (reasoning support, contextual knowledge classification, contextual knowledge sharing, etc.);

- **Degree of Formalism [37,38,123,124]** addresses the existence of a formal definition and whether the formalization expresses the intuition. The formalization enables the computation of the business contextual knowledge;

- **Expressiveness [4,122]** denotes the possibility to unambiguously represent business context of arbitrary complexity. A model should allow to describe as much context as possible in arbitrary detail;

- **Flexibility [38,122,125] (Extensibility [40,122,125])** denotes the ability of the model to easily adapt to different business contexts. A model should support the simple addition of new business context elements and relations;

- **Generality [124,125] (Richness and Detail [124])** specifies the quality of having widespread applicability. For instance, a model can be « application-domain bounded » [38] if it substantially focuses on a specific business scenario, or « fully general » [38] if it can deal with different business scenarios;

- **Variable Business Context Granularity [38, 125]** specifies the ability of the model to represent the characteristics of the business context at different levels of detail;

- **Valid Business Context Constraints [38]** addresses the possibility to reduce the number of admissible context by imposing semantic constraints that the contexts must satisfy for a given target business scenario;

- **Business Context Construction [37,38,124]** highlights if the context description is built centrally or via a distributed effort. This indicates whether a central, typically design-time, description of the possible contexts is provided, or if a set of partners reaches an agreement about the description of the current context at run-time;

- **Interoperability [97,124,125]** highlights the ability to exchange and re-use business contextual knowledge;

- **Business Context Reasoning [38,124,125]** indicates whether the business context model enables reasoning on business contextual data to infer more specific or more abstract business contextual information;

- **Inconsistency Detection [73, 126]** indicates the capability of reasoning to ensure that a model does not contain any contradictory contextual data;

- **Knowledge Classification [73]** indicates the capability of reasoning to classify business contextual data;

- **Cost of Implementation** denotes the design time and effort needed to develop and maintain a business context model and its instances;

- **Existing Developing Tools [73]** refers to the existence of user-friendly graphical program solutions which can be used to develop a business context model and its instances;

- **Applicability to Real Environments [37, 124]** represents the ability to use a business context model in real environments and make it interoperable to existing systems;

- **Knowledge Base Extensibility [73]** indicates the capability to automatically correlate a business context model which does not define some specific context to external context models where the missing business context is defined.

The overview of the comparisons between the UCM model, the E-UCM model and the BCOnt model is presented in Figure 8.2. We highlight the most important outcomes in the following.

UN/CEFACT Context Model. The UN/CEFACT Context Model (UCM) [4] is a graph based business context model. It has been developed for the representation of the business context which is applied under the umbrella of the UN/CEFACT standardization initiative. As explained in Chapter 4, business context is organized in the form of a directed acyclic graph (DAG). A business context edge in the DAG represents the parent-business-context-node / children-business-context-node restriction property. In contrary to the BCOnt model, this is the only allowed property type defined by the UCM model.

The UCM model is a centralized context model which must be able to address all possible variations of business context. Therefore, in contrary to our concurrent E-UCM and BCOnt models, it addresses the highest degree of generality. Consequently, the underlying DAG structure is bewildering complex and usually comprises thousands of business context nodes and millions of possible business context edges (Chapter 4, Formula 4.1).

According to graph theory, memory and time complexities of the graph management operations strictly depend on the total number of nodes and edges [106]. Thereby, the representation of business context in the form of the centralized UCM business context graph undermines the usability of the model. For instance, the ensured generality of the approach increases the model implementation costs and violates the flexibility. At the same time, the user friendly tools - which could support the design of the UCM model and its instances - currently do not exist.

	Evaluation Criterion	UCM Model	E-UCM Model	BCOnt Model
1.	Type of Formalism	Graph based	Graph based Logic based	Ontology based
2.	Degree of Formalism	Low (-)	High (+)	High (+)
3.	Expressiveness	-	+	++
4.	Flexibility	-	+ +	+
5.	Generality	++	-	+
6.	Variable Granularity	+	+	+
7.	Valid Constraints	-	+	+
8.	Business Context Construction	Centralized	Decentralized	Decentralized
9.	Interoperability	-	+	+ +
10.	Business Context Reasoning	-	+	+ +
11.	Inconsistency Detection	-	-	+ +
12.	Knowledge Classification	-	-	+ +
13.	Cost of Implementation	High (- -)	Low (+ +)	High (-)
14.	Existing Development Tools	-	-	+
15.	Applicability to Real Environments	Limited (-)	+	+ +
16.	Knowledge Base Extensibility	-	-	+ +

Figure 8.2: Business context models appropriateness

Furthermore, as explained in Chapter 4, the UCM model expression grammar is incomplete and ambiguously defined. Therefore, the degree of formalism is low, the expressiveness is limited and the business context reasoning is not feasible. Likewise, the related evaluation criteria, such as the valid business context constraints, inconsistency detection, knowledge classification and interoperability, can not be fulfilled. This, in turn, limits the usability of the model to real business environments.

The complete explanation of the UCM model is presented in Chapter 4. In a nutshell, as highlighted in Figure 8.2, the generality is the most important benefit of the model application. However, (i) the organization of business context in the form of the centralized graph and (ii) the low degree of formalism are the main drawbacks.

Enhanced UN/CEFACT Context Model. The Enhanced UN/CEFACT Model (E-UCM) is built upon the standard UCM model. It is important that the standard model completely conforms to our enhanced version. In other words, the already existing solutions developed in correspondence to the original UCM model are valid and directly applicable in regards to our E-UCM business context model.

As explained in Chapter 4, the E-UCM model represents a transition from a centralized to a decentralized business context graph approach. Consequently, a total number of nodes and

edges in the business context graph is significantly reduced, which, in turn decreases the cost of model implementation. Furthermore, this new approach allows re-using existing subgraphs and building them into the decentralized business context graph. Hence, in contrary to the original model, the E-UCM model addresses the norms, such as the business contextual knowledge re-use, interoperability and flexibility. However, the generality (typical for the standard UCM model) is reduced.

Furthermore, we have developed the E-UCM model following the design criteria typical for the Logic based Models (introduced in Chapter 3.3.5). In particular, the E-UCM model expression grammar is unambiguously defined and a high degree of formalism is guaranteed. Consequently, as highlighted in Figure 8.2, the expressiveness of the UCM model is enriched and the foundations for the business contextual reasoning are established. However, in contrary to the BCOnt model, the E-UCM model does not provide an automatic inconsistency detection and a business contextual knowledge classification.

The complete explanation of the E-UCM model is presented in Chapter 4. In a nutshell, (i) the organization of business context in the form of a decentralized graph, (ii) the flexibility of the knowledge re-use and (iii) the high degree of formalism are the main benefits of the usability of the model. However, as shown in the example presented in Chapter 6.3.4, the usability of the model is strictly limited by its business contextual knowledge base. Therefore, Figure 8.2 highlights that the E-UCM model does not address the knowledge base extensibility evaluation criterion.

Business Context Ontology Model. The Business Context Ontology Model (BCOnt) represents our OWL DL [92–94] based business context model. Similarly to the E-UCM model, this is a decentralized model which provides the high degree of formalism. However, besides the parent-children (superclass-subclass) contextual restriction property (also defined by the previously summarized UCM and E-UCM context models), the additional restrictions (such as the functional property, inverse property and property cardinality) enrich more the expressiveness of the BCOnt model. Furthermore, as highlighted in Figure 8.2, this model provides the capabilities of automatic business contextual knowledge classification and inconsistency detection.

The business contextual knowledge extensibility represents the most important benefit of the BCOnt model usability. As explained in Chapter 5, the BCOnt model is built upon a three level subontology structure. It is, thus, possible to automatically extend this model with external pluggable concepts - defined in different ontologies from the scope of Linked Open Data [73] - depending on the current business scenario. This is demonstrated in the example presented in Chapter 6.3.4. Consequently, the generality of the BCOnt model is improved. Furthermore, this model follows the widely used Semantic Web Standards [127], which, in turn significantly underpins its interoperability and usability to real business environments.

The design of the BCOnt model is a time-consuming and demanding task. Therefore, as highlighted in Figure 8.2, the cost of the model implementation is high. However, the existence of the development tools (e.g., the Protégé modeling tool [83]), Semantic Web frameworks (e.g., the Jena Semantic Web Framework [117]), business context reasoners (e.g., the Pellet reasoner [71]), and query languages (e.g., SPARQL RDF query language [118, 119]) significantly eases the ontology design. The application of these tools is demonstrated in Chapter 7.

The complete explanation of the BCOnt model is presented in Chapter 5. In a nutshell,

(i) the distributed organization of business context in the form of a three level DL subontology structure, (ii) the high degree formalism enabling the capabilities of reasoning, automatic classification and inconsistency detection, (iii) the knowledge base extensibility and (iv) the Semantic Web openness are the main benefits of the usability of the model. However, the generality and extensibility of the model increase its design requirements.

8.4.2 Business Context Model Choice

Our research described in Chapter 3 has concluded that the uniform context modeling technique does not exist. Accordingly, the choice of the most proper context modeling approach tightly depends on a domain specific nature of the context. Furthermore, our analysis of different context modeling techniques has directed our research on the business context modeling to the logic based and ontology based context modeling directions.

The already existing UCM model, our logic based E-UCM model and our ontology based BCOnt model are evaluated in the previous Subsection on the basis of the 16 criteria. Based on the fulfillment of these criteria (summarized in Figure 8.2), we have stipulated that the standard UN/CEFACT Model is incomplete, and that its usability is streamlined by the Enhanced UN/CEFACT and Business Context Ontology models. Furthermore, we can conclude that the choice between the E-UCM and BCOnt modeling approaches tightly depends on the particular business scenario and its requirements.

On the one hand, the application of the E-UCM model is encouraged when the flexibility of the business contextual knowledge base is more important than its generality. Likewise, the completeness and consistency of the knowledge base must be guaranteed beforehand. Furthermore, the complexity of the business contextual knowledge must conform to the only allowed parent-children relationships. Finally, the re-usability of instances of the E-UCM model additionally eases its application to new business scenarios.

On the other hand, the application of the BCOnt model is encouraged when the generality of the business contextual knowledge is more important than its flexibility. When needed, the local knowledge base can always be automatically correlated with external, on-line published knowledge bases. In addition to the fundamental reasoning capabilities, the BCOnt model has the capabilities of automatic knowledge classification and inconsistency check. Therefore, the beforehand guaranteed completeness and inconsistency of the knowledge base are not preconditions of the application of the model.

Furthermore, the expressiveness of the BCOnt model is not restricted by the only allowed parent-children relationships. Consequently, this model can represent more complex business contextual knowledge than the E-UCM model. Finally, the BCOnt model conforms to the widely used standards (such as the Semantic Web standards [127]). Consequently, the already existing software solutions based on these standards can easily access and process the knowledge which is stored by instances of this model.

8.4.3 Evaluation of the Business Context Aware Core Components Modeling

Chapter 6 described our approach to model new business context aware business document implementation guidelines (BDocIGs) applying our business context models (evaluated in the pre-

vious Subsection). Chapter 7 implemented the corresponding conceptual solutions and the underlying algorithms. According to the DSR methodology, the evaluation of this phase of our work was conducted as build and evaluate loop iterated a number of times before the final algorithms were developed.

- We created the instances of the business context (BC) models based on the geopolitical classification proposed by the Food and Agriculture Organization of the United Nations (FAO) [112] and on the industry classification proposed by the International Standard Industrial Classification of All Economic Activities (ISIC) [105].
  - The excerpt of the centralized UCM BC graph which consisted of 900 BC nodes and 2000 BC edges was built.
  - The decentralized E-UCM BC subgraphs were built and combined into different decentralized E-UCM BC graphs. The number of BC nodes in the decentralized E-UCM BC graphs varied from 5 to 100.
  - The exemplary BCOnt which consisted of 35 classes, 1230 individuals and 7325 triples was built. Afterwards, its geopolitical BCFAO subontology was interconnected with the external DBpedia ontology [73, 80].
- We implemented the prototype of our business context aware Core Components modeling approach based on the architecture described in Chapter 7.
- We had the set of 20 available excerpts of the already existing contextualized business document implementation guidelines. These guidelines were based on the CCTS document standard and enhanced NDR specification as explained in Chapter 7.1.

During the evaluation we were especially considering the processing steps of the algorithm *AlgA* described in Chapter 6.3, the processing steps of the algorithm *AlgB* described in Chapter 6.5 as well as the processing steps of the algorithms emphasized in Chapter 7. Before every iterative step of the evaluation, one of the guideline excerpts (denoted by $BDocIG_{selected}$) was arbitrarily selected to be a missing guideline. Therefore, the business context - in which this guideline was valid - was processed by the system as the requested business context (BC_{req}) input parameter (shown in Chapter 7, Figure 7.3, Mark 6). The rest of the guidelines from the introduced set of guidelines was processed as the existing business document implementation guidelines (*ExistBDocIGs*) input parameter (shown in Chapter 7, Figure 7.3, Mark 5).

The resulting set of the Core Components contained by the generated business document implementation guideline (*CustBDocIG*) was analyzed and compared with the set of the Core Components contained by the original $BDocIG_{selected}$. Finally, in the later iterations of the evaluation we could conclude that the corresponding data building blocks located in both guidelines were matched with 95+%. Thus, usability and functionality of the business context aware Core Components modeling approach have been proved. A matching degree between the *CustBDocIG* and the $BDocIG_{selected}$ was measured using precision rate and recall rate [128–131].

Precision Rate and Recall Rate. Precision rate (*pr*) and recall rate (*rr*) are two measures which are widely applied in evaluating the performance of information retrieval systems. In our

testing, precision rate denotes a capability to generate only BIEs which are relevant in the BC_{req}. This can be specified by the following Formula:

$$pr = \frac{number\ of\ true - relevant,\ generated\ BIEs}{total\ number\ of\ generated\ BIEs} \tag{8.1}$$

A total number of generated BIEs represents a sum of true-relevant BIEs and false-relevant BIEs. True-relevant BIEs are data building blocks which are generated applying our Core Components modeling approach, and which should be contained by the generated BDocIG. In contrary, false-relevant BIEs are data building blocks which are generated applying our Core Components modeling approach, but which should not be contained by the generated BDocIG.

Recall rate denotes a capability to generate all of the BIEs which are relevant in the BC_{req}. This can be specified by the following Formula:

$$rr = \frac{number\ of\ true - relevant,\ genereated\ BIEs}{total\ number\ of\ relevant\ BIEs} \tag{8.2}$$

Testing Results. Our evaluation tests were processing a different number of input *ExistBDocIGs*. Furthermore, we especially considered an influence of the knowledge base extensibility and reasoning capabilities of the BCOnt model on changes in precision rates and recall rates. The comparable results are presented in Figure 8.3 and are graphically illustrated in Figure 8.4.

The testing results show that recall rate strongly depends on the size of the contextual knowledge base. In case that a business contextual knowledge base is small, the precision rate is high, but the recall rate is low. For instance, for the set of 3 input *ExistBDocIGs*, our tests calculated the precision rate *pr* = 0.80, and the recall rate *rr* = 0.20 (the knowledge base extensibility and the BC reasoning were not applied). This means that most of the generated BIEs are really relevant in the BC_{req} (true-relevant BIEs), but many of relevant BIEs are not generated. Furthermore, the data in Figure 8.3 show that with an increase in a size of the contextual knowledge base, the recall rate also grows, and the precession rate stays high. For instance, for the set of 19 input *ExistBDocIGs*, our tests calculated the precision rate *pr* = 0.89, and the recall rate *rr* = 0.80 (the knowledge base extensibility and the BC reasoning were not applied).

Likewise, the testing results indicate that the knowledge base extensibility reduces the precision rate (e.g., *pr* = 0.89 → 0.78, number of *ExistBDocIGs* = 19). In contrary, the application of the BC reasoning increases the precision rate (e.g., *pr* = 0.78 → 0.95, number of *ExistBDocIGs* = 19). The interconnection of the BCOnt ontology with external ontologies results in generating not only true-relevant BIEs, but also in generating a significant number of false-relevant BIEs. Clearly, the total number of generated BIEs is increased, and, consequently, the precision rate is reduced. However, many of these false-relevant BIEs can be eliminated applying the BC reasoning rules. Furthermore, these rules can be used to generate additional true-relevant BIEs. Clearly, the total number of generated BIEs might be reduced, the total number of true-positive BIEs is increased, and, consequently, the precision rate is increased.

Finally, the testing results point out that both the knowledge base extensibility and the BC reasoning increase the recall rate (e.g., *rr* = 0.80 → 0.90 → 0.95, number of *ExistBDocIGs* = 19). This is explained by the fact that there exist true-relevant BIEs which can be generated only when these capabilities are applied.

input BDocIGs	true-relevant BIEs	false-relevant BIEs	relevant BIEs	exter. KB	reasoning	pr	rr
3	4	1	20	-	-	0.80	0.20
3	5	2	20	+	-	0.71	0.25
3	7	1	20	+	+	0.88	0.35
14	14	2	20	-	-	0.88	0.70
14	15	4	20	+	-	0.79	0.75
14	17	1	20	+	+	0.94	0.85
19	16	2	20	-	-	0.89	0.80
19	18	5	20	+	-	0.78	0.90
19	19	1	20	+	+	0.95	0.95

Figure 8.3: Evaluation results - precision rate and recall rate

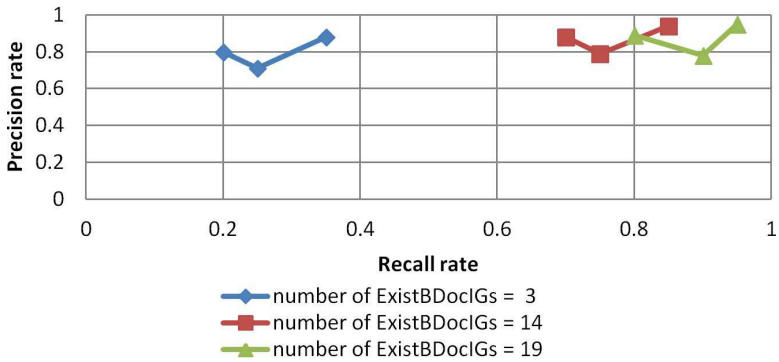


Figure 8.4: Changes in precision rate and recall rate

Therefore, based on the discussed result, we can conclude that our business context aware Core Components modeling approach holds not only in theory, but also in practice. The accuracy of the generated guidelines depends tightly on the business contextual knowledge base, and can be improved by interconnection with external knowledge bases and by application of the BC reasoning rules.

8.4.4 Expertise from the Relevant Scientific Community

Each phase of our research (Figure 8.1) was analyzed and influenced by the relevant scientific community. The corresponding pieces of the research are described in the form of academic publications which were presented and discussed within the conferences relevant in our research domain. This third important phase of the evaluation of our work is described in details in the following Section of this Chapter.

8.5 Design Science Research Guideline - Communication on Research

The communication on the research described in this Doctoral Thesis has been conducting within three main communication channels:

- Academic publishing,
- Presentations at the relevant conferences, and
- Website presentation.

8.5.1 Academic Publications

Publication 1. Danijel Novakovic and Christian Huemer. *Applying Business Context to Calculate Subsets of Business Document Standards*. To appear in *Information Technology and Management* - [15]

This scientific paper will appear in a Special Issue (Advances in E-business Engineering) of the Journal of Information Technology and Management. It describes the most important conclusions explained in this Doctoral Thesis focusing on the business context ontology modeling approach. Speaking briefly, it discusses the research phases 7, 6, 5 and 3 (Figure 8.1, Marks 7, 6, 5 and 3, respectively) and explains the most important outcomes of the research phase 1 (Figure 8.1, Marks 1).

Publication 2. Danijel Novakovic and Christian Huemer. *Context aware business documents modeling*. In Patrick Brézillon, Patrick Blackburn, and Richard Dapoigny, editors, *Modeling and Using Context*, volume 8175 of *Lecture Notes in Computer Science*, CONTEXT 2013, pages 357–363. Springer Berlin Heidelberg, 2013. - [104]

This scientific paper is published in the Proceedings of the 8th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2013), October 28 - November 1, Annecy, France. It describes the research phase 7 which is illustrated in our research framework in Figure 8.1, Mark 7. Therefore, it focuses on the implementation of the business context aware Core Components modeling using the Enhanced UN/CEFACT Business Context Model. Additionally, it addresses the main outcomes of the research phases 6, 4, 2 and 1 which are explained earlier in Section 8.1.

Publication 3. Danijel Novakovic and Christian Huemer. *Putting services in context*. In *Proceedings of the 6th IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2013), December 16-18, Kauai, Hawaii, The United States*, pages 38–42, IEEE, 2013. - [111]

This academic publication describes the research phase 7 which is illustrated in our research framework in Figure 8.1, Mark 7. Therefore, it focuses on the implementation of the business context aware Core Components modeling using the Business Context Ontology Model. Additionally, it addresses the main outcomes of the research phases 6, 5, 3 and 1 which are explained

in Section 8.1.

Publication 4. Danijel Novakovic and Christian Huemer. *Business context sensitive business documents: Business context aware core components modeling using the E-UCM model.* In *Proceedings of the 11th IEEE International Conference on Industrial Informatics (INDIN 2013), July 29-31, Bochum, Germany*, pages 523–528, IEEE, 2013. - [102]

This academic publication describes the research phases 6 and 4 shown in our research framework in Figure 8.1. Therefore, it explains the contextualization of Core Components and business context aware Core Components modeling. The business context is represented using our Enhanced UN/CEFACT Model. Additionally, this publication addresses the main outcomes of the research phases 2 and 1 which are discussed in Section 8.1.

Publication 5. Danijel Novakovic and Christian Huemer. *Contextualizing business documents.* In *Proceedings of the 10th IEEE International Conference on e-Business Engineering (ICEBE 2013), September 11-13, Coventry, The United Kingdom*, pages 236–243, IEEE, 2013. - [109]

This academic publication describes the research phases 6 and 5 shown in our research framework in Figure 8.1. Therefore, it explains the contextualization of Core Components and business context aware Core Components modeling. The business context is represented using our Business Context Ontology Model. Additionally, this publication addresses the main outcomes of the research phases 3 and 1 which are discussed in Section 8.1.

Publication 6. Danijel Novakovic, Christian Huemer and Christian Pichler. *Context model for business context sensitive business documents.* In Patrick Brézillon, Patrick Blackburn, and Richard Dapoigny, editors, *Modeling and Using Context*, volume 8175 of *Lecture Notes in Computer Science*, CONTEXT 2013, pages 336–342. Springer Berlin Heidelberg, 2013. - [103]

This scientific paper is published in the Proceedings of the 8th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2013), October 28 - November 1, Annecy, France. It describes the research phase 2 illustrated in our research framework in Figure 8.1, Mark 2. Therefore, it analyzes the standard UN/CEFACT Business Context Model and presents our Enhanced UN/CEFACT Business Context Model. Additionally, this publication addresses the main outcomes of the research phase 1 which is discussed in Section 8.1.

Publication 7. Danijel Novakovic and Christian Huemer. *Business context sensitive business documents: an Ontology based business context model for Core Components*, In *Proceedings of the 10th International Conference for Informatics and Information Technology (CIIT 2013), April 18-21, Bitola, Macedonia*, 2013. - [110]

This academic publication describes the research phase 3 illustrated in our research framework in Figure 8.1, Mark 3. Therefore, it develops the first version of our Business Context Ontology Model. Additionally, this publication addresses the main outcomes of the research phase 1 which is discussed in Section 8.1.

Publication 8. Danijel Novakovic and Christian Huemer. *A survey on business context.* In Durga Prasad Mohapatra and Srikanta Patnaik, editors, *Intelligent Computing, Networking, and*

Informatics, volume 243 of *Advances in Intelligent Systems and Computing*, ICACNI 2013, pages 199–211. Springer India, 2014. - [16]

This scientific paper is published in the Proceedings of the International Conference on Advanced Computing, Networking, and Informatics, ICACNI 2013, Raipur, Chhattisgarh, India,12-14 June 2013. It describes the research phase 1 illustrated in our research framework in Figure 8.1, Mark 1. First, our publication summarizes the existing scientific definitions of context and context awareness. Second, it defines business context and business context awareness. Third, it explains different context modeling techniques. Finally, it proposes the guidelines for modeling our understanding of business context.

8.5.2 Website Presentation

The research described in this thesis is published on-line [132]. The Website is regularly updated making our research progress permanently available to both scientific and business community.

8.6 Design Science Research Guideline - Research Contribution

The research described in this thesis utilizes *business contextual knowledge* in the scope of *electronic business documents* which are exchanged between business partners when executing inter-organizational business processes. The main contributions address the gaps in:

- Standard context theory,
- UN/CEFACT Context Methodology, and
- Core Components Technical Specification.

The complete contribution framework is illustrated in Figure 8.1.

8.6.1 Contributions to the Standard Context Theory

The standard context theory [6, 17, 19–23] provides numerous definitions of context in general. However, it does not address the business context in which electronic business documents are valid. Therefore, our research contributes to the standard context theory by providing: (i) the summary of existing context definitions, (ii) the summary of existing context awareness definitions, (iii) the definition of (internal) business context, (iv) the definition of business context awareness, (v) the summary of existing context modeling techniques, and (vi) the modeling guidelines for developing specification of business context models.

These contributions are denoted by Mark 1 in our research framework illustrated in Figure 8.1. The corresponding research phase 1 is discussed in Chapter 3 and published in [15, 16, 102–104, 109–111].

8.6.2 Contributions to the UN/CEFACT Context Methodology

The UN/CEFACT Context Methodology [3, 4] has been developed to manage representations and applications of business context, especially under the scope of the UN/CEFACT standardization effort. On the one hand, the Context Methodology has been recognized as the cornerstone to reduce ambiguity and to fully enable the CCTS component model. On the other hand, it has been evaluated as inadequate and/or underspecified. The main contributions of our related work can be grouped into:

- Correlation of the UN/CEFACT Context Methodology with the standard context theory,
- Enhanced UN/CEFACT Business Context Model (E-UCM), and
- Business Context Ontology Model (BCOnt).

Contributions - Correlation with Existing Scientific Foundations. The UN/CEFACT Context Methodology has been developed without considering the previous relevant scientific definitions of context. However, the correctness, applicability and further development of the UN/CEFACT Context Methodology can be guaranteed only if it is underpinned by the scientifically accepted context understandings. Our research detects the common objectives of context defined and/or exploited in both UN/CEFACT Context Methodology and standard context theory (such as context entity, context value and primary context category). Therefore, it precisely locates the UN/CEFACT Methodology and defines its main concepts in the scope of the standard context theory.

This contribution is denoted by Mark 1 in our research framework illustrated in Figure 8.1. The corresponding research phase 1 is discussed in Chapter 3 and published in [15, 16, 102–104, 109–111].

Contributions - Enhanced UN/CEFACT Context Model. The standard UN/CEFACT Context Methodology is incomplete, ambiguously defined and not directly applicable to real-world scenarios. Therefore, this Doctoral Thesis brings the distributed approach in the standard Context Methodology, more formally defines the existing context grammar, additionally extends the existing context grammar, and develops the Enhanced UN/CEFACT Business Context Model. The corresponding contributions can be grouped as: (i) the applicability analysis of the UN/CEFACT Context Methodology, (ii) the development of the logic based Enhanced UN/CEFACT Business Context Model, and (iii) the establishment of the business context reasoning foundations.

The highlighted contributions are denoted by Mark 2 in our research framework which is illustrated in Figure 8.1. The corresponding research phase 2 is discussed in Chapter 4 and published in [102–104].

Contributions - Business Context Ontology. The relevant scientific community [37, 40, 95, 95–101] agrees that ontologies are the most promising approach for modeling context in ubiquitous environments. Therefore, the ontology based modeling has already been widely applied to model different scopes of context (such as the context restricted to the pervasive systems [95], smart environments [97], mobile devices [101], etc.).

Our ontology based business context modeling approach utilizes and further develops the already existing ontology based modeling foundations to particularly model the business context in which electronic business documents are valid. The corresponding work extends the standard UN/CEFACT Context Methodology providing the following main contributions: (i) the Business Context Ontology Model, and (ii) the establishment of the business context reasoning foundations.

The highlighted contributions are denoted by Mark 3 in our research framework which is illustrated in Figure 8.1. The corresponding research phase 3 is discussed in Chapter 5 and published in [15, 109–111].

8.6.3 Contributions to the Core Components Technical Specification

As described in Chapter 2, CCTS is a methodology which defines models and rules for representing the structures and contents of electronic business documents. Our work underpins the usage of business context in the scope of the CCTS standardization activities and provides an approach for the CCTS based business documents modeling. Therefore, the corresponding contributions can be grouped into the following main pillars:

- Contextualization of Core Components, and
- Business context aware Core Components modeling.

Contributions - Core Components Contextualization. Both the Enhanced UN/CEFACT Model and the Business Context Ontology Model can be used to contextualize Core Components. Therefore, our corresponding research provides the following contributions: (i) the contextualization of Core Components by means of the Enhanced UN/CEFACT Business Context Model, and (ii) the contextualization of Core Components by means of the Business Context Ontology Model. These contributions are denoted by Marks 4 and 5 in our research framework (Figure 8.1), respectively. The corresponding research phases 4 and 5 are discussed in Chapters 4 and 5, respectively. They are published in [15, 102–104, 109, 111].

Contributions - Business Context Aware Core Components Modeling. This thesis presents the algorithms which perform the *business context aware CCTS derivation by restriction*. Consequently, the Core Component based business document implementation guidelines can (semi-) automatically be generated and the semantic interoperability of Core Components is ensured.

Our corresponding work provides the following contributions: (i) the definition of business context matching, (ii) the definition of business context aware Core Components mapping, (iii) the algorithms which conduct the *business context aware CCTS derivation by restriction*, (iv) the XML based representation of contextualized Core Components, and (v) the prototypical implementation of the business context aware Core Components modeling approach.

The highlighted contributions are presented in our research framework illustrated in Figure 8.1. The corresponding research phases 8 and 9 are discussed in Chapters 6 and 7, respectively. They are published in [15, 102, 104, 109, 111].

8.7 Design Science Research Guideline - Design as a Search Process

The research was organized in the form of a well-defined research phases. These phases are highlighted in Figure 8.1 and already explained in Section 8.1.

Each research phase was interpreted as a build and evaluate loop iterated a number of times before the final algorithms were developed. The temporal outcomes were evaluated, discussed with the relevant scientific community (communication of research is described in Section 8.5) and used as the starting foundations for the following iterations. The output of every research phase served as a basis for processing the succeeding research phases. Therefore, the design process was conducted as a heuristic and iterative search process until the solutions - which addressed the corresponding problem statements - were built.

8.8 Design Science Research Guideline - Research Rigor

Each phase of our research was developed considering the well founded theories and concepts. They were applied under specific and well-defined conditions. We highlight some of them:

- Context theory (business context definition, business context model development, etc.);
- Graph theory (Enhanced UN/CEFACT Model development);
- Set theory (Enhanced UN/CEFACT grammar development);
- Distributed systems (business context modeling);
- UN/CEFACT Context Methodology (business context definition, business context model development, Core Components contextualization, etc.);
- CCTS business document standard (business context aware Core Components modeling, Core Components contextualization, Business context modeling, etc.);
- CCTS's XML Naming and Design Rules (prototypical implementation);
- Business process modeling (Context Aware Core Components Modeling);
- Logic based context modeling (Enhanced UN/CEFACT Model development);
- Ontology based context modeling (Business Context Ontology modeling);
- Semantic Web (Business Context Ontology modeling);
- Linked Open Data (Business Context Ontology modeling);
- Description Logic (business context reasoning);
- Design science research (evaluation, planning);

- Precision and recall (evaluation), etc.

These scientific approaches are interwoven and they are discussed throughout the whole thesis.

8.9 Final Assessments

This Chapter described the methodology applied during our research on the business context aware Core Components modeling. The corresponding work has been conducted following the design guidelines which are defined by the design science research methodology (DSR) [5]. The overview of the fulfilment of these guidelines is shown in Figure 8.5. Accordingly, the research was organized as a design science research process comprising iterative research phases. The corresponding research framework is illustrated in Figure 8.1.

As highlighted in Figure 8.5, our research develops the following main artifacts: (i) the definition of the business context and business context awareness, (ii) the business context representation models (Enhanced UN/CEFACT Model and Business Context Ontology Model), (iii) the approach to contextualize Core Components by means of the developed business context mod-

	DSR Guideline	Description
1.	Design as an Artifact	Definition of the business context; Definition of business context awareness; Business context representation models (E-UCM and BCOnt models); Contextualization of Core Components by means of the developed business context models; Business context aware Core Components modeling
2.	Problem Relevance	UN/CEFACT; CCTS; UN/CEFACT Context Methodology
3.	Evaluation	Evaluation of the business context models (UCM model, E-UCM model and BCOnt model); Evaluation of the business context aware Core Components modeling; Expertise from the relevant scientific community
4.	Communication of Research	Academic publishing; Relevant conferences; Website; Prototype
5.	Research Contribution	Summary of existing context definitions; Summary of existing context awareness definitions; Definition of (internal) business context; Definition of business context awareness, Summary of existing context modeling techniques; Business context modeling guidelines; Correlation of the UN/CEFACT Context Methodology with the standard context theory; Applicability analysis of the UN/CEFACT Context Methodology; E-UCM model; BCOnt model; Business context reasoning foundations; Contextualization of Core Component; Business context aware Core Components modeling; Definition of business context aware matching; Definition of business context aware Core Components mapping; Business context aware CCTS derivation by restriction; XML based representation of contextualized Core Components; Prototype
6.	Research Rigor	context theory; graph theory; set theory; distributed systems; UN/CEFACT Context Methodology; CCTS business document standard; CCTS's XML Naming and Design Rules; business process modeling; logic based context modeling; ontology based context modeling; Semantic Web; Linked Open Data; Description Logic; design science research; precision and recall
7.	Design as a Search Process	Iterative and heuristic research until final algorithms which satisfy our requirements were developed

Figure 8.5: Overview of the design science research guidelines

els, and (iv) the approach to (semi-) automatically model the subsets of the Core Components Technical Specification (CCTS) business document standard [3].

The evaluation of our research has been interpreted as the build and evaluate loop iterated a number of times before the final design artifacts were developed. The evaluation consisted of three main interwoven phases: (i) the evaluation of the business context models, (ii) the evaluation of the business context aware Core Components modeling approach and (iii) the expertise from the relevant scientific community.

The evaluation of the business context models has been performed against the 16 previously introduced evaluation criteria, such as the type of formalism, expressiveness and knowledge base extensibility. The summary of the comparisons between the UN/CEFACT Model, the Enhanced UN/CEFACT Model and the Business Context Ontology Model is presented in Figure 8.2. Accordingly, we have concluded that the drawbacks of the standard UN/CEFACT Model are overcome by the Enhanced UN/CEFACT and Business Context Ontology models. Furthermore, we have stipulated that the choice of the business context model tightly depends on the particular business scenario and its requirements.

For instance, in case that (i) the flexibility of the business contextual knowledge base is more important than its generality, (ii) the completeness and consistency of the business contextual knowledge base are guaranteed beforehand, and (iii) the higher degree of complexity of the business contextual knowledge is not required, the application of the Enhanced UN/CEFACT Model is encouraged. However, in case that (i) the generality of the business contextual knowledge is more important than its flexibility, (ii) the completeness and consistency of the business contextual knowledge base are not guaranteed beforehand, (iii) the high degree of complexity of the business contextual knowledge is required, and (iv) the seamless integration with the widely used standards (such as the Semantic Web standards [127]) is needed, the application of the Business Context Ontology Model is encouraged.

As part of the evaluation of the business context aware Core Components modeling, we implemented the prototypical architecture explained in Chapter 7. We built the instances of the business context models based on the geopolitical classification proposed by the Food and Agriculture Organization of the United Nations (FAO) [112] and on the industry classification proposed by the International Standard Industrial Classification of All Economic Activities (ISIC) [105]. We tested the executions of the algorithms explained in Chapters 6 and 7 on 20 available excerpts of electronic business document implementation guidelines. The generated guidelines were analysed adapting the precision rate and recall rate measures [128–131]. Based on the corresponding outcomes (shown in Figure 8.3) we can conclude that our Core Components modeling approach holds not only in theory, but also in practice.

Finally, this Chapter highlighted and classified the contributions of the thesis. They address the gaps in different domains, such as the standard context theory [6], UN/CEFACT Context Methodology [4] and the Core Components Technical Specification [3]. The most important contributions are: (i) the definition of business context (explained in Chapter 3), (ii) the definition of business context awareness (explained in Chapter 3), (iii) the Enhanced UN/CEFACT Model (explained in Chapter 4), (iv) the Business Context Ontology Model (explained in Chapter 5), (v) the Core Components contextualization (explained in Chapters 4 and 5) and (vi) the business context aware Core Components modeling (explained in Chapters 6 and 7).

CHAPTER 9

Conclusion and Open Research Issues

Inter-organizational business processes are more and more executed by a flow of well-defined electronic business documents. The concrete structures of these documents significantly differ depending on business context (geopolitical region, industry, activity, etc.) in which the underlying business transactions are executed. Therefore, the electronic business documents must be built upon the generic business document standards that can easily be adapted to any business context. These adaptations always involve the business context aware restrictions of the document standards resulting into their small subsets, so called business document implementation guidelines.

This Doctoral Thesis has developed an approach to (semi-) automatically generate the business context aware business document implementation guidelines. It especially considered the guidelines built upon the Core Components Technical Specification (CCTS) (explained in Chapter 2). Accordingly, every business document comprises the set of semantically interoperable data building blocks, defined as Core Components. The outcomes of the described work address the research gaps in different domains, such as the standard context theory [6], the UN/CEFACT Context Methodology [4] and the Core Components Technical Specification [3].

9.1 Final Assessments on the Business Context Aware Core Components Modeling

This Doctoral Thesis builds the following main artifacts: (i) the definition of the business context and business context awareness, (ii) the business context representation models (Enhanced UN/CEFACT Model and Business Context Ontology Model), (iii) the approach to contextualize Core Components by means of the developed business context models, and (iv) the approach to (semi-) automatically model business document implementation guidelines.

Business Context and Business Context Awareness. Business context is any information that can be used to characterize the situation of an entity within a scope where business operates. An entity is a person, place, or object that is considered relevant to the execution of a business

process in a business environment, including the business process and business environments themselves. The introduced entities can be described by different attributes, where each of these attributes can be grouped into one of the primary business context categories.

This Doctoral Thesis develops the approach which is independent on the number of business context categories. When explaining corresponding solutions, we considered three business context categories which were specified as most important for the characterization of business context: (i) geopolitical business context category, (ii) industry business context category and (iii) activity business context category. However, the proposed solutions can process the knowledge which originates from additional primary business context categories (e.g., official constraints, product classification and business process role business context categories) as well.

Business context awareness represents the capability to sense, process and (partially) re-use business contextual information. This Doctoral Thesis detects and utilizes business contextual information comprised by the business context aware electronic business documents.

Business Context Models. This Doctoral Thesis compared different techniques applied to model context in general (Key Value Models, Markup Scheme Models, Graphic based Models, Object Oriented Models, Logic based Models and Ontology based Models). The corresponding conclusions have stipulated that the uniform context modeling technique does not exist. In practice, the choice of the most proper context modeling approach tightly depends on a domain specific nature of the context. Consequently, the analysis of different context modeling techniques (summarized in Chapter 3, Figure 3.7) directed our research on the business context modeling to the logic based and ontology based context modeling directions.

This Doctoral Thesis presented three business context models: (i) the UN/CEFACT Model, (ii) the Enhanced UN/CEFACT Model (E-UCM) and (iii) the Business Context Ontology Model (BCOnt). The UN/CEFACT Model was proposed by the UN/CEFACT Context Methodology [4] for the representation of the business context in the scope of the UN/CEFACT's standardization activities. In contrary, the Enhanced UN/CEFACT and Business Context Ontology Models are concurrent modeling approaches which were directly developed during the research described in this Doctoral Thesis. On the one hand, the Enhanced UN/CEFACT Model represents are logic based extension to the standard UN/CEFACT Model. On the other hand, the Business Context Ontology Model follows an ontology based approach for context modeling.

The UN/CEFACT Model, the Enhanced UN/CEFACT Model and the Business Context Ontology Model have been evaluated on the basis of the 16 introduced evaluation criteria, such as the degree of formalism, generality and knowledge base extensibility. The corresponding summary is highlighted in Chapter 8, Figure 8.2. Accordingly, this thesis has concluded that the standard UN/CEFACT Model is partially defined, and that its usability is streamlined by the Enhanced UN/CEFACT and Business Context Ontology models.

Furthermore, we have stipulated that the choice of the business context model tightly depends on the particular business scenario and its concrete requirements. For instance, in case that (i) the flexibility of the business contextual knowledge base is more important than its generality, (ii) the completeness and consistency of the business contextual knowledge base are guaranteed beforehand, and (iii) the higher degree of complexity of the business contextual knowledge is not required, the application of the Enhanced UN/CEFACT Model is encouraged. However, in case that (i) the generality of the business contextual knowledge is more important than its flex-

ibility, (ii) the completeness and consistency of the business contextual knowledge base are not guaranteed beforehand, (iii) the higher degree of complexity of the business contextual knowledge is required, and (iv) the seamless integration with the widely exploited standards (such as the Semantic Web standards [127]) is needed, the application of the Business Context Ontology Model is encouraged.

Business Context Aware Core Components Modeling. The developed business context models can be applied to contextualize semantically interoperable Core Components. Therefore, the including electronic business documents become business context aware, and, thus, can be (partially) re-used to (semi-) automatically generate the Core Component structures of the missing documents.

The main use case scenario, considered in this Doctoral Thesis, consists of *n* ($n > 1$) business domains specified by different business contexts. In *(n-1)* domains there are *m* ($m > 1$) related electronic business documents per domain (e.g., Purchase Order and Invoice documents). The last domain comprises *m-1* business documents, while one document is missing. This Doctoral Thesis has developed the algorithm which performs different types of mappings between the existing Core Components and calculates the business context matching criteria. Based on the outcomes of these processing steps, the missing subset of the Core Component based business document standard is generated.

The feasibility of the presented approach was demonstrated by a prototype implementation illustrated in Chapter 7, Figure 7.3. The evaluation was conducted following the design science research methodology [5]. *It has shown that the designed business context aware Core Components modeling approach automates the development of new business document implementation guidelines, avoids heterogeneous interpretations of the applied Core Components, and, finally, improves the interoperability and re-usability of business information.* Therefore, this Doctoral Thesis concludes that our business context aware Core Components modeling approach holds not only in theory, but also in practice.

9.2 Open Research Issues

This Doctoral Thesis provides the contributions in business context aware business documents modeling for inter-organizational business processes. However, some open research directions still remain and could be subject to further research. We enumerate them as: (i) the transformation of business document standards, (ii) the extension of business context reasoning foundations, (iii) the Hybrid business context model, and (iv) the business context aware inter-organizational business process modeling.

Transformation of Business Document Standards. A huge number of different business document standards, such as the traditional ones like UN/EDIFACT [8], EANCOM [133] or ASC X12 [134] and XML-based ones like xCBL [135], UBL [136] or CIDX [137] is concurrently exploited in business. This diversity undermines the global use of electronic business information in an interoperable, secure, and consistent manner by all business partners. For this reason, UN/CEFACT has proposed a conceptual Core Components based modeling approach on the top of these standards. This Doctoral Thesis envisions that transformations from a Core Components model to the different business document standards are established. The presented

work was concentrated on the conceptual level of the Core Components, and it did not consider their transformations to other business document standards.

Therefore, the development of the transformation rules (which translate contextualized Core Components to other document standards) opens new horizons in this research domain. Having such an approach at hand, will streamline the application of the Core Components based standardization as well as our context aware Core Components modeling approach.

Business Context Reasoning. Both the Enhanced UN/CEFACT Model and the Business Context Ontology Model provide the foundations for business context reasoning. The reasoning techniques supported by the Enhanced UN/CEFACT Model are explained in Chapter 4.2.3, while the reasoning techniques supported by the Business Context Ontology Model are explained in Chapter 5.1.3. In both cases the business context reasoning is underpinned by the two main tenets: (i) learning from a business context model and (ii) learning from a business contextual knowledge database. The reasoning capabilities applied in the scope of our business context aware Core Components modeling approach are built upon these reasoning tenets (explained in Chapters 6.3.3 and 6.5.4).

As highlighted in the previous Chapters, this Doctoral Thesis establishes only the foundations for business context reasoning. It stipulates that the existing reasoning instruments (such as description logic based reasoning) can be applied to business context reasoning. However, the thesis neither discusses the performances of these techniques nor develops new approaches for their improvements. This scope of the work passes the boundaries of our business informatics research domain and enters the domain of computational logic.

Hybrid Business Context Model. The evaluation of the Enhanced UN/CEFACT Model and Business Context Ontology Model (based on the 16 evaluation criteria which are introduced in Chapter 8.4.1) has stipulated that in some situations these models perform contradictory behaviours. Consequently, the choice of the business context model tightly depends on the particular business scenario and its requirements. The development of more sophisticated and computationally expensive business context reasoning capabilities (suggested in the previous Subsection) will additionally broaden the differences between cases using the Enhanced UN/CEFACT Model and Business Context Ontology Model.

We argue that combining both the Enhanced UN/CEFACT and the Business Context Ontology Models into a single Hybrid Business Context Model could overcome the disadvantages and unite the advantages typical when these models are applied as single units. For instance, the unique business context model could be developed as a three level model. The first level could be implemented by means of the Enhanced UN/CEFACT Model. The second level could be implemented by means of the Business Context Ontology Model. Finally, the third level could represent all pluggable ontologies located in the scope of Linked Open Data [73]. Depending on the current business needs and richness of the business contextual knowledge base, different levels of the Hybrid Business Context Model could be involved for business context aware Core Components modeling.

Business Context Aware Inter-organizational Business Process Modeling. An artifact-centric inter-organizational business process modeling [138–140] is an approach where the changes and evolutions of business data are considered as the main driver of the processes. Accordingly, artifacts combine both data aspects and process aspects into holistic units, and

serve as the basic building blocks from which models of business operations and processes are constructed.

This Doctoral Thesis provides the methods for supporting the re-use of already existing electronic business documents of already existing inter-organizational business processes. The following research could focus on developing the analogous approaches to automatically generate the other artifacts of inter-organizational business processes (such as business activities, swimlanes and business roles). Finally, the corresponding research direction could propose the techniques to combine these generated process building blocks into new inter-organizational business processes.

The described list of possible future research directions is not finite. The new research developments in the corresponding domain will be regularly published on the Web location [132].

List of Abbreviations

ABIE	Aggregate Business Information Entity
ACC	Aggregate Core Component
ANSI	American National Standards Institute
ASBIE	Association Business Information Entity
ASCC	Association Core Component
ASMA	Message Assembly
BBIE	Basic Business Information Entity
BC	Business Context
BCC	Basic Core Component
BCG	Business Context Graph
BCOnt	Business Context Ontology Model
BDoc	Business Document
BDocIG	Business Document Implementation Guideline
BDT	Business Data Type
BIE	Business Information Entity
CC	Core Component
CCL	The Core Component Library
CCTS	The Core Components Technical Specification
CDT	Core Data Type
CustBDocIG	Customized Business Document Implementation Guideline
DAG	Directed Acyclic Graph
DL	Description Logic
DSR	Design Science Research
E-UCM	Enhanced UN/CEFACT Business Context Model
ebXML	Electronic Business Using eXtensible Markup Language
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange For Administration, Commerce and Transport
ExistBDocIGs	Existing Business Document Implementation Guideline
FAO	Food and Agriculture Organization of the United Nations
GenBDocIG	Generic Business Document Implementation Guideline
IN	Invoice Document
INIG	Invoice Implementation Guideline
ISIC	International Standard Industrial Classification
LOD	Linked Open Data

MA	Message Assembly
NDR	Naming and Design Rules
OASIS	The Organization for the Advancement of Structured Information Standard
PO	Purchase Order Document
POIG	Purchase Order Implementation Guideline
SBDH	Standard Business Document Header
SGML	Standard Generic Markup Language
UCM	The UN/CEFACT Business Context Model
UML	Unified Modeling Language
UN/CEFACT	The United Nations Centre for Trade Facilitation and Electronic Business

Bibliography

[1] John Berge. *The EDIFACT Standards*. NCC Blackwell Limited, Oxford, England, 1991.

[2] Charles F. Goldfarb and Paul Prescod. *Charles F. Goldfarb's XML Handbook, Fifth Edition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2003.

[3] UN/CEFACT. Core Components Technical Specification CCTS, version 3.0. `http://www.unece.org/`, September 2009. Last Visit: August 2014.

[4] UN/CEFACT. Unified Context Methodology technical specification. `http://www1.unece.org/`, January 2011. Last Visit: August 2014.

[5] Alan Hevner and Samir Chatterjee. *Design Research in Information Systems*. Springer US, 2010.

[6] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the Conference on Human Factors in Computing Systems (CHI 2000)*, April 2000.

[7] The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). `http://www.unece.org/cefact`. Last Visit: August 2014.

[8] UN/CEFACT. United Nations Electronic Data Interchange For Administration, Commerce and Transport (UN/EDIFACT). `http://www.unece.org/`, 1988. Last Visit: August 2014.

[9] UN/CEFACT OASIS. ebXML - Technical Architecture Specification v1.0.4. `http://www.ebxml.org/`, February 2001. Last Visit: August 2014.

[10] UN/CEFACT. UN/CEFACT's Core Component Library UN/CCL version 13B. `http://www.unece.org/`, 2014. Last Visit: August 2014.

[11] UN/CEFACT. UN/CEFACT Core Components Data Type Catalogue. `http://www.unece.org/`, October 2011. Last Visit: August 2014.

[12] Central European Free Trade Agreement. Central European Free Trade Agreement. `http://http://www.cefta.int/`, 2006. Last Visit: August 2014.

[13] World Trade Organization. WTO Agreement on Textiles and Clothing. http://www.wto.org/, 2005. Last Visit: August 2014.

[14] UN/CEFACT. UN/CEFACT XML Naming and Design Rules technical specification version 3.0. http://www.unece.org/, 2009. Last Visit: August 2014.

[15] Danijel Novakovic and Christian Huemer. Applying business context to calculate subsets of business document standards. *Information Technology and Management*, (to appear), 2014.

[16] Danijel Novakovic and Christian Huemer. A survey on business context. In Durga Prasad Mohapatra and Srikanta Patnaik, editors, *Intelligent Computing, Networking, and Informatics, (Proceedings of the International Conference on Advanced Computing, Networking, and Informatics, ICACNI 2013, Raipur, Chhattisgarh, India, 12-14 June 2013)*, volume 243 of *Advances in Intelligent Systems and Computing*. Springer, 2014.

[17] Bill N. Schilit and Marvin M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, Sept 1994.

[18] Peter J. Brown, J. D. Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. *Personal Communications, IEEE*, 4(5):58–64, Oct 1997.

[19] Jason Pascoe Ryan Nick and David Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology 1997*, British Archaeological Reports, Oxford, October 1998. Tempus Reparatum.

[20] Anind K. Dey, Gregory D. Abowd, and Andrew Wood. Cyberdesk: a framework for providing self-integrating context-aware services. *Knowl.-Based Syst.*, 11(1):3–13, 1998.

[21] Mary Bazire and Patrick Brézillon. Understanding context before using it. In Anind Dey, Boicho Kokinov, David Leake, and Roy Turner, editors, *Modeling and Using Context: 5th International and Interdisciplinary Conference*, pages 29–40. Springer-Verlag, Berlin, 2005.

[22] Jeanne E. Parker, Debra L. Hollister, Avelino J. Gonzalez, Patrick Brézillon, and Shane T. Parker. Looking for a synergy between human and artificial cognition. In Patrick Brézillon, Patrick Blackburn, and Richard Dapoigny, editors, *Modeling and Using Context (Proceedings of the Eighth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'13), Annecy, France, 2013.)*, volume 8175 of *Lecture Notes in Computer Science*, pages 45–58. Springer Berlin Heidelberg, 2013.

[23] Roland Klemke. Context framework - an open approach to enhance organisational memory systems with context modelling techniques. In Ulrich Reimer, editor, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM 2000), Switzerland, October 30-31, 2000*, volume 34 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2000.

[24] Peter J. Brown. The stick-e document: a framework for creating context-aware applications. In *Proceedings of the 3rd international conference on Intelligent user interfaces, 1996.*, pages 259–272. also published in it EP–odd, January 1996.

[25] Andy Ward, A. Jones, and A. Hopper. A new location technique for the active office. *Personal Communications, IEEE*, 4(5):42–47, October 1997.

[26] David Franklin and Joshua Flachsbart. All gadget and no representation makes jack a dull environment. In *Proceedings of AAAI Spring Symposium on Intelligent Environments. AAAI TR*, pages 155–160, 1998.

[27] Tom Rodden, Keith Chervest, Nigel Davies, and Alan Dix. Exploiting context in HCI design for mobile systems. In *Proceedings of the First Workshop on Human Computer Interaction with Mobile Devices*, page 3, Glasgow, Scotland, 1998-05-21/1998-05-22.

[28] Richard Hull, Philip Neaves, and James Bedford-Roberts. Towards situated computing. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, pages 146–, Washington, DC, USA, 1997. IEEE Computer Society.

[29] Jeremy R. Cooperstock, Koichiro Tanikoshi, Garry Beirne, Tracy Narine, and William A. S. Buxton. Evolution of a reactive environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 170–177, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[30] Martin Brown. Supporting user mobility. In *Proceedings of IFIP World Conference on Mobile Communications*, pages 69–77. Chapman and Hall, 1996.

[31] Scott Elrod, Gene Hall, Rick Costanza, Michael Dixon, and Jim Des Rivières. Responsive office environments. *Commun. ACM*, 36(7):84–85, July 1993.

[32] Stephen Fickas, Gerd Kortuem, and Zary Segall. Software organization for dynamic and adaptable wearable systems. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, pages 56–, Washington, DC, USA, 1997. IEEE Computer Society.

[33] Jun Rekimoto, Yuji Ayatsuka, and Kazuteru Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *Proceedings of the Second International Symposium on Wearable Computers, 1998.*, pages 68–75, 1998.

[34] Mr. Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2Nd IEEE International Symposium on Wearable Computers*, ISWC '98, pages 92–, Washington, DC, USA, 1998. IEEE Computer Society.

[35] Nigel Davies, Keith Mitchell, Keith Cheverst, and Gordon Blair. Developing a context sensitive tourist guide. In *Proceedings of the First Workshop on Human Computer Interaction with Mobile Devices*, page 10, Glasgow, Scotland, 1998-05-21/1998-05-22.

[36] Jadwiga Indulska and Peter Sutton. Location management in pervasive systems. In CRPITS'03, editor, *Proceedings of the Australasian Information Security Workshop*, volume 21 of *CRPIT*, pages 143–151, Adelaide, Australia, 2003. ACS.

[37] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[38] Cristiana Bolchini, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A data-oriented survey of context models. *ACM SIGMOD Rec.*, 36(4):19–26, December 2007.

[39] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2(4):263–277, June 2007.

[40] P. Moore, Bin Hu, Xiaomei Zhu, W. Campbell, and M. Ratcliffe. A survey of context modeling for pervasive cooperative learning. In *Proceedings of the first IEEE International Symposium on Information Technologies and Applications in Education, 2007. ISITAE '07.*, pages K5–1–K5–6, Nov 2007.

[41] Ioanna Roussaki, Maria Strimpakou, Nikos Kalatzis, Miltiades E. Anagnostou, and Carsten Pils. Hybrid context modeling: A location-based scheme using ontologies. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 2–7. IEEE Computer Society, 2006.

[42] Bill N. Schilit, Norman I. Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, pages 85–90, Washington, DC, USA, 1994. IEEE Computer Society.

[43] Bill N. Schilit, Norman Adams, Rich Gold, Michael M. Tso, and Roy Want. The parctab mobile computing system. In *Proceedings of the Fourth Workshop on Workstation Operating Systems, 1993.*, pages 34–39, 1993.

[44] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. An overview of the PARCTAB ubiquitous computing experiment. *IEEE Personal Communications*, 2(6):28–33, December 1995.

[45] Michael Samulowitz, Florian Michahelles, and Claudia Linnhoff-Popien. Capeus: An architecture for context-aware selection and execution of services. In *Proceedings of the IFIP TC6 / WG6.1 Third International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, pages 23–40, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V.

[46] Steven J. DeRose. *The SGML FAQ Book: Understanding the Foundations of HTML and SGML.* Kluwer, 1997.

[47] Jadwiga Indulska, Ricky Robinson, Andry Rakotonirainy, and Karen Henricksen. Experiences in using cc/pp in context-aware systems. In *Proceedings of the 4th International Conference on Mobile Data Management*, MDM '03, pages 247–261, London, UK, UK, 2003. Springer-Verlag.

[48] WAPForum. User agent profile (uaprof). `http://www.wapforum.org/`, October 2001. Last Visit: August 2014.

[49] Albert Held, Sven Buchholz, Alexander Schill, and Alexander Schill. Modeling of context information for pervasive computing applications. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, July 2002.

[50] Ekaterina Chtcherbina and Marquart Franz. Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w) (L'Aquila/Italy)*, 2003.

[51] Lalana Kagal, Vlad Korolev, Harry Chen, Anupam Joshi, and Timothy Finin. Centaurus A framework for intelligent services in a mobile environment. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, ICDCSW '01, pages 195–, Washington, DC, USA, 2001. IEEE Computer Society.

[52] Grady Booch, Ivar Jacobson, and James Rumbaugh. *OMG Unified Modeling Language Specification Version 1.4*.

[53] Quanzheng Sheng and B. Benatallah. Contextuml: A uml-based modeling language for model-driven development of context-aware web services development. In *Proceedings of the International Conference on Mobile Business*, ICMB '05, pages 206–212, Washington, DC, USA, 2005. IEEE Computer Society.

[54] Terry A. Halpin. Object-role modeling. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1941–1946. Springer US, 2009.

[55] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Generating context management infrastructure from high-level context models. In *Proceedings of the Fourths International Conference on Mobile Data Management (MDM) - Industrial Track*, pages 1–6, 2003.

[56] Terry Halpin. *Information modeling and relational databases: from conceptual analysis to logical design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[57] G. M. Nijssen and Terry Halpin. *Conceptual Schema and Relational Database Design: a fact oriented approach*. Prentice-Hall, Sydney, Australia, 1989.

[58] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, 2nd Edition*. Benjamin/Cummings, 1994.

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

[60] Thomas Hofer, Wieland Schwinger, Mario Pichler, Gerhard Leonhartsberger, Josef Altmann, and Werner Retschitzegger. Context-awareness on mobile devices - the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9 - Volume 9*, HICSS '03, pages 292.1–, Washington, DC, USA, 2003. IEEE Computer Society.

[61] Albrecht Schmidt, Michael Beigl, and Hans-Werner Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

[62] Keith Cheverst, Keith Mitchell, and Nigel Davies. Design of an object model for a context sensitive tourist GUIDE. *Computers & Graphics*, 23(6):883–891, 1999.

[63] Bruno Bouzy and Tristan Cazenave. Using the object oriented paradigm to model context in computer go. In *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context*, 1997.

[64] John McCarthy and Sasa Buvac. Formalizing context (expanded notes). Technical report, Stanford, CA, USA, 1994.

[65] Chiara Ghidini and Fausto Giunchiglia. What is local models semantics? In Paolo Bouquet, Luciano Serafini, and Richmond H. Thomason, editors, *Perspectives on Context*, pages 19–41. CSLI Publications, Stanford, California, 2007.

[66] Varol Akman and Mehmet Surav. The use of situation theory in context modeling. *Computational Intelligence*, 13:427–438, 1997.

[67] Philip D. Gray and Daniel Salber. Modelling and using sensed context information in the design of interactive applications. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, EHCI '01, pages 317–336, London, UK, UK, 2001. Springer-Verlag.

[68] Kathrin Dentler, Ronald Cornet, Annette ten Teije, and Nicolette de Keizer. Comparison of reasoners for large ontologies in the OWL 2 EL profile. *Semantic Web*, 2(2):71–87, 2011.

[69] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 292–297. Springer Berlin Heidelberg, 2006.

[70] Rob Shearer, Boris Motik, and Ian Horrocks. HermiT: A highly-efficient OWL reasoner. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[71] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semant.*, 5(2):51–53, June 2007.

[72] Boris Motik. KAON2 - scalable reasoning over ontologies with large data sets. *ERCIM News*, 2008(72), 2008.

[73] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[74] Simon Perreault. vcard format specification. Internet RFC 6350, August 2011.

[75] The dublin core metadata initiative. `http://purl.oclc.org/dc/`, 1998. Last Visit: August 2014.

[76] Stuart A. Sutton and Jon Mason. The dublin core and metadata for educational resources. In Keizo Oyama and Hironobu Gotoda, editors, *Dublin Core Conference*, pages 25–31. National Institute of Informatics, Tokyo, Japan, 2001.

[77] Asma Ounnas, Ilaria Liccardi, Hugh C Davis, David E Millard, and Su A White. Towards a semantic modeling of learners for social networks. In *In SW-EL (2006), Workshop at the AH 06*, pages 1–6.

[78] Martin Hepp. Products and services ontologies: A methodology for deriving owl ontologies from industrial categorization standards. *Int. J. Semantic Web Inf. Syst.*, 2(1):72–99, 2006.

[79] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[80] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735, Busan, Korea, November 2007.

[81] John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards semantically-interlinked online communities. In *Proceedings of the Second European Conference on The Semantic Web: Research and Applications*, ESWC'05, pages 500–514, Berlin, Heidelberg, 2005. Springer-Verlag.

[82] Alexandre Passant, Uldis Bojars, John G. Breslin, and Stefan Decker. The sioc project: Semantically-interlinked online communities, from humans to machines. In *Proceedings of the 5th International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems*, COIN'09, pages 179–194, Berlin, Heidelberg, 2010. Springer-Verlag.

[83] Daniel L. Rubin, Holger Knublauch, Ray W. Fergerson, Olivier Dameron, and Mark A. Musen. Protégé-owl: Creating ontology-driven reasoning applications with the web ontology language. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA 2005), Washington, DC, USA, October 22-26, 2005*, 2005.

[84] Adam Pease and Christoph Benzmüller. Sigma: An integrated development environment for formal ontology. *AI Commun.*, 26(1):79–97, January 2013.

[85] Altova Gmb. Altova semanticworks 2012 user and reference manual. `http://www.altova.com/documents/SemanticWorks.pdf`, January 2012. Last Visit: August 2014.

[86] Adil Kabbaj. Development of intelligent systems and multi-agents systems with amine platform. In Henrik Schärfe, Pascal Hitzler, and Peter Øhrstrøm, editors, *Conceptual Structures: Inspiration and Application*, volume 4068 of *Lecture Notes in Computer Science*, pages 286–299. Springer Berlin Heidelberg, 2006.

[87] Xiaohang Wang, Daqing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22, 2004.

[88] Harry Chen, Timothy W. Finin, and Anupam Joshi. Using owl in a pervasive computing broker. In *OAS*, pages 9–16, 2003.

[89] Harry Chen, Filip Perich, Timothy W. Finin, and Anupam Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 258–267, 2004.

[90] Carsten Pils, Ioanna Roussaki, and Maria Strimpakou. Location-based context retrieval and filtering. In *Proceedings of the Worksop on location and context awareness (LOCA 2006)*, pages 256–273, 2006.

[91] Thomas Strang. *Service interoperability in ubiquitous computing environments*. PhD thesis, PhD thesis, Ludwig-Maximilians-University Munich, 2003.

[92] Sebastian Rudolph. Foundations of description logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter Patel-Schneider, editors, *Reasoning Web. Semantic Technologies for the Web of Data*, volume 6848 of *Lecture Notes in Computer Science*, pages 76–136. Springer Berlin Heidelberg, 2011.

[93] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. OWL Web ontology language guide. Technical Report 20040210, World Wide Web Consortium, 2004.

[94] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.

[95] Dejene Ejigu, Vasile-Marian Scuturici, and Lionel Brunie. An ontology-based approach to context modeling and reasoning in pervasive computing. In *Proceedings of the PerCom Workshops*, pages 14–19, 2007.

[96] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2(4):263–277, June 2007.

[97] Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. An ontology-based context model in intelligent environments. In *Proceedings of communication network and distributed systems modeling and simulation conference*, pages 270–275, 2004.

[98] Wonil Hwang, Jinyoung Park, Hyowon Suh, Hyungwook Kim, and Il Hong Suh. Ontology-based framework of robot context modeling and reasoning for object recognition. In *Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery*, FSKD'06, pages 596–606, Berlin, Heidelberg, 2006. Springer-Verlag.

[99] Simone Fuchs, Stefan Rass, Bernhard Lamprecht, and Kyandoghere Kyamakya. A model for ontology-based scene description for context-aware driver assistance systems. In *Proceedings of the 1st international conference on Ambient media and systems*, Ambi-Sys '08, pages 5:1–5:8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[100] Eun Jung Ko, Hyung Jik Lee, and Jeun Woo Lee. Ontology-based context modeling and reasoning for u-healthcare. *IEICE Transactions*, 90-D(8):1262–1270, 2007.

[101] Suparna De and Klaus Moessner. Ontology-based context inference and query for mobile devices. In *Proceedings of the 19th International IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE,2008.*, pages 1–5, Sept 2008.

[102] Danijel Novakovic and Christian Huemer. Business context sensitive business documents: Business context aware core components modeling using the E-UCM model. In *Proceedings of the 11th IEEE International Conference on Industrial Informatics (INDIN 2013), July 29-31, Bochum, Germany*, pages 523–528, IEEE, 2013.

[103] Danijel Novakovic, Christian Huemer, and Christian Pichler. Context model for business context sensitive business documents. In Patrick Brézillon, Patrick Blackburn, and Richard Dapoigny, editors, *Modeling and Using Context (Proceedings of the 8th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 13), October 28 - November 1, Annecy, France)*, volume 8175 of *Lecture Notes in Computer Science*, pages 336–342. Springer Berlin Heidelberg, 2013.

[104] Danijel Novakovic and Christian Huemer. Context aware business documents modeling. In Patrick Brézillon, Patrick Blackburn, and Richard Dapoigny, editors, *Modeling and Using Context (Proceedings of the 8th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 13), October 28 - November 1, Annecy, France)*, volume 8175 of *Lecture Notes in Computer Science*, pages 357–363. Springer Berlin Heidelberg, 2013.

[105] Department of Economic and Social Affairs Statistics Division. *International Standard Industrial Classification of All Economic Activities (ISIC), Revision 4*. United Nations Publications, 2009.

[106] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[107] Paul R. Halmos. *Naive Set Theory*. Springer New York, 1974.

[108] Michael R. A. Huth and Mark Ryan. *Logic in computer science: modelling and reasoning about systems*. Cambridge, 2000.

[109] Danijel Novakovic and Christian Huemer. Contextualizing business documents. In *Proceedings of the 10th IEEE International Conference on e-Business Engineering (ICEBE 2013), September 11-13, Coventry, United Kingdom*, pages 236–243, IEEE, 2013.

[110] Danijel Novakovic and Christian Huemer. Business context sensitive business documents: an Ontology based business context model for Core Components. In *Proceedings of the 10th International Conference for Informatics and Information Technology (CIIT 2013), April 18-21, Bitola, Macedonia*, 2013.

[111] Danijel Novakovic and Christian Huemer. Putting services in context. In *Proceedings of the 6th IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2013), December 16-18, Kauai, Hawaii, United States*, pages 38–42, IEEE, 2013.

[112] Caterina Caracciolo, Marta Iglesias Sucasas, and Johannes Keizer. Towards interoperability of geopolitical information within FAO. *Computing and Informatics*, 27(1):119–129, 2008.

[113] Jingzhi Guo. Inter-enterprise business document exchange. In *Proceedings of the 8th International Conference on Electronic Commerce: The New e-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet*, ICEC '06, pages 427–437, New York, NY, USA, 2006. ACM.

[114] Robert J. Glushko and Tim McGrath. Document engineering for e-business. In *Proceedings of the 2002 ACM Symposium on Document Engineering*, DocEng '02, pages 42–48, New York, NY, USA, 2002. ACM.

[115] B. Omelayenko. Preliminary ontology modeling for b2b content integration. In *Proceedings of the 12th International Workshop on Database and Expert Systems Applications, 2001.*, pages 7–13, 2001.

[116] Michael Strommer, Christian Pichler, and Philipp Liegl. A document model management framework based on core components. In *Proceedings of the 12th IEEE International Conference on Commerce and Enterprise Computing*, CEC '10, pages 72–79, Washington, DC, USA, 2010. IEEE Computer Society.

[117] HP Labs. Jena - A Semantic Web Framework for Java. http://jena.sourceforge.net/, 2004. Last Visit: August 2014.

[118] W3C SPARQL Working Group. SPARQL 1.1 overview. World Wide Web Consortium, Recommendation REC-sparql11-overview-20130321, March 2013.

[119] Lee Feigenbaum, Gregory Todd Williams, Kendall Grant Clark, and Elias Torres. SPARQL 1.1 protocol. World Wide Web Consortium, Recommendation REC-sparql11-protocol-20130321, March 2013.

[120] Birgit Hofreiter, Christian Huemer, and Wolfgang Klas. ebXML: status, research issues, and obstacles. In *Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on*, pages 7–16, 2002.

[121] Organization for the Advancement of Structured Information Standards (OASIS). https://www.oasis-open.org/. Last Visit: August 2014.

[122] Panu Korpipaa, Jani Mantyjarvi, Juha Kela, Heikki Keranen, and Esko-Juhani Malm. Managing context information in mobile devices. *Pervasive Computing, IEEE*, 2(3):42–51, July 2003.

[123] Juan Ye, Lorcan Coyle, Simon Dobson, and Paddy Nixon. Ontology-based models in pervasive computing systems. *ACM Knowl. Eng. Rev.*, 22(4):315–347, December 2007.

[124] Ramón Hervás, José Bravo, and Jesús Fontecha. A context model based on ontological languages: a proposal for information visualization. *J. UCS*, 16(12):1539–1555, 2010.

[125] Reto Krummenacher and Thomas Strang. Ontology-based context modeling. In *In Workshop on Context-Aware Proactive Systems*, 2007.

[126] Xiaohang Wang, Daqing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using OWL. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22, March 2004.

[127] Vipul Kashyap, Christoph Bussler, and Matthew Moran. *The Semantic Web, Semantics for Data and Services on the Web*. Springer-Verlag, Berlin, Heidelberg, 2008.

[128] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[129] Marie-Francine Moens. Evaluation of information extraction technologies. In *Information Extraction: Algorithms and Prospects in a Retrieval Context*, volume 21, pages 179–197. Springer Netherlands, 2006.

[130] Shian-Hua Lin and Jan-Ming Ho. Discovering informative content blocks from web documents. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 588–593, New York, NY, USA, 2002. ACM.

[131] B. Sheth and Pattie Maes. Evolving agents for personalized information filtering. In *Proceedings of the Ninth Conference on Artificial Intelligence for Applications, 1993.*, pages 345–352, Mar 1993.

[132] Danijel Novakovic, Vienna University of Technology. Business context aware Core Components modeling project. `http://web.student.tuwien.ac.at/~e1028296/phdwork/`. Last Visit: August 2014.

[133] UN/CEFACT. EANCOM standard specification. `http://www.unece.org/`, 2003. Last Visit: August 2014.

[134] American National Standards Institute ANSI. X12 industry implementation guidelines reference manual. Technical report, American Standards Committee Publication, USA, 1990.

[135] Commerce One Inc. XML Common Business Library (xCBL), version 4.0. `http://www.xcbl.org/`, 2003. Last Visit: August 2014.

[136] OASIS. Universal Business Language Version 2.1. `https://www.oasis-open.org/`, November 2013. Last Visit: August 2014.

[137] Chemical Industry Data Exchange. Chemical Industry Data Excahnge Standard (CIDX). `https://www.cidx.org/`, December 2013. Last Visit: August 2014.

[138] Tian Chao, David Cohn, Adrian Flatgard, Sandy Hahn, Mark Linehan, Prabir Nandi, Anil Nigam, Florian Pinel, John Vergo, and Fredericky Wu. Artifact-Based Transformation of IBM Global Financing. In Umeshwar Dayal, Johann Eder, Jana Koehler, and HajoA. Reijers, editors, *Business Process Management*, volume 5701 of *Lecture Notes in Computer Science*, pages 261–277. Springer Berlin Heidelberg, 2009.

[139] David Cohn and Richard Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009.

[140] Sira Yongchareon and Chengfei Liu. A process view framework for artifact-centric business processes. In *Proceedings of the 2010 International Conference on On the Move to Meaningful Internet Systems - Volume Part I*, OTM'10, pages 26–43, Berlin, Heidelberg, 2010. Springer-Verlag.