



UN/CEFACT

**DRAFT**

United Nations Centre for Trade Facilitation and Electronic Business

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

**UN/CEFACT – ebXML Core Components Technical  
Specification, Part 1**

**8 February 2002  
Version 1.8**



## 15    **1   Status of This Document**

16    This Technical Specification is being developed in accordance with the  
17    UN/CEFACT/TRADE/22 Open Development Process for Technical Specifications. It  
18    has been approved by the eBTWG for public review as defined in Step 5 of the Open  
19    Development Process.

20    This document contains information to guide in the interpretation or implementation  
21    of ebXML concepts.

22    Distribution of this document is unlimited.

23    The document formatting is based on the Internet Society's Standard RFC format.

24    This version: *Core Components* Technical Specification, Version 1.80 of 8 February  
25    2002

26    Previous version: *Core Components* Technical specification, Version 1.75 of 15  
27    January 2002

## 2 eBTWG - ebXML Core Components Specification

### Project Team Participants

We would like to recognise the following for their significant participation to the development of this document.

Project Team Leader:	Hartmut Hermes	Siemens
Lead Editor:	Mark Crawford	Logistics Management Institute
Editing Team	Mike Adcock	APACS
	Alan Stitzer	Marsh, Inc.
	James Whittle	e Centre
Contributors:	Mary Kay Blantz	IONA Technologies
	Bernd Boesler	DIN
	Todd Boyle	NetAccount
	Kerstien Celis	Seagha c.v.
	Jean-Luc Champion	Enterprise Integration Partners
	Marianne Cockle	APACS
	Scott Colthurst	State Farm
	Alain Dechamps	CEN/ISSS
	Arofan Gregory	CommerceOne
	Eduardo Gutentag	Sun Microsystems
	Paula Heilig	Worldspan
	Stig Korsgaard	Danish Bankers Association
	Richard May	Marsh, Inc
	Dale McKay	Logicon/Northrop Grumman
	Bill Murray	General Motors
	Margaret Pemberton	Diskray
	Sue Probert	CommerceOne
	Andreas Schultz	DKV
	Lisa Seaburg	CommerceOne
	Gunther Stuhec	SAP AG
	Hisanao Sugamata	ECOM-Japan
	Herbert Thomas	AustriaPro
	Frank Vandamme	SWIFT
	Tom Warner	Boeing
	James Werner	Boeing
	Nigel Wooden	Accord
	Joe Zurlo	Logistics Management Institute

### 66 **3 Table of Contents**

67	1	Status of This Document.....	2
68	2	eBTWG CC Project Team Participants .....	3
69	3	Table of Contents .....	4
70	4	Introduction.....	7
71	4.1	Scope and Focus .....	7
72	4.2	Structure of this Specification.....	8
73	4.2.1	Notation.....	8
74	4.3	Related Documents .....	9
75	4.4	Overview .....	10
76	4.5	Core Component Key Concepts.....	10
77	5	Working Process and Methodology.....	15
78	5.1	Overview .....	15
79	5.1.1	Discovery .....	15
80	5.1.2	How to use UN/CEFACT Core Components .....	16
81	5.1.2.1	Core Components and Semantic Interoperability .....	16
82	5.1.2.2	Overall Discovery and Document Creation .....	17
83	5.2	Core Components Discovery .....	20
84	5.2.1	Core Component Discovery – Preparation Steps.....	20
85	5.2.2	Core Components Discovery – Search Registry/Repository .....	22
86	5.2.3	Core Component Discovery – Basic Business Information Entities .....	24
87	5.3	Preparation for Submission.....	24
88	5.3.1	Applying the Naming Convention to a New Item .....	25
89	5.3.2	Preparation for Submitting New Aggregate Core Components .....	27
90	5.3.3	Preparation Steps for Requesting a New Basic Core Component .....	28
91	5.3.4	Preparation for Requesting a New Aggregate Business Information Entity which re-uses an Existing Aggregate Core Component .....	29
92			
93	5.4	Harmonisation.....	30
94	5.5	Technical Assessment and Approval.....	30
95	5.6	Context in the Discovery Process .....	31
96	5.6.1	Context Categories.....	32
97	5.6.2	Guidelines for Analyzing Business Information Entities in Context.....	33
98	6	Technical Details .....	37
99	6.1	Core Components and Business Information Entities .....	37
100	6.1.1	Core Components.....	37
101	6.1.2	Business Information Entities .....	40
102	6.1.3	Naming Convention .....	41
103	6.1.3.1	Core Component Naming Rules .....	41
104	6.1.3.1.1	Core Component Dictionary Information .....	41
105	6.1.3.1.2	Core Component General Rules .....	42
106	6.1.3.1.3	Core Component Rules for Definitions .....	42
107	6.1.3.1.4	Core Component Rules for Dictionary Entry Names .....	43
108	6.1.3.1.5	Rules for Core Component Business Terms.....	45
109	6.1.3.2	Rules for Business Information Entities .....	45
110	6.1.3.2.1	Business Information Entity Dictionary Information .....	45
111	6.1.3.2.2	Business Information Entity General Rules.....	46
112	6.1.3.2.3	Business Information Entity Rules for Definitions.....	46
113	6.1.3.2.4	Rules for Business Information Entity Dictionary Entry Names....	47

114	6.1.3.2.5	Rules for Business Information Entity Business Terms .....	48
115	6.1.3.3	List of Representation Terms.....	48
116	6.1.4	Catalogue of Core Components.....	50
117	6.1.5	Catalogue of Business Information Entities .....	52
118	6.2	Context.....	52
119	6.2.1	Overview of Context Specification.....	52
120	6.2.1.1	Context Categories.....	53
121	6.2.1.2	Constraint Language .....	53
122	6.2.1.3	Syntax Binding.....	54
123	6.2.2	Approved Context Categories.....	54
124	6.2.2.1	Business Process Context .....	55
125	6.2.2.2	Product Classification Context .....	56
126	6.2.2.3	Industry Classification Context.....	57
127	6.2.2.4	Geopolitical Context.....	57
128	6.2.2.5	Official Constraints Context .....	59
129	6.2.2.6	Business Process Role Context.....	59
130	6.2.2.7	Supporting Role Context.....	60
131	6.2.2.8	System Capabilities Context.....	60
132	6.2.3	Context Values.....	60
133	6.2.4	Core Components Context Constraints Language.....	61
134	6.2.4.1	Assembly Construct.....	67
135	6.2.4.2	ContextRule Construct.....	67
136	6.2.4.3	Output Constraints .....	68
137	6.2.4.4	Ordering and Application .....	68
138	7	Technical Details - Core Component Repository Storage.....	69
139	7.1	Storing Core Components.....	69
140	7.1.1	Stored Core Components .....	71
141	7.1.2	Stored Basic Core Components .....	71
142	7.1.3	Stored Core Component Types.....	71
143	7.1.4	Stored Aggregate Core Components .....	71
144	7.1.5	Stored Data Types.....	72
145	7.1.6	Stored Representation Term .....	72
146	7.1.7	Stored Supplementary Components.....	72
147	7.1.8	Stored Supplementary Component Value.....	73
148	7.1.9	Stored Content Components .....	73
149	7.1.10	Stored Content Component Restrictions.....	73
150	7.2	Stored Context .....	74
151	7.2.1	Context categories.....	75
152	7.2.2	Classification Scheme .....	76
153	7.2.3	Context Value .....	76
154	7.3	Stored Business Information Entities.....	76
155	7.3.1	Stored Aggregate Business Information Entities.....	78
156	7.3.2	Stored Assembly Component .....	78
157	7.3.3	Stored Assembly Data Type .....	79
158	7.3.4	Stored Basic Business Information Entities.....	79
159	7.3.5	Stored Basic Business Information Entity Data Types.....	79
160	7.3.6	Stored Business Information Entities.....	79
161	7.4	Core Component Storage Metadata .....	80
162	7.4.1	General Metadata Storage Rules.....	81
163	7.4.2	Management Information.....	82

164	7.4.2.1	Administrative Information .....	82
165	7.4.2.2	Status Information.....	82
166	7.4.2.3	Change History .....	82
167	7.4.2.4	Replacement Information.....	83
168	7.4.3	Content Information.....	83
169	7.4.3.1	Descriptive Information .....	83
170	7.4.3.2	Representation Information .....	84
171	7.4.3.3	Association Information.....	84
172	8	Approved Core Component Type, Content, and Supplementary	
173		Components .....	85
174	8.1	Approved Core Component Types .....	85
175	8.2	Approved Core Component Type Content and Supplementary	
176		Components .....	86
177	9	Definition of Terms.....	89
178	10	References.....	93
179	11	Disclaimer .....	95
180	12	Contact Information .....	96
181		Copyright Statement .....	97
182			

## 183 4 Introduction

184 This *Core Components* technical specification describes and specifies a new approach  
185 to the well-understood problem of the lack of information interoperability between  
186 applications in the e-business arena. Traditionally, standards for the exchange of  
187 business data have been focused on static message definitions that have not enabled a  
188 sufficient degree of interoperability or flexibility. A more flexible and interoperable  
189 way of standardising business semantics is required. The UN/CEFACT *Core*  
190 *Component* solution described in this technical specification presents a methodology  
191 for developing a common set of semantic building blocks that represent the general  
192 types of business data in use today.

193 The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD,  
194 SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in  
195 this document, are to be interpreted as described in Internet Engineering Task Force  
196 (IETF) Request For Comments (RFC) 2119.<sup>1</sup>

### 197 4.1 Scope and Focus

198 This *Core Components Technical Specification* can be employed wherever business  
199 information is being shared or exchanged amongst and between enterprises,  
200 governmental agencies, and/or other organisations in an open and worldwide  
201 environment. The prime users are business people, business process modellers, and  
202 application developers of different organisations that require interoperability of  
203 business information. This interoperability covers both interactive and batch  
204 exchanges of business data between applications through the use of Internet and Web  
205 based information exchanges as well as traditional Electronic Data Interchange (EDI)  
206 systems.

207 This document will form the basis for standards development work of business  
208 analysts, business users and information technology specialists supplying the content  
209 of and implementing applications that will employ the UN/CEFACT *Core Component*  
210 *Library* (CCL).

211 Due to the evolving nature of the UN/CEFACT *Core Component Library*, the  
212 specification includes material that focuses on the business community doing further  
213 discovery and analysis work. Some of the contents of this specification are not typical  
214 of this type of technical document. However, they are critical for successful adoption  
215 and standardisation in this area to move forward.

---

<sup>1</sup> Key words for use in RFCs to Indicate Requirement Levels - Internet  
Engineering Task Force, Request For Comments 2119, March 1997,  
<http://www.ietf.org/rfc/rfc2119.txt?number=2119>

## 216 4.2 Structure of this Specification

217 Due to the diversity of the intended audience, this document has been divided into  
218 five main Sections.

- 219 • Section 5: Working Process and Methodology for Business  
220 Users—Discovery, Harmonisation, Assessment and How to Use  
221 [informative]
- 222 • Section 6: Technical Details—*Core Components* and Context [normative]
- 223 • Section 7: Technical Details—Storage and Metadata [normative]
- 224 • Section 8: Technical Details—Approved *Core Component Type*, Content,  
225 and *Supplementary Components* [normative]
- 226 • Section 9: Definition of Terms [normative]

227 Sections 5, 6, 7 and 8 are complementary, but may also be used independently of each  
228 other. Section 5 is informative. A business audience may choose to read through the  
229 working process and methodology section (Section 5) and only reference the  
230 Technical Details (Sections 6, 7 and 8) as needed. Sections 6, 7 and 8 are normative.  
231 A technical audience may choose to focus on the technical details (Sections 6, 7, and  
232 8), referring to the methodology (Section 5) and example (Part 2 a separate document)  
233 sections as appropriate, using the current approved *Core Component Type*, *Content*,  
234 and *Supplementary Components* (Section 8) and the glossary (Section 9).

235 In addition, the *Core Components* Team has prepared the *Core Components Technical*  
236 *Specification*, Parts 2 and 3. *Part 2—Core Components Primer* details how the  
237 contents of Sections 5, 6, and 7 would be used. *Part 3—Catalogue of Discovered*  
238 *Core Components* represents the work of various organisations working in a joint  
239 endeavour to develop a beginning catalogue of *Core Components*.

### 240 4.2.1 Notation

241 [Definition] - A formal definition of a term. Definitions are normative.

242 [Example] - A representation of a definition or a rule. Examples are informative.

243 [Note] – Explanatory information. Notes are informative.

244 [Rn] - Identification of a rule that requires conformance to ensure discovered *Core*  
245 *Components* are properly discovered, named and stored. The value R is a prefix to  
246 categorise the type of rule where R=B for Business Information rule, R=C for *Core*  
247 *Component* rule, or R=S for Storage rule; and n (1..n) indicates the sequential number  
248 of the rule]. Rules are normative.



### 249 **4.3 Related Documents**

250 The following documents provided significant levels of influence in the development  
251 of this document:

- 252 — ebXML Technical Architecture Specification v1.04
- 253 — ebXML Business Process Specification Schema v1.01
- 254 — ebXML Registry Information Model v1.0
- 255 — ebXML Registry Services Specification v1.0
- 256 — ebXML Requirements Specification v1.06
- 257 — ebXML Collaboration-Protocol Profile and Agreement Specification v1.0
- 258 — ebXML Message Service Specification v1.0
- 259 — ebXML Technical Report, Business Process and Business Information  
260 Analysis Overview v1.0
- 261 — Business Process Analysis Worksheets & Guidelines v1.0
- 262 — ebXML Technical Report, E-Commerce Patterns v1.0
- 263 — ebXML Technical Report, Catalog of Common Business Processes v1.0
- 264 — ebXML Technical Report, *Core Component* Overview v1.05
- 265 — ebXML Technical Report, *Core Component* Discovery and Analysis v1.04
- 266 — ebXML Technical Report, Context and Re-Usability of *Core Components*  
267 v1.04
- 268 — ebXML Technical Report, Guide to the *Core Components* Dictionary v1.04
- 269 — ebXML Technical Report, Naming Convention for *Core Components* v1.04
- 270 — ebXML Technical Report, Document Assembly and Context Rules v1.04
- 271 — ebXML Technical Report, Catalogue of Context Categories v1.04
- 272 — ebXML Technical Report, *Core Component* Dictionary v1.04
- 273 — ebXML Technical Report, *Core Component* Structure v1.04
- 274 — Information Technology — Metadata registries: Framework for the  
275 Specification and Standardization of Data Elements, International  
276 Standardization Organization, ISO 11179-1
- 277 — Information Technology — Metadata registries: Classification of Concepts for  
278 the Identification of Domains, International Standardization Organization, ISO  
279 11179-2
- 280 — Information Technology — Metadata registries: Registry Metamodel,  
281 International Standardization Organization, ISO 11179-3
- 282 — Information Technology — Metadata registries: Rules and Guidelines for the  
283 Formulation of Data Definitions, International Standardization Organization,  
284 ISO 11179-4
- 285 — Information Technology — Metadata registries: Naming and Identification  
286 Principles for Data Elements, International Standardization Organization, ISO  
287 11179-5

288 — Information Technology — Metadata registries: Framework for the  
289 Specification and Standardization of Data Elements, International  
290 Standardization Organization, ISO 11179-6

## 291 4.4 Overview

292 This *Core Components Technical Specification* provides a way to identify, capture  
293 and maximise the reuse of business information to support and enhance information  
294 interoperability across multiple business situations. The specification focuses both on  
295 human-readable and machine-processable representations of this information.

296 The *Core Components* approach described in this document is more flexible than  
297 current standards in this area because the semantic standardisation is done in a syntax-  
298 neutral fashion. Using Core Components as part of the ebXML framework will help  
299 to ensure that two trading partners using different syntaxes (e.g. XML and EDIFACT)  
300 are using business semantics in the same way on condition that both syntaxes have  
301 been based on the same *Core Components*. This enables clean mapping between  
302 disparate message definitions across syntaxes, industry and regional boundaries.

303 UN/CEFACT *Business Process* and *Core Component* solutions capture a wealth of  
304 information about the business reasons for variation in message semantics and  
305 structure. In the past, such variations have introduced incompatibilities. The *Core*  
306 *Components* mechanism uses this rich information to allow identification of exact  
307 similarities and differences between semantic models. Incompatibility becomes  
308 incremental rather than wholesale, i.e. the detailed points of difference are noted,  
309 rather than a whole model being dismissed as incompatible.

## 310 4.5 Core Component Key Concepts

311 The key concepts in the *Core Components Technical Specification* are:

- 312 • *Core Component* — The *Core Component* is a semantic building block  
313 that is used as a basis to construct all electronic business messages.

314 [Definition] *Core Component* (CC)

315 A building block for the creation of a semantically correct and meaningful  
316 information exchange ‘parcel’. It contains only the information pieces necessary to  
317 describe a specific concept.

- 318 • *Business Context* – *Business Context* is a mechanism for qualifying and  
319 refining *Core Components* according to their use within a particular  
320 process. Once business contexts are identified, the appropriate *Core*  
321 *Components* can be selected or created and differentiated to indicate any  
322 necessary qualification and refinement needed to support the business  
323 process in a given business context.

324 [Definition] *Business Context*

325 The formal description of a specific business circumstance as identified by the values  
 326 of a set of context categories, allowing different business circumstances to be  
 327 uniquely distinguished.

- 328 • *Business Information Entity* –When a *Core Component* is used in a real  
 329 business situation it is used to define a *Business Information Entity*. The  
 330 *Business Information Entity* is the result of using a *Core Component*  
 331 within a specific business context.

332 [Definition] *Business Information Entity* (BIE)

333 A piece of business data or a group of pieces of business data with a unique business  
 334 semantic definition. A *Business Information Entity* can be either a *Basic Business*  
 335 *Information Entity* (BBIE) or an *Aggregate Business Information Entity* (ABIE).

336 There are three different categories of *Core Components*: *Basic Core Component*,  
 337 *Core Component Type* and *Aggregate Core Component*. The following definitions  
 338 explain each of these:

339 [Definition] *Basic Core Component* (BCC)

340 A *Core Component* that represents a singular business concept with a unique business  
 341 semantic definition. A *Basic Core Component* is constructed by using a *Core*  
 342 *Component Type*. *Basic Core Components* are used in developing *Aggregate Core*  
 343 *Components*.

344

345 [Definition] *Core Component Type* (CCT)

346 A *Core Component* that consists of one and only one *Content Component* that carries  
 347 the actual content plus one or more *Supplementary Components* giving an essential  
 348 extra definition to the *Content Component*. *Core Component Types* do not have  
 349 business meaning.

350

351 [Example] *Core Component Types*

352 For a *Core Component Type* of *Amount. Type*, the *Content Component* carries the  
 353 value of *12* and this value has no meaning on its own. But *12 Kilometres* or *12 Euro*,  
 354 where *Kilometres* or *Euro* are the *Supplementary Component* that gives essential extra  
 355 definition, do have meaning.

356

357 [Definition] *Aggregate Core Component*

358 A Collection of *Core Components* that convey a distinct business meaning. An  
359 *Aggregate Core Component* will consist of two or more *Basic Core Components*, or at  
360 least one *Basic Core Component* plus one or more *Aggregate Core Components*

361

362 [Example] – *Aggregate Core Component*363 Aggregate: *Financial Account. Details*

364 Definition: A service through a bank or other organisation through which funds are  
365 held on behalf of a client or goods or services are supplied on credit.

366 Basic Core Components:

367 Financial Account. Identifier

368 Financial Account. Name

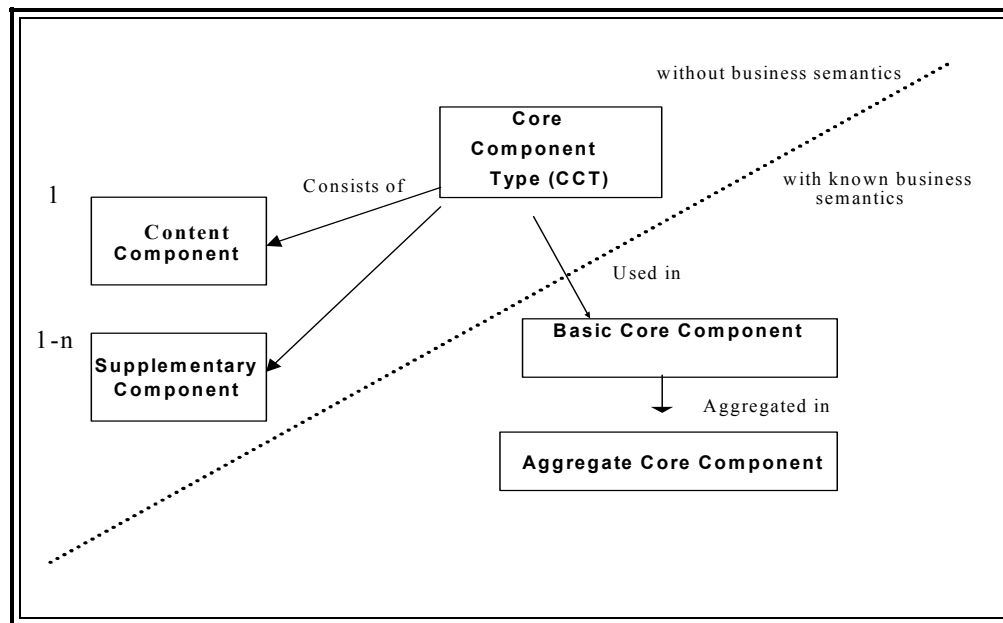
369 Financial Account. Country. Code

370 Financial Account Product. Type. Identifier

371 Financial Account Nickname. Name

372 The simple diagram in Figure 4-1 shows the relationships between these three  
373 categories.

374 **Figure 4-1. Core Component Overview**



375  
 376 A specific relationship exists between *Core Components* and *Business Information*  
 377 *Entities*. *Core Components* and *Business Information Entities* are complementary in  
 378 many respects. *Core Components* are intended to be the linchpin for creating  
 379 interoperable business process models and business documents using a *Controlled*  
 380 *Vocabulary*.

381 [Definition] *Basic Business Information Entity*

382 A *Core Component* used in a specific business context. A *Basic Business Information*  
 383 *Entity* is derived from a *Basic Core Component*.

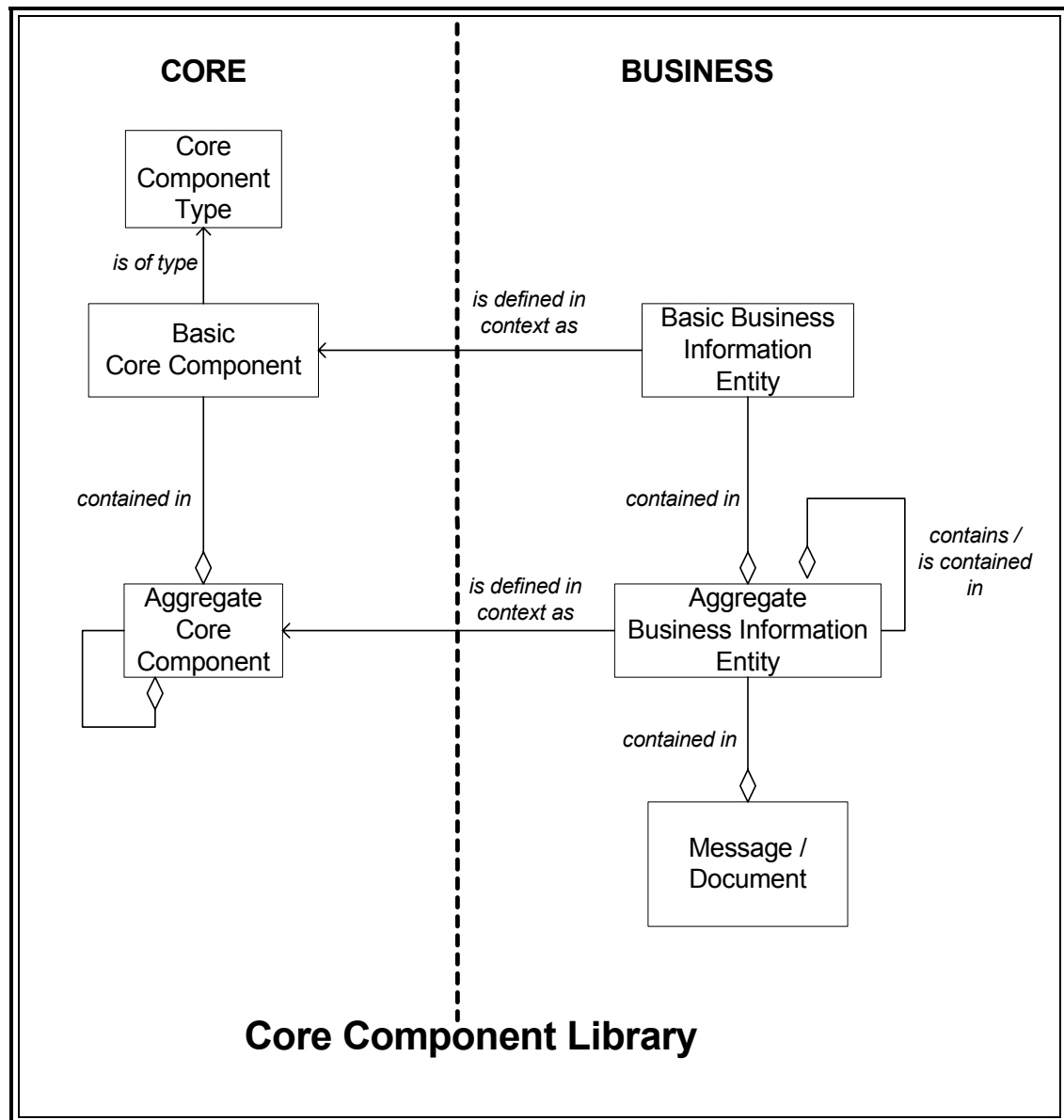
384

385 [Definition] *Aggregate Business Information Entity*

386 A collection of related pieces of business information that together convey a distinct  
 387 business meaning in a specified business context.

388 The features of the relationship between *Core Components* and *Business Information*  
 389 *Entities* are described in Figure 4-2.

390 **Figure 4-2. Relationships between Core Components and Business Information**  
 391 **Entities**



394 **[Note]**

395 The term *Core Component* is used as a generic term that encompasses *Basic Core*  
 396 *Components* and *Aggregate Core Components*, and their associated *Core Component*  
 397 *Types*. Equally the term *Business Information Entity* is used as a generic term  
 398 encompassing *Basic Business Information Entities* and *Aggregate Business*  
 399 *Information Entities*.

## 5 Working Process and Methodology

This chapter identifies aspects of *Core Component* working processes and methodologies for use. It includes an overview of the discovery and use characteristics of *Core Components*. In addition, it includes detailed recommendations for conducting discovery, storage, approval, and application of context.

### 5.1 Overview

The analysis of business processes builds a picture of requirements, identifying the business collaboration, i.e. timing and purpose of each process step. Detailed examination of the business processes at this level reveals the individual pieces of business information that are used and at what stage they are exchanged.

#### 5.1.1 Discovery

A business process should be modelled using a standard approach. UN/CEFACT requires the *UN/EDIFACT Modelling Methodology* (UMM) as the approach.<sup>2</sup> One of the results is a model (e.g. a class diagram) that shows the business information and its inter-relationships. *Business Information Entities* can be identified from the class diagram.

For example, if a domain team has modelled the publication of catalogue data to trading partners, the result will be a *Business Information Entity* representing the distributed catalogue data that is made up of a set of smaller *Business Information Entities* that are its component parts. Thus, the description of an item is identified as a *Business Information Entity* for this business process.

Ultimately, *Business Information Entities* must be based on a basic library of clearly defined semantic constructs to help ensure that they will inter-operate. This library must include a set of globally agreed semantic definitions such as those, which will be contained in the *UN/CEFACT Core Components Library*.

A *Business Information Entity* is a *Core Component* used in a specific business context and given its own unique name. As *Basic Core Components* are single pieces of business information, when they are used directly in specific business contexts, the structure (components) does not change.

---

<sup>2</sup> The UN/CEFACT Modelling Methodology (UMM) is a methodology for business process and information modelling that is based on the Unified Modelling Language.

[Example]

An invoicing business process uses a piece of information such as *VAT Amount*. *VAT Amount* is a *Basic Business Information Entity* that is based on the *Basic Core Component* of *Tax. Amount*. The invoicing business process is using *Tax. Amount* in a specific business context where the *Business Process Context* = *Purchasing*, and the *Geopolitical Context* = *EU*. Therefore the application of context adds a specialised definition, but in all other respects the *Basic Business Information Entity* is the same as the associated *Core Component* of *Tax. Amount*, i.e. it has the same structure and data type.

Just as each *Business Information Entity* must ultimately be based on *Basic Core Components*, each *Aggregate Business Information Entity* must ultimately be based on an existing *Aggregate Core Component*. The underlying *Aggregate Core Component* identifies the generic, standard definition of business information that is being used in the *Aggregate Business Information Entity*. The *Aggregate Business Information Entity* inherits the generic description, which is then modified and enhanced to be specific to the business process in which the *Aggregate Business Information Entity* is used. An *Aggregate Business Information Entity* is thus directly tied to a specific business process, or to a *Business Context*. (See Section 5.6 for a fuller understanding of context.)

An important aspect of information interoperability is that each *Business Information Entity* inherits a *Core Component* structure and associated semantic definitions derived from the *Core Component Library*.

The following section describes the procedures by which the UN/CEFACT ebXML compliant library may be developed and maintained.

### 5.1.2 How to use UN/CEFACT Core Components

This section, 5.1.2, provides a procedure for the technical user who wants to understand how to use *Core Components*. It assumes the user is dealing with an established set of *Core Components*, context categories and metadata/storage. The established set of *Core Components* being used should be based on those discovered, harmonised, and published by recognised standards groups. It is further assumed that the recognised standards group(s) and other business association group(s) have also made available sets of *Business Information Entities* for use in a published set of business processes.

#### 5.1.2.1 Core Components and Semantic Interoperability

Today, the e-business community generally agrees on the definition of a standard message structure expressed as an UN/EDIFACT *Message Implementation Guide*



466 (MIG), an XML schema, or similar syntax specific representation. UN/CEFACT will  
 467 produce standards based representations of these artefacts for implementation.<sup>3</sup>

468 Under the *Core Components* concept, defining and storing *Core Components* and  
 469 associated context mechanisms occur prior to the creation of a MIG or a XML  
 470 schema. In this manner, the focus of the user changes from examining the MIG or  
 471 XML schema, and moves to an examination of the semantic models. Accordingly,  
 472 interoperability between syntaxes no longer depends on analysing specific instances,  
 473 but naturally occurs during the business process model definition phase.

#### 474 5.1.2.2 Overall Discovery and Document Creation

475 Overall discovery and document creation can be thought of as a series of steps that  
 476 starts with determining the availability of existing business process definitions and  
 477 ultimately results in standard business documents. Figure 5-1 illustrates this process.  
 478 Specific steps to be followed are further described below.  
 479

480 Step 1: Search the registry/repository— A search should be made in the registry on  
 481 all available published business processes in the repository to find an inter-  
 482 operable business process that meets the business requirement.

483 Step 1a: If no existing business process is found to be appropriate, then the new  
 484 business process should be modelled using *UN/CEFACT Modelling*  
 485 *Methodology* and submitted to the registry.

486 Step 1b: Conduct a thorough analysis of the business information requirements by  
 487 following the *Core Component Discovery Steps* (Section 5.2.2)

488 Step 2: Identify relevant context categories - Access the registry interface and  
 489 identify the relevant context categories of the selected business process by  
 490 determining the following context categories (See Section 6.2.2):

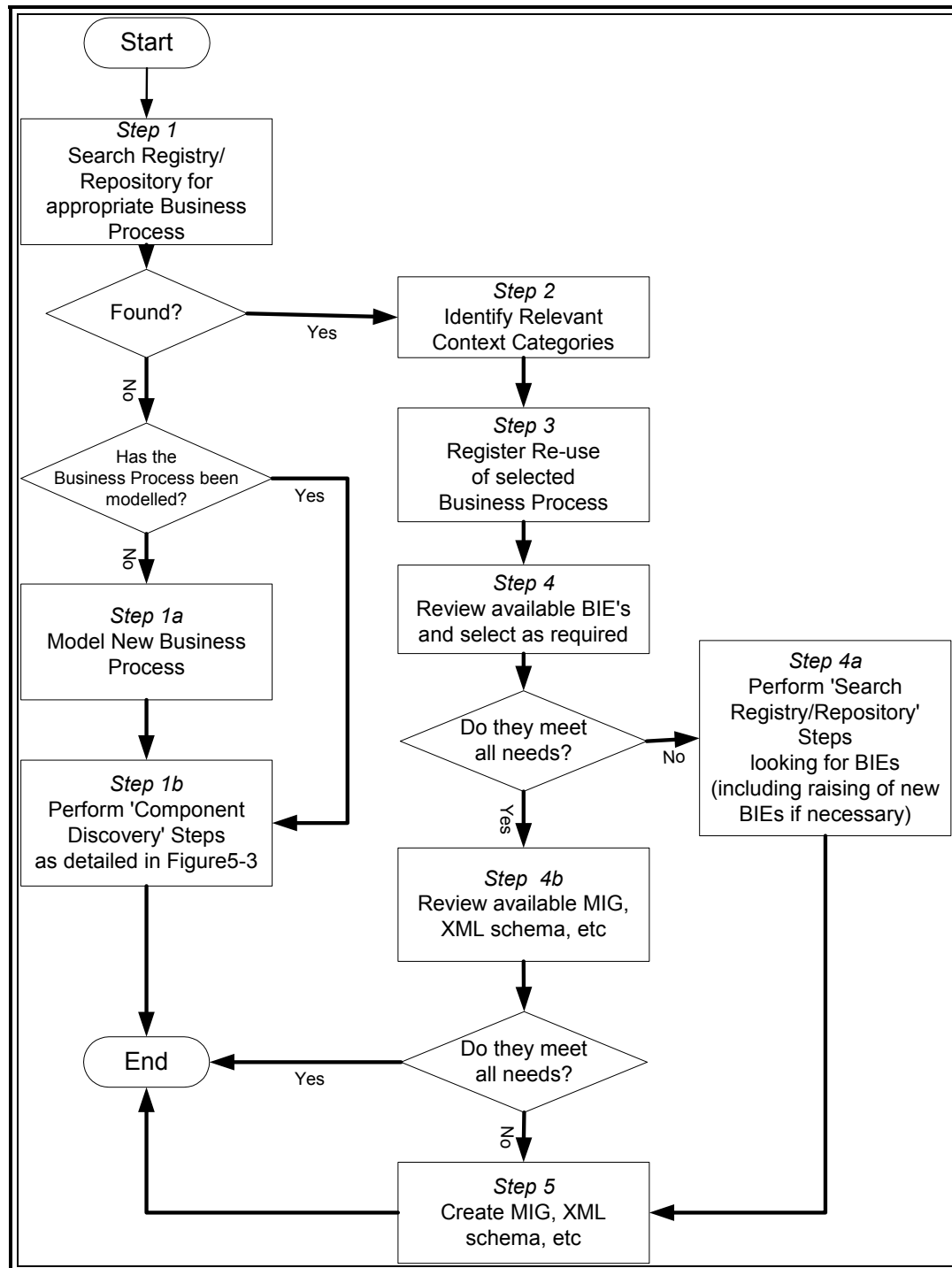
- 491 • *Business Process Context* – Identify the role played by the user and their  
 492 trading partners.
- 493 • *Product Classification Context* – Determine the goods or services  
 494 concerned in the collaboration.
- 495 • *Industry Classification Context* – Determine the relevant trading partner  
 496 industries.
- 497 • *Geopolitical Context* – Determine where the business process is to be  
 498 conducted. Determine if the business process crosses international  
 499 boundaries.
- 500 • *Official Constraints Context* – Determine any legal restrictions or  
 501 requirements on this business process.

---

<sup>3</sup> The term XML schema includes XML Schema as defined in World Wide Web Consortium Extensible Markup Language Version 1.0 and XML Document Type Definitions.

- *Business Process Role Context* – Identify the roles played by the user and their trading partners. These can be derived from the business process.

**Figure 5-1. Steps from Business Process Discovery to Core Component Discovery**



505

- *Supporting Role Context* – Determine what other significant parties will be using the data in the messages. Determine their role in the overall process.

506  
507

- 508       • *System Capabilities Context* – Determine any major restrictions derived  
509       from legacy systems. Identify the type of system.
- 510       The registry will provide a list of pre-defined *Business Information Entities*  
511       that are available to the selected business process, and which meet the  
512       context criteria specified. These will come with identified relationships to the  
513       *Core Components* that they are based on, and the context rules/values that  
514       fully qualify them. The registry should also return partial matches with an  
515       indication of how closely they match the specified context.
- 516   Step 3:   Register re-use of selected *Business Process* in the set of contexts in which it  
517              is being used. Registration of each re-use ensures the gradual development  
518              of a library of re-uses that will be available to the widening user base.
- 519   Step 4:   Review the available *Business Information Entities* and select the  
520              appropriate subset for use that meets the needs of the business process  
521              requirement that is being developed.
- 522   Step 4a:   If the *Business Information Entities* available for the specific business  
523              process do not address all of the data requirements, the repository of all  
524              *Business Information Entities* should be searched to see if the appropriate  
525              *Business Information Entities* already exist. The procedure for this is  
526              described under Search Repository (Section 5.2), which includes the steps to  
527              raise any new *Business Information Entities*, required because no appropriate  
528              *Business Information Entities* can be found.
- 529   Step 4b:   If all required *Business Information Entities* are already available, review the  
530              available MIG, XML schema, etc and select the appropriate one(s) for use  
531              that meet the technical implementation/solution requirements identified. If  
532              no appropriate technical implementation/solution is already available,  
533              continue with Step 5 to create new ones.
- 534   Step 5:   Create MIG, XML schema, etc. – The resulting semantic model (the set of  
535              *Business Information Entities*) is manually or programmatically rendered  
536              into a syntax-specific message description. The resulting MIG, XML schema  
537              is submitted to the repository where it is associated with the *Business*  
538              *Information Entities* it represents.

[Note]

When selecting a business process and defining the required messages, searches may be made against potential trading partners' data requirements and processes. The context rules and *Business Information Entities* represent useful metadata in determining the best possible match between the user and their partners. The fact that the rules can be made available in processable formats means that the comparison itself could be automated and made available as a feature of the repository implementation.

## 5.2 Core Components Discovery

The steps in *Core Component* discovery are preparation and search. In order to properly define the *UN/CEFACT Core Component Library*, domain or project groups must follow the prescribed preparation and search steps as outlined in the following subsections. See **Part 2—Core Components Primer** for a detailed end-to-end example of discovering *Core Components*.

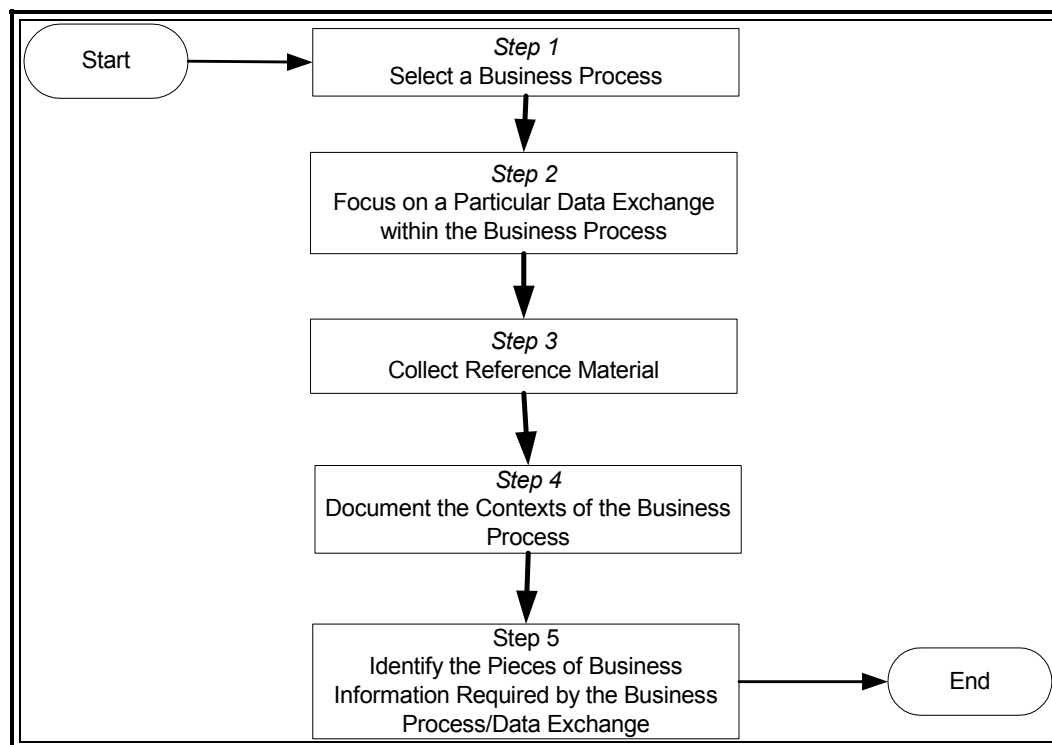
### 5.2.1 Core Component Discovery – Preparation Steps

These steps identify pieces of business information such as *Basic* and *Aggregate Business Information Entities*. An analysis of *Business Information Entities* from a variety of similar business processes leads to the underlying core structures and semantics of the *Core Components*. Figure 5-2 graphically portrays the prescribed preparation steps that are described below.

- Step 1. Select the *Business Process* that provides the widest range of business information content within the domain being addressed. (e.g. *Make a Payment, Place an Order, Issue an Invoice*)
- Step 2. Focus on a particular data exchange within the *Business Process* that contains key business information (e.g. *Payment Order, Purchase Order, Invoice*).
- Step 3. Collect all the business information and associated details that are relevant to the chosen business exchange for the previously identified business process. Use a cross section of *Message Implementation Guides*, *RosettaNet Partner Interface Process (PIP)*, *Business Process Information Models (BPIMs)* or similar domain-specific artefacts as sources of information about the business exchange.
- Step 4. Document the context(s) of the business process being analysed. Identify what is applicable for each category of context, i.e. whether it is *none*, *in all contexts*, or *one or multiple specific context value(s)*. (See Section 5.6 for a more detailed explanation of how to determine context). The context categories are:
  - *Business Process Context*

- 577 • *Product Classification Context*
- 578 • *Industry Classification Context*
- 579 • *Geopolitical Context*
- 580 • *Official Constraints Context*
- 581 • *Business Process Role Context*
- 582 • *Supporting Role Context*
- 583 • *System Capabilities Context*

584 **Figure 5-2 Preparation Steps**



585

586 Step 5. Compile a list of the pieces of information required for the business process.

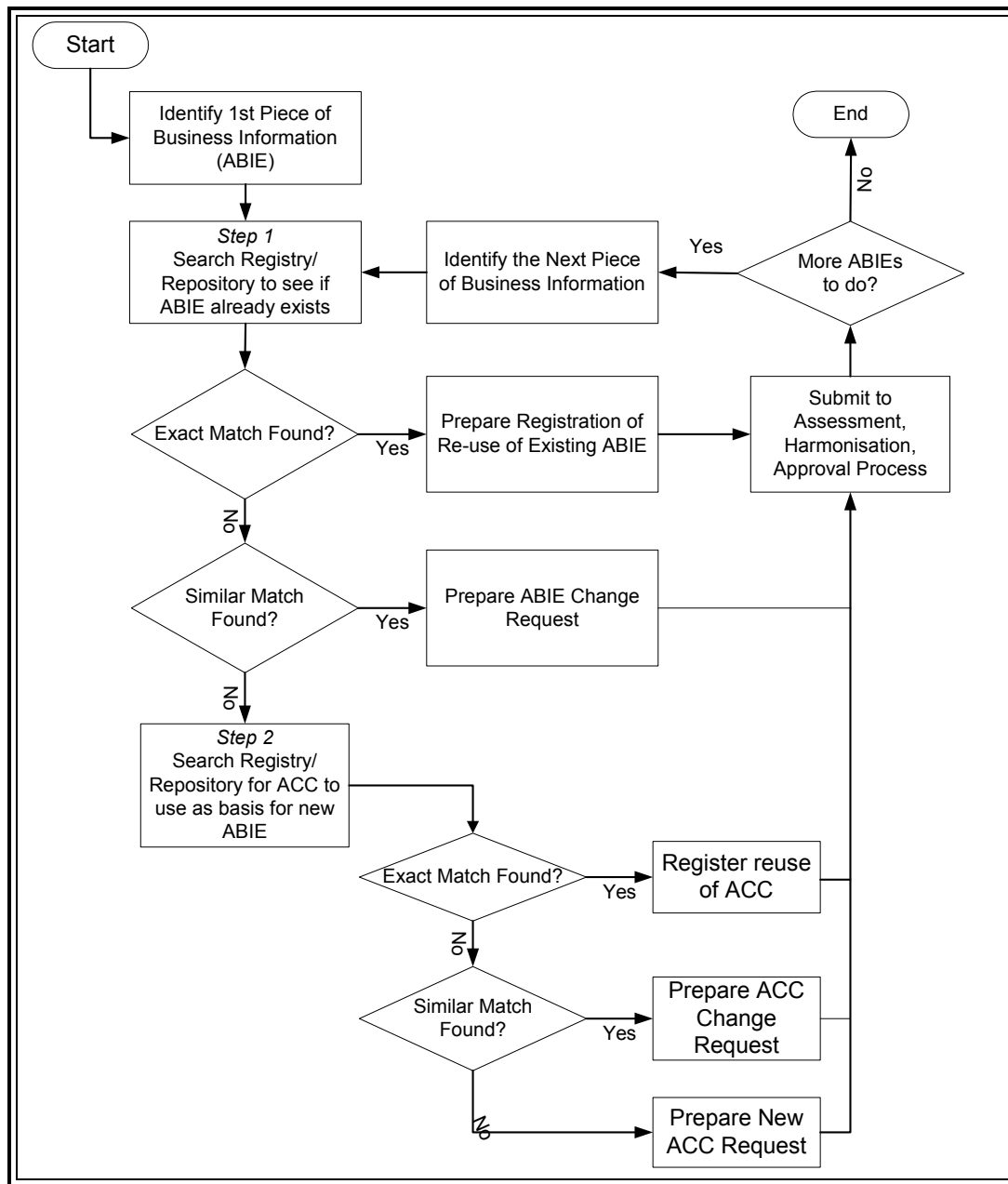
- 587 • If starting from a model (UN/CEFACT recommends UMM models of  
588 business processes), identify the objects (*Aggregate Business Information*  
589 *Entities*) that are needed.
- 590 • If not starting from a model, collect the pieces of information into object-  
591 like groups (*Aggregate Business Information Entities*). It is important to  
592 recognise and avoid pieces of information that are purely used for legacy  
593 system or syntax purposes.

- For each *Aggregate Business Information Entity*, capture its semantic definition, any *Business Terms* by which it is commonly known, and other information identified in the previous steps.

## 5.2.2 Core Components Discovery – Search Registry/Repository

Having discovered a number of *Aggregate Business Information Entities* in the preparation Step 5 identified in Section 5.2.1 above, repeat the following steps for each *Aggregate Business Information Entities* as shown in Figure 5-3.

**Figure 5-3 Search Steps**



603 Step 1 We recommend starting with *Aggregate Business Information Entities* at the  
 604 highest level of aggregation. Search the Catalogue of *Aggregate Business*  
 605 *Information Entities* for an existing *Aggregate Business Information Entity*  
 606 that has the same definition.

607 • If there is an *Aggregate Business Information Entity* with a definition that  
 608 meets the business need, register the re-use including business context and  
 609 any business terms. (Go to next *Aggregate Business Information Entity*)

610 • If there is an *Aggregate Business Information Entity* with a definition that  
 611 potentially could be modified to meet the business need, prepare an  
 612 *Aggregate Business Information Entity* change request for submission to  
 613 the harmonisation and approval process. Proposed changes need to be  
 614 assessed to ensure that any adaptation is sensible, reasonable and applied  
 615 in the most appropriate way. This, together with registration of re-uses,  
 616 will ensure the availability of a real and usable pool of material to a  
 617 widening user base. Include re-use, business context and any business  
 618 terms. (Go to next *Aggregate Business Information Entity*)

619 [Note]

620 Proposed changes need to be assessed to ensure that any adaptation is sensible,  
 621 reasonable and applied in the most appropriate way. This, together with registration of  
 622 re-uses, will ensure the availability of a real and usable pool of material to a widening  
 623 user base.

624 • If there is not an *Aggregate Business Information Entity* with a suitable  
 625 definition, go to Step 2.

626 Step 2 Search the *Catalogue of Core Components* for an existing *Aggregate Core*  
 627 *Component* that has the appropriate generic definition and structure from  
 628 which the new required *Aggregate Business Information Entity* can be  
 629 formed.

630 • If there is an existing *Aggregate Core Component* with a definition and  
 631 structure that meets the business needs, register the re-use of the *Aggregate*  
 632 *Core Component* as an *Aggregate Business Information Entity* including  
 633 the definition and name created according to the naming convention. (Go  
 634 to next *Aggregate Business Information Entity*)

635 • If there is an *Aggregate Core Component* with a definition and structure  
 636 that potentially could be modified to meet the business need, prepare an  
 637 *Aggregate Core Component* change request for submission to the  
 638 harmonisation and approval process. Include the re-use of the *Aggregate*  
 639 *Core Component* as an *Aggregate Business Information Entity*, including  
 640 the definition and name created according to the naming convention. (Go  
 641 to next *Aggregate Business Information Entity*)

- 642 • If there is not an *Aggregate Core Component* with a suitable definition  
643 and structure, prepare a new *Aggregate Core Component* request for  
644 submission to the harmonisation and approval process. Include the re-use  
645 of the *Aggregate Core Component* as an *Aggregate Business Information*  
646 *Entity*, including the definition and name created according to the naming  
647 convention. (Go to next *Aggregate Business Information Entity*)

### 648 **5.2.3 Core Component Discovery – Basic Business Information Entities**

649 This procedure is exactly the same as that described in Section 5.2.2, except that the  
650 reader should read *Basic Business Information Entity* for *Aggregate Business*  
651 *Information Entity* and *Basic Core Component* for *Aggregate Core Component*.

## 652 **5.3 Preparation for Submission**

653 Following the search of the *Core Component Library*, there may be a need to prepare  
654 submissions for the harmonisation and approval process. (See Section 5.4)

- 655 • Preparation of submissions will be carried out by the business domain or  
656 project group making the discovery.
- 657 • Harmonisation and approval will be conducted by appropriate Assessment,  
658 Harmonisation and Approval teams to be set up as part of the  
659 UN/CEFACT electronic business standards forum.

660 The different types of submissions that may be required are detailed below.

661 The following submissions are simple documented requests, following procedures to  
662 be established by the Assessment, Harmonisation and Approval teams.

- 663 • To register a Re-use of an existing *Aggregate Business Information Entity*
- 664 • To make a Change Request for an existing *Aggregate Business*  
665 *Information Entity*
- 666 • To make a Change Request for an existing *Aggregate Core Component*

667 The following submissions require more significant preparation, as part of the *Core*  
668 *Component* working methodology, to be carried out by the business domain or project  
669 group conducting the discovery and analysis.

- 670 • Preparation for Requesting a new *Basic Core Component*
- 671 • Preparation for Requesting a new *Aggregate Core Component*
- 672 • Preparation for Requesting a new *Aggregate Business Information Entity*  
673 which re-uses an existing *Aggregate Core Component*

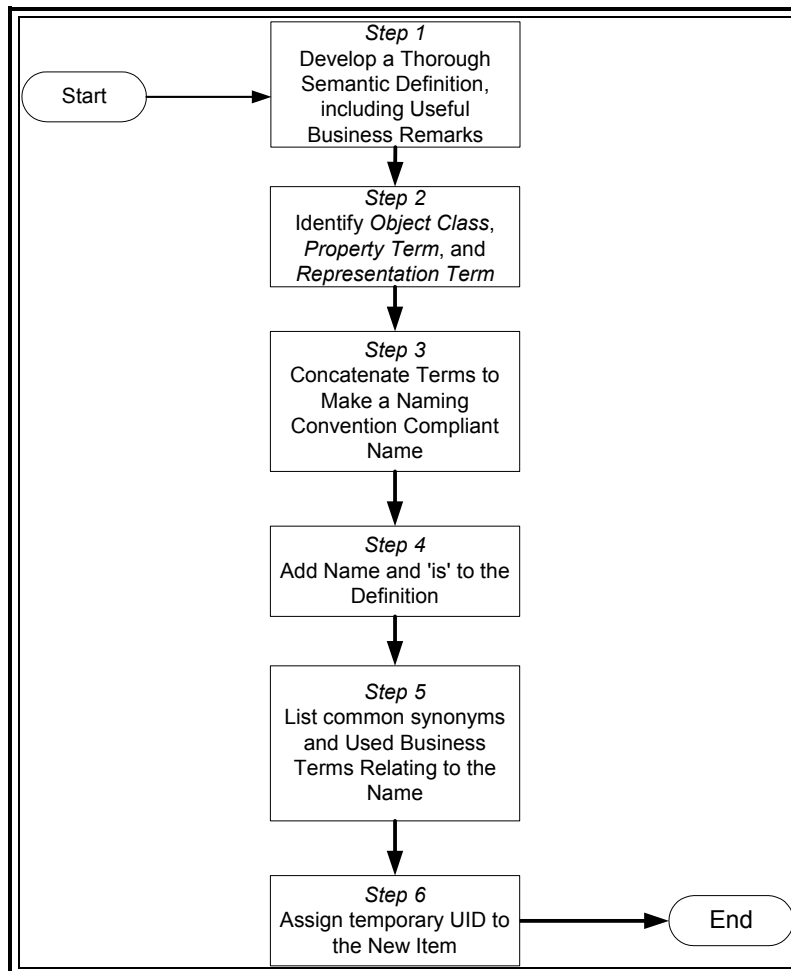


Each of these needs to initially follow the same steps in applying the *Naming Convention* (Section 6.1.3) to arrive at the name of the new item.

### 5.3.1 Applying the Naming Convention to a New Item

For all new items, the *Naming Convention* and associated rules that are defined in Section 6.1.3 must be applied. Figure 5-4 shows the steps that must be taken, each of which is described in the accompanying text.

**Figure 5-4 Applying the Naming Convention**



681

682 Step 1. Develop a thorough semantic definition and include any useful business  
683 comments as remarks. Semantic definitions should:

- 684 • use words different to those being defined *provided* that no ambiguity is  
685 thereby introduced,
- 686 • be globally applicable,
- 687 • be generic (i.e. able to cover the same business concept for different  
688 products/services),

- 689           • be applicable across multiple industries or domains, and
- 690           • be simple and clear to enable unambiguous translation to other languages
- 691   Step 2. Follow the *Naming Convention for Core Components or Business*
- 692           *Information Entities* (Section 6.1.3) to identify as appropriate:
- 693           • *Object Class*
- 694           • *Property Term*
- 695           • *Representation Term*
- 696           • *Qualifier Term(s)*
- 697   Step 3. Concatenate the terms to create a *Naming Convention* compliant name.

698   [Note]

699   The resultant name may seem artificial in that it might not be the same as any of the

700   business terms used for that concept. However, rigor of the *Naming Convention*

701   enables future translation of the name into other languages.

- 702   Step 4. Check the quality of the definition by adding the words “[*Dictionary Name*]
- 703           is” to the front of the definition, where [*Dictionary Name*] is the agreed
- 704           name.
- 705   Step 5. List common synonyms or *Business Term(s)* that are used within the domain
- 706           to identify the piece of business information (e.g. *Account Number*, *Account*
- 707           *Identifier*).

708   [Note]

709   Some *Business Terms* are used for several different pieces of business information. It

710   is perfectly acceptable to have the same business term listed as a synonym for two or

711   more pieces of business information. For example, as shown in Figure 5-5, *Account*

712   *Number* is a synonym for *Financial Account Identifier* and for *Sales Account*

713   *Identifier*.

- 714   Step 6. Assign a temporary UID to the new item in the form of a 6 digit
- 715           alphanumeric string, chosen at the discretion of the user.

716 **Figure 5-5 Core Component Catalogue Extract**

Temp UID	Definition	Remarks	Business Terms	CCT	Dictionary Entry Name			
					Name	Object Class	Property Term	Represent- ation Term
C00010	A Financial Account is a service through a bank or other organisation through which funds are held on behalf of a client or goods or services are supplied on credit	Not a general ledger.	Account	r/a	Financial Account. Details	Financial Account	Details	
F00012	A Sales Account is a relationship between a vendor and a customer.	Usually includes a contract specifying the terms of	Account	r/a	Sales Account. Details	Sales Account	Details	

Same Business Term

718 **5.3.2 Preparation for Submitting New Aggregate Core Components**

719 The development of a new aggregate requires adherence to the *Naming Convention*  
 720 rules for naming and definition. Once named, the new aggregate's constituent parts  
 721 need to be individually examined. The following diagram and text describes the  
 722 procedure that is to be followed.

723 Step 1. Apply the *Naming Convention* rules to arrive at the name of the new  
 724 *Aggregate Core Component*

725 Step 2. Identify all of the components within the new *Aggregate Core Component*.

726 Repeat the following step for each constituent component identified in step 2:

727 Step 3. Search the Registry for an existing *Core Component* that has the appropriate  
 728 generic definition and structure.

729 • If there is an existing *Core Component* with a definition and structure that  
 730 meets the requirement, register this re-use of the *Core Component*  
 731 including the context in which it is used.

732 • If there is an existing *Core Component* with a definition and structure that  
 733 potentially could be modified to meet the requirement, prepare a *Core*  
 734 *Component* change request for submission to the harmonisation and  
 735 approval process, including the re-use of the *Core Component* and the  
 736 context in which it is used.

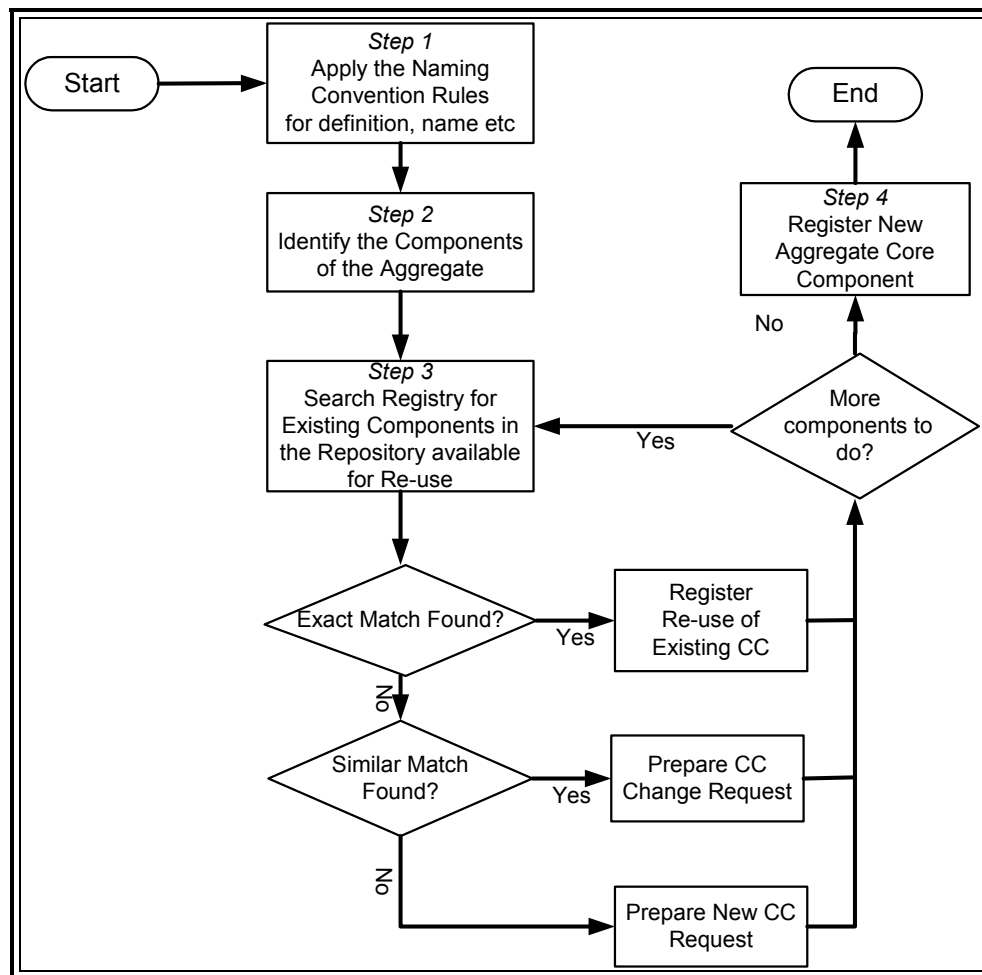
737 • If there is not an existing *Core Component* with a suitable definition and  
 738 structure, prepare a new *Core Component* request for submission to the  
 739 harmonisation and approval process, including the re-use of the *Core*  
 740 *Component* and the context.

741 When all the constituent components identified in step 2 have been checked as  
 742 described in Step 3, then:

743 Step 4. Register new *Aggregate Core Component*.

744 Prepare the new *Aggregate Core Component* request for submission to the  
 745 harmonisation and approval process.

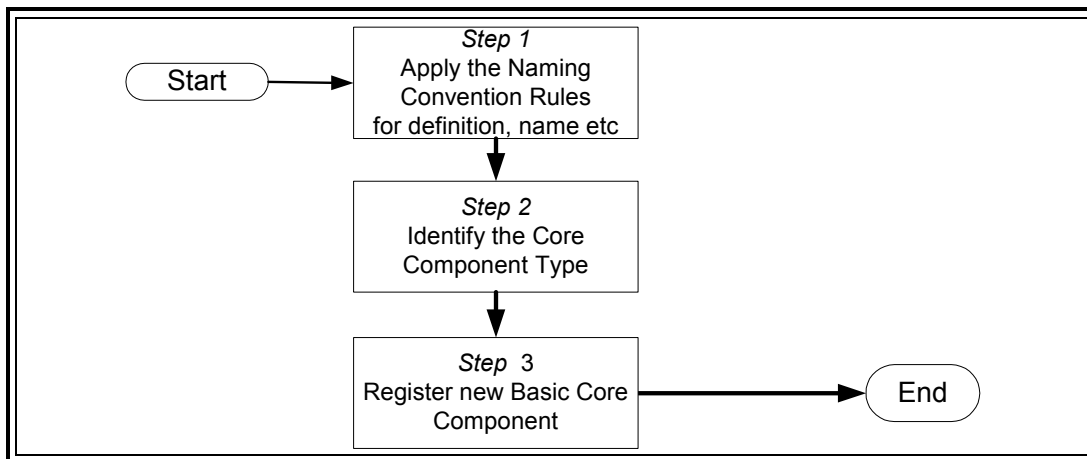
746 **Figure 5-6 Preparation for requesting a new Aggregate Core Component**



### 748 5.3.3 Preparation Steps for Requesting a New Basic Core Component

749 As shown in Figure 5-7, there are three steps necessary to prepare for requesting a  
 750 new *Basic Core Component*. These three steps are:

751 **Figure 5-7 Preparation Steps for Requesting a New Core Component.**



753 Step 1. Apply the *Naming Convention* and *Rules* to arrive at the name of the new  
754 *Basic Core Component*

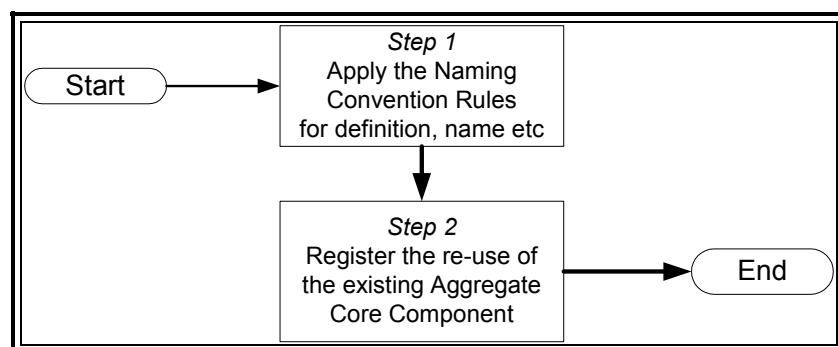
755 Step 2. Select the appropriate *Core Component Type*. (See Section 6.1.1 for an  
756 explanation and listing of *Core Component Types*).

757 Step 3. Register the new *Basic Core Component*

#### 758 **5.3.4 Preparation for Requesting a New Aggregate Business** 759 **Information Entity which re-uses an Existing Aggregate Core** 760 **Component**

761 As shown in Figure 5-8, there are two steps necessary to prepare for requesting a new  
762 *Aggregate Business Information Entity* that re-uses an existing *Aggregate Core*  
763 *Component*. These two steps are:

764 **Figure 5-8 Preparation Steps for Requesting a New ABIE using Existing ACC**



766 Step 1. Apply the *Naming Convention* and *Rules* to arrive at the name of the new  
767 *Aggregate Business Information Entity*.

768 Step 2. Register the re-use of the existing *Aggregate Core Component* by this new  
769 *Aggregate Business Information Entity*.

## 770 5.4 Harmonisation

771 The purpose of harmonisation is to take a set of proposed *Core Components* or  
772 *Business Information Entities* from different domains, identify differences and  
773 similarities between the various submissions, and produce a single, complete cross-  
774 domain set. Harmonisation is a critical step in the overall *Core Component*  
775 procedures. The following describes the recommended areas that harmonisation  
776 procedures should cover.

- 777 • Evaluate each submitted *Core Component* for consistent application of the  
778 Discovery methodology. Resolve any questions or issues by discussion  
779 with the submitting groups.
- 780 • Compare the definition and structure of each submitted *Core Component*  
781 with what already exists in the *Core Component Library*.
  - 782 - If the submitted *Core Component* is the same or similar, compare the  
783 properties of each to identify any differences. If the submitted *Core*  
784 *Component* has properties missing in the existing one, enforce a  
785 harmonised form that contains the properties of each. If the submitted  
786 *Core Component* is a subset of the existing *Core Component*  
787 definition, then recommend the use of the existing one.
  - 788 - If the definition of the *Core Component* does not match any existing  
789 ones, then proceed.
- 790 • Publish the results of harmonisation to the submitting groups for review  
791 and finalisation.

792 [Note]

793 When submissions are received from different domains simultaneously, they are each  
794 processed in their own right against the full cross-domain library. The submissions are  
795 not compared against each other before comparison with existing library entries. In  
796 other words, all submissions are processed separately and serially against the full  
797 cross-domain library.

798 Once the submitted material has passed the harmonisation procedure, it may now be  
799 submitted for technical assessment and approval.

## 800 5.5 Technical Assessment and Approval

801 Technical assessment must be done in close coordination with the discovery teams  
802 and the harmonisation process in order to minimise domain re-working after technical  
803 assessment and harmonisation review. This section, 5.5, defines a recommended

804 process for conducting technical assessment and approval of all newly submitted and  
805 changed *Core Components*. A technical assessment and approval process for *Business*  
806 *Information Entities* should also be developed and applied.

807 Technical assessment procedures define the processing that shall be followed by the  
808 joint development groups, the harmonisation group, submission entry points, the  
809 technical assessment group, and the secretariat as related to the review of *Core*  
810 *Components*. The result of this process is the final publication of approved *Core*  
811 *Components*.

812 These procedures were developed in order to facilitate the process of reviewing and  
813 approving submissions to the *Core Component* library. In order to minimise the  
814 requirements for technical assessment and harmonisation, and to expedite the review  
815 and approval process, *Core Component* development groups should work with the  
816 technical assessment group, and the harmonisation group during the early  
817 development stages of component discovery.

818 In outline, the procedures cover:

- 819 1) Submission of *Core Component* work that is ready to be reviewed to a designated  
820 secretariat.
- 821 2) Recording of all *Core Component* submissions and distribution to the  
822 harmonisation group members.
- 823 3) Review procedures and criteria followed by the harmonisation group.
- 824 4) Return of harmonised *Core Component* submissions for technical assessment.
- 825 5) Review procedures and criteria followed by the technical assessment group.
- 826 6) Registration of the approved *Core Component(s)* in the appropriate *Core*  
827 *Component Registry*.

## 828 **5.6 Context in the Discovery Process**

829 Information that is needed by a business process is used in a context that is defined by  
830 how and where the business process can be used. The initial analysis will be  
831 performed on a set of *Business Information Entities*, i.e. both *Basic* and *Aggregate*  
832 *Business Information Entities*, and not on a set of *Core Components* (See Figure 5-1).  
833 The analysis that produces *Core Components* is, among other things, a process of  
834 identifying the various context categories and values, to determine those properties  
835 that exist in all possible contexts.

836 The guidelines presented here facilitate the analysis of *Business Information Entities*  
837 to determine core business semantics, or provide a mechanism to describe *Business*  
838 *Information Entities* when they are published in a repository.

839 When doing analysis, there is a key question: “Is a particular property of a *Business*  
840 *Information Entity* derived from its contextual business use, or is it a core property of  
841 the component?”

842 The answer to this question can be found by looking at as many different instances of  
843 that *Business Information Entity* as possible. If there is a single semantic property of  
844 that *Business Information Entity* that is found in every example available for analysis,  
845 then it can be assumed that the property in question is in fact a core semantic, and is  
846 not derived from the contextual business use.

847 If there are any instances of the *Business Information Entity* in which the property in  
848 question is not present, then this raises the issue of identity: Is the *Business*  
849 *Information Entity* which lacks that property really the same *Business Information*  
850 *Entity*, just used in a different context?

851 If the answer to this question is *yes*, then that property is not part of the *Core*  
852 *Component*, but is derived contextually, and the property should be removed from the  
853 *Basic Core Component* or *Aggregate Core Component* being discovered. If the  
854 answer is *no*, then it is possible that a second, different *Core Component* has been  
855 discovered.

## 856 **5.6.1 Context Categories**

857 Context categories are introduced here and are followed by a brief description. After  
858 which the various guidelines used to determine context are introduced:

- 859 • *Business Process Context*: This is the classification of the business  
860 process, business collaboration, or business transaction as described in the  
861 *Catalogue of Common Business Processes*. It is the primary context  
862 category, and provides many useful distinctions in the analysis of *Core*  
863 *Components*.
- 864 • *Product Classification Context*: There are many types of information that  
865 are specific to products or services being traded or referred to in a business  
866 process.
- 867 • *Industry Classification Context*: Traditionally, business vocabularies are  
868 divided up into industry verticals. This *Context Category* specifies a  
869 particular industry vertical.
- 870 • *Geopolitical Context*: Specifies the semantic and structural variation. This  
871 is often the result of regional or cultural factors.
- 872 • *Official Constraints Context*: Specifies the legal or contractual influences  
873 upon business semantics.
- 874 • *Business Process Role Context*: Every partner in a business process data  
875 exchange has a particular role – buyer, seller, etc. These roles are  
876 described in the *Catalogue of Common Business Processes* and in other



877 *Business Libraries* (libraries of business process models). Depending on  
878 the business process, the nature of these roles may require that certain  
879 semantics and data be employed in the messages exchanged. In any  
880 *Business Process Role Context*, one must either be a sender or receiver of  
881 data in that particular exchange – otherwise, role is described by the  
882 *Supporting Role Context*.

883 • *Supporting Role Context*: Parties in a business process who are neither  
884 senders nor receivers of data in a particular exchange, may place  
885 requirements on the data exchanged by partners who are sending or  
886 receiving of data in that exchange. These non-sending, non-receiving  
887 parties in this exchange play a supporting role, and are described by the  
888 *Supporting Role Context*.

889 • *System Capabilities Context*: When a particular semantic or structure is  
890 primarily the result of system constraints, or compliance with a standard,  
891 then it is attributable to the *System Capabilities Context*.

## 892 **5.6.2 Guidelines for Analyzing Business Information Entities in Context**

893 Using the criteria given in section 5.6.1 for determining that a particular property of a  
894 *Business Information Entity* is in fact the product of its use in context, the analyst  
895 must ascertain and document the applicable context categories. To accomplish this,  
896 the analyst should list all the context categories, and assign a value or values to each  
897 category for that component. If a context category has no particular value or values,  
898 then the analyst should assign a value of *In All Contexts* (for all contexts except  
899 *Official Constraints*) or *None* (for *Official Constraints*). As this analysis is conducted,  
900 different context categories might appear to be in competition for application. The  
901 analyst must ascertain which context category is responsible. This section provides  
902 some guidelines for answering this question in a systematic and consistent fashion, by  
903 examining the typical ambiguities that arise.

904 It is possible that a particular property of a *Business Information Entity* may be the  
905 result of several context factors. These context factors are identified by analysis of  
906 differences and similarities across particular contexts. For example, comparing the  
907 same *Business Information Entity* as used in different regions of the world, variation  
908 will probably be the result of a geopolitical context or official constraints context (see  
909 below). If a single *Business Information Entity* differs between business processes,  
910 then the business process context is probably the cause. For each non-core property of  
911 every *Business Information Entity* analysed the relevant influences and hence context  
912 factors should be identified.

913 The following guidelines apply:

### 914 1) *Geopolitical Context* versus *Official Constraints Context*

915 If a property can be traced to a specific body of law or international treaty then it is  
916 the result of an official constraint. For example, if a warning about hazardous

917 goods is required as part of a goods description, and it is required on all uses of that  
918 goods description within the United States, then both *Geopolitical* and *Official*  
919 *Constraints* are involved. The value of an *Official Constraint Context* should  
920 always be the body of law or treaty that is being cited. The value of a *Geopolitical*  
921 *Context* always expresses the region or regions that are relevant.

922 2) *Product Classification Context* versus *Industry Classification Context*

923 When a particular variation on a given product or service is specific to a particular  
924 industry, then the *Industry Classification Context* is adequate to specify the  
925 context. If all examples of the particular product or service are described by the  
926 same unique set of properties across industries, then only a *Product Classification*  
927 *Context* is required. In other cases, a value or values should be supplied for both  
928 context categories.

929 3) *Business Process Context* versus *Business Role Context*

930 *Business Role Context* is employed when one actor in the business process has an  
931 information requirement and the other does not. If both actors have the same  
932 information requirement, then it is a *Business Process Context*.

933 4) *System Capability Context* categories

934 This context is the result of system or classes of systems that *primarily* influence  
935 data variation. For example, if a specific Enterprise Resource Planning (ERP)  
936 provider's proprietary data formats use a particular field, and no other applications  
937 use that field, then the presence of the data can be attributed to the processing  
938 capabilities of that specific system.

939 The following detailed example illustrates the process of assigning values for all  
940 context categories as part of the *Business Information Entity* analysis process:

[Example]

Case: A buyer address *Business Information Entity* is taken from a standard that is used across all industry boundaries and in all processes within the United States. The *Business Information Entity* also contains a child field that holds the *State* information.

The following set of values could be ascribed to this child field for this *Business Information Entity*:

*Business Process* = *In All Contexts*

*Product Classification* = *In All Context*

*Industry Classification* = *In All Contexts*

*Geopolitical* = *United States*

*Official Constraint* = *None*

*Business Process Role* = *In All Contexts*

*Supporting Role* = *In All Contexts*

*System Capabilities* = *In All Contexts*

These values were selected based on the following analysis:

The *Business Information Entity* construct is the same in every business process covered by the standard in question – the address always contains a *State* field. Therefore, for the range of business processes covered by the *Business Information Entity* being analysed, – the *Business Process Context* category is marked *In All Contexts*.

The products that might be described in the same business message do not affect the address. Since the standard from which the *Business Information Entity* has been extracted is horizontal across industry boundaries, it is equally valid in all *Industry Classification Contexts*.

As a child element of *Buyer Address*, it is clear that the *State* field is intended to hold a value specific to United States geopolitical demarcations. Therefore the *Geopolitical Context Category* is properly assigned the value *United States*.

No specific law can be cited that requires the presence of the *State* field in the address. Therefore, a value of *None* is given to the *Official Constraint Context Category*.

On inspection of *Business Process Role*, it appears that all addresses in the standard in question are required to provide the *State* information, regardless of what role they play in the transaction. The fact that a *Buyer Role* is being analysed has no effect on this field: all types of addresses have the same semantics. Therefore, all roles provide the data equally when giving an address. A value of *In All Contexts* is applicable here. The same reasoning holds for the *Supporting Role Context*.

Finally, considering the *System Capabilities Context*, there are no specific systems that act as the primary reason for the presence or absence of the semantic. Instead, the primary existence of the field can be ascribed to the fact that in common usage, US addresses include the *State* field. Therefore, we can provide the value *In All Contexts* here. Note that as wide of a range of values as possible should be provided to ensure completeness.

984 If, in the above example, the address was taken from a French standard, it might be  
985 that some child elements are common across a number of countries in the same  
986 region, and perhaps even in multiple regions. Providing the value *France* as a  
987 *Geopolitical Context* here would be incomplete – every known valid value should be  
988 given.

## 989    **6   Technical Details**

990    This section provides a detailed technical explanation of the *Core Component*,  
991    *Business Process* integration, storage and metamodel elements of the UN/CEFACT  
992    *Core Components* concept.

### 993    **6.1   Core Components and Business Information Entities**

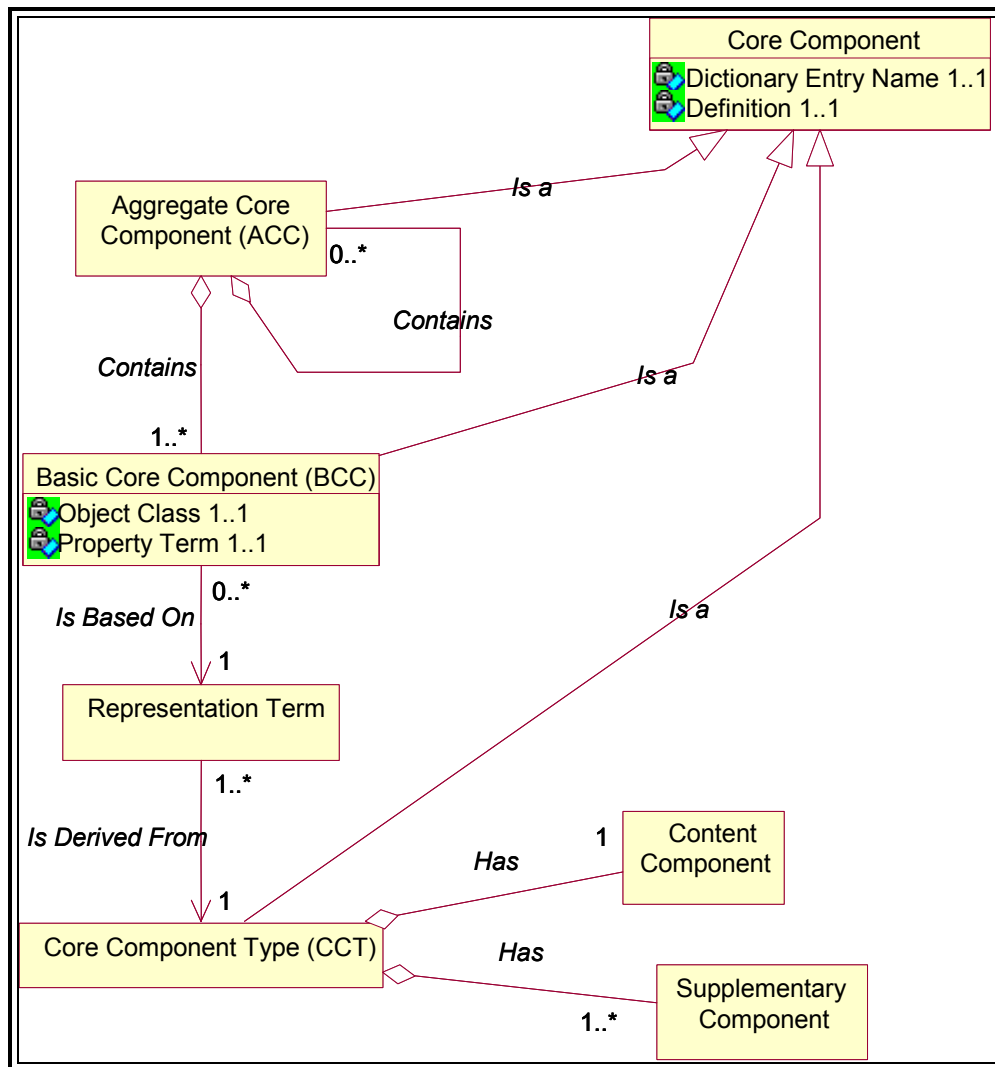
994    This section defines the following:

- 995            • *Core Component* rules
- 996            • *Naming Conventions*,
- 997            • Allowable *Core Component Types*
- 998            • *Content* and *Supplementary Component* types, and
- 999            • *Representation Terms*.

1000    This section also specifies relationships for *Core Components* and *Business*  
1001    *Information Entities* and includes details required for constructing the *Core*  
1002    *Components Catalogue* and a larger *Core Component Library*.

#### 1003    **6.1.1   Core Components**

1004    A *Core Component* is a building block for the creation of a semantically correct and  
1005    meaningful business information exchange ‘parcel’, containing the information pieces  
1006    needed to describe a specific concept. There are three categories of *Core Components*:  
1007    *Basic Core Component*, *Core Component Type* and *Aggregate Core Component*.  
1008    Figure 6-1 illustrates these three categories and their relationships.

1009 **Figure 6-1. Core Components Metamodel**

1010 The following general rules must be followed in discovering and documenting the  
 1011 three types of *Core Components*:

1012 [C1] Each *Core Component Type*, *Basic* or *Aggregate Core Component* must have  
 1013 its own business semantic definition. The definition shall be developed first  
 1014 and the *Dictionary Entry Name* shall be extracted from it. Remarks can be  
 1015 used to further clarify the definition, to provide examples and/or to reference a  
 1016 recognised standard.

1017 [C2] Within an *Aggregate Core Component*, all embedded entities shall be related  
 1018 to the concept of the aggregate.

- 1019 [C3] There shall be no semantic overlap between the *Core Components* embedded  
1020 within the same *Aggregate Core Component*.
- 1021 [C4] The representation of the information in a *Core Component* of the *Core*  
1022 *Component Type Code* should use a standard issued by a recognised standards  
1023 body, whenever a standard exists. If international standards are not used a  
1024 business driven justification shall be provided.
- 1025 [C5] An *Aggregate Core Component* must contain at least one *Basic Core*  
1026 *Component*.

1027 [Note]

1028 The issue of allowing aggregates without at least one *Basic Core Component* has been  
1029 discussed in detail. Modelling has shown that aggregates that contain only other  
1030 aggregates result in models where the outer bracket has no meaning on its own. The  
1031 aggregated aggregates in their basic form provide the same benefits as using the basic  
1032 aggregates on their own.

1033

1034 [Note]

1035 For the purpose of exchanging information a practical compromise on the level of  
1036 detail of a *Basic Core Component* is required. This compromise shall be based on the  
1037 business need. It is not necessary to have absolute detail, which decomposes a piece  
1038 of information down to its lowest level.

- 1039 [C6] The *Core Component Type* shall be one of the approved *Core Component*  
1040 *Types*

1041 [Note]

1042 Table 8-1 provides a complete list of the currently approved *Core Component Types*.  
1043 This Table will subsequently be published separately to facilitate maintenance outside  
1044 the body of this specification.

- 1045 [C7] The *Content Component* shall be one of the approved *Content Components*

- 1046 [C8] The *Supplementary Component* shall be one of the approved *Supplementary*  
1047 *Components*

1048

[Note]

1049

1050

1051

Table 8-2 provides a complete list of the currently approved *Content Components* and *Supplementary Components*. This Table will subsequently be published separately to facilitate maintenance outside the body of this specification.

1052

1053

### 6.1.2 Business Information Entities

1054

1055

1056

1057

1058

1059

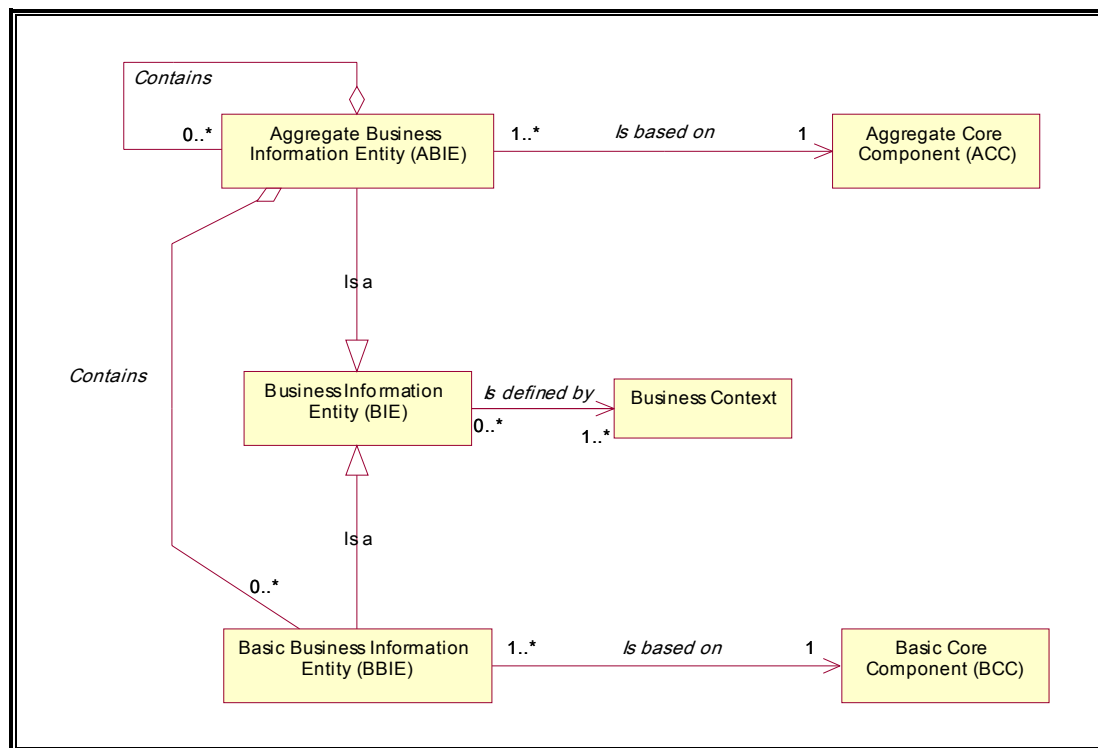
1060

1061

A *Business Information Entity* is a piece of business data or a group of pieces of business data with a unique business semantic definition. A *Business Information Entity* can be either a *Basic Business Information Entity* (BBIE) or an *Aggregate Business Information Entity* (ABIE). A *Basic Business Information Entity* is based on a *Basic Core Component* (BCC). An *Aggregate Business Information Entity* is a re-use of an *Aggregate Core Component* (ACC) in a specified business context. Figure 6-2 describes the *Business Information Entity* types and shows relationships to the *Core Component* counterparts.

1062

**Figure 6-2. Business Information Entities Basic Definition Model**



1063

1064

1065

1066

1067

1068

1069

- [B1] A *Business Information Entity* shall be either a *Basic Business Information Entity* or an *Aggregate Business Information Entity*.
- [B2] A *Business Information Entity* shall be defined by one or more *Business Contexts*.
- [B3] A *Basic Business Information Entity* shall be based on a *Basic Core Component*.



1070 [B4] An *Aggregate Business Information Entity* shall be based on an *Aggregate*  
1071 *Core Component*.

1072 [B5] An *Aggregate Business Information Entity* shall consist of two or more *Basic*  
1073 *Business Information Entities* and/or *Aggregate Business Information Entities*.

### 1074 **6.1.3 Naming Convention**

1075 A naming convention is necessary to gain consistency in the naming and defining of  
1076 all *Core Components* and *Business Information Entities*. The resulting consistency  
1077 facilitates comparison during the discovery and analysis process, and precludes  
1078 ambiguity, such as the creation of multiple *Core Components* with different names  
1079 that have the same semantic meaning.

1080 The *Naming Convention* is derived from the guidelines and principles described in  
1081 document ISO 11179 Part 5 -- *Naming and Identification Principles For Data*  
1082 *Elements*. In certain instances, these guidelines have been adapted to the *Core*  
1083 *Component* environment. In particular, the guidelines have been extended to cover the  
1084 naming and defining of *Core Component Types* and *Business Information Entities*.

1085 In order to ensure absolute clarity and understanding of the names and definitions it is  
1086 essential to use words from the *Oxford English Dictionary*. A supplementary  
1087 *Controlled Vocabulary* will be developed to define uniquely any words that are  
1088 potentially ambiguous. This will ensure that each word within any of the names and  
1089 definitions is used in a consistent and unambiguous way. The resultant semantic  
1090 integrity will also mean that translation into other languages retains the precise  
1091 original meaning.

#### 1092 **6.1.3.1 Core Component Naming Rules**

1093 The following subsections contain all *Core Component* naming rules.

##### 1094 **6.1.3.1.1 Core Component Dictionary Information**

1095 Each *Core Component* contains the following dictionary information that is impacted  
1096 by the naming rules in subsequent sub-sections:

- 1097 • **Dictionary Entry Name** (Mandatory). This is the unique official name of  
1098 the *Core Component* in the dictionary.
- 1099 • **Definition** (Mandatory). This is the unique semantic business meaning of  
1100 that *Core Component*.
- 1101 • **Business Term** (Optional). This is a synonym term under which the *Core*  
1102 *Component* is commonly known and used in the business. A *Core*  
1103 *Component* may have several business terms or synonyms.

1104	[Example]
1105	<i>Dictionary Entry Name – Person. Tax. Identifier</i>
1106	Definition – The registered national tax identification of a person
1107	<i>Business Term – Income tax number, national register number, personal tax register</i>
1108	<i>number, social security number, national insurance number</i>

1109 The naming rules are also based on the following concepts as defined in ISO 11179:

- 1110 • **Object Class.** This represents the logical data grouping or aggregation (in a
- 1111 logical data model) to which a data element belongs. The *Object Class*
- 1112 thus is the part of a *Core Component's Dictionary Entry Name* that
- 1113 represents an activity or object in a specific context.
- 1114 • **Property Term.** This represents the distinguishing characteristic or
- 1115 property of the dominant area of interest and shall occur naturally in the
- 1116 definition.
- 1117 • **Representation Term.** This defines the type of valid values for an
- 1118 information entity.

#### 1119 6.1.3.1.2 Core Component General Rules

1120 [C9] The dictionary content shall be in *English Language* following the

1121 primary *Oxford Dictionary* English spellings to assure unambiguous

1122 spelling.

1123	[Note]
1124	There may be restrictions in specific languages, which need to be applied when
1125	transforming the <i>Core Component</i> dictionary into other languages. These restrictions
1126	shall be formulated as additional rules and added as separate language specific
1127	annexes to this document.

#### 1128 6.1.3.1.3 Core Component Rules for Definitions

1129 [C10] The definition shall be consistent with the requirements of ISO 11179-4

1130 Section 4.4 and will provide an understandable meaning, which should also be

1131 translatable to other languages.

1132 [C11] The definition shall take into account the fact that the users of the *Core*

1133 *Component Dictionary* are not necessarily native English speakers. It shall

1134 therefore contain short sentences, using normal words. Wherever synonym

1135 terms are possible, the definition shall use the preferred term as identified in  
1136 the ebXML *Core Components Glossary of Terms*.

1137 [C12] The definition of a *Basic Core Component* shall use a structure that is based  
1138 on the existence of the *Object Class*, the *Property Term*, and its  
1139 *Representation Term*.

1140 [C13] Whenever both the definite (i.e. *the* ) and indefinite article (i.e. *a* ) are  
1141 possible in a definition, preference shall be given to the indefinite article (i.e.  
1142 *a* ).

1143 [Note]

1144 To check the quality of the definition, place the *Dictionary Entry Name* followed by *is*  
1145 before the definition to ensure that it is not simply a repetition of the *Dictionary Entry*  
1146 *Name*.

#### 1147 6.1.3.1.4 Core Component Rules for Dictionary Entry Names

1148 [C14] The *Dictionary Entry Name* shall be unique.

1149 [C15] The *Dictionary Entry Name* shall be extracted from the *Core Component*  
1150 definition.

1151 [C16] The *Dictionary Entry Name* shall be concise and shall not contain consecutive  
1152 redundant words.

1153 [C17] The *Dictionary Entry Name* and all its components shall be in singular form  
1154 unless the concept itself is plural.

1155 [Example]

1156 The singular *Good* does not exist, whereas *Goods* is a plural noun whose concept  
1157 involves one or multiple (plural) items

1158 [C18] The *Dictionary Entry Name* shall not use non-letter characters unless required  
1159 by language rules.

1160 [C19] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e.  
1161 no words like *and*, *of*, *the*, etc.). This rule shall be applied to the English  
1162 language, and may be applied to other languages as appropriate.

1163 [C20] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall  
1164 be expanded or explained in the definition.

1165 [C21] The *Dictionary Entry Name* of a *Basic Core Component* shall consist of the  
1166 name of an *Object Class*, the name of a *Property Term* and the name of a  
1167 *Representation Term*.

1168 [Example]

1169 *Tax. Description. Text*

1170 [C22] The components of a *Dictionary Entry Name* shall be separated by dots. The  
1171 space character shall separate words in multi-word *Object Classes* and/or  
1172 multiword *Property Terms*. Every word shall start with a capital letter. To  
1173 allow spell checking of the *Dictionary Entry Names*' words, the dots after  
1174 *Object Class* and *Property Terms* shall be followed by a space character.

1175 [Note]

1176 The use of CamelCase for *Dictionary Entry Names* has been considered, but has been  
1177 rejected for following reasons:

- 1178 ♦ Use of CamelCase will not allow the use of spell checkers
- 1179 ♦ Strict use of CamelCase makes it impossible to use separators (“.”) and  
1180 therefore doesn't allow an unambiguous identification of the composing  
1181 parts of the *Dictionary Entry Name*.

1182 [C23] The name of an *Object Class* shall be unique throughout the dictionary and  
1183 may consist of more than one word.

1184 [C24] The name of a *Property Term* shall occur naturally in the definition and may  
1185 consist of more than one word. A name of a *Property Term* shall be unique  
1186 within the context of an *Object Class* but may be reused across different  
1187 *Object Classes*.

1188 [Example]

1189 *Car. Colour. Code* and *Shirt. Colour. Code* may both exist.

1190 [C25] If the name of the *Property Term* uses the same word as the *Representation*  
1191 *Term* (or an equivalent word), this *Property Term* shall be removed from  
1192 *Dictionary Entry Name*. The *Representation Term* word in this case only will  
1193 remain.

1194 [Example]  
 1195 If the *Object Class* is *Goods*, the *Property Term* is *Delivery Date*, and *Representation*  
 1196 *Term* is *Date*, the *Dictionary Entry Name* is *Goods. Delivery. Date*; the *Dictionary*  
 1197 *Entry Name* for an identifier of a party (*Party. Identification. Identifier*) will be  
 1198 truncated to *Party. Identifier*.

1199 [C26] The name of the *Representation Term* shall be one of the terms specified in the  
 1200 *List of Representation Terms* as included in this document (See section  
 1201 6.1.3.3).

1202 [C27] The name of the *Representation Term* shall not be truncated in the *Dictionary*  
 1203 *Entry Name*.

1204 [C28] The *Dictionary Entry Name* of a *Core Component Type* shall consist of a  
 1205 meaningful type name followed by a dot, a space character, and the term *Type*.

1206 [Example]  
 1207 *Amount. Type; Date Time. Type*

1208 [C29] The *Dictionary Entry Name* of an *Aggregate Core Component* shall consist of  
 1209 a meaningful *Object Class* followed by a dot, a space character, and the term  
 1210 *Details*. The *Object Class* may consist of more than one word.

1211 [Example]  
 1212 *Postal Address. Details; Party. Details*

#### 1213 6.1.3.1.5 Rules for Core Component Business Terms

1214 *Core Component Business Terms* are those terms that are commonly used for day-to-  
 1215 day information exchanges within a given domain. As such, no specific naming rules  
 1216 apply to *Business Terms*. Interoperability of *Business Terms* will be given by linking  
 1217 them to *Core Component* dictionary entries.

#### 1218 6.1.3.2 Rules for Business Information Entities

1219 The following subsections contain the naming rules for *Business Information Entities*.

##### 1220 6.1.3.2.1 Business Information Entity Dictionary Information

1221 Each *Business Information Entity* contains the following dictionary information that is  
 1222 impacted by the naming rules:

- 1223 • **Dictionary Entry Name** (Mandatory). This is the unique official name of  
1224 the *Business Information Entity* in the dictionary.
- 1225 • **Definition** (Mandatory). This is the unique semantic business meaning of  
1226 that *Business Information Entity*.
- 1227 • **Business Term** (Optional). This is a synonym term under which the  
1228 *Business Information Entity* is commonly known and used in the business  
1229 for a specific context. A *Business Information Entity* may have several  
1230 business terms or synonyms.

1231 The *Business Information Entity* naming rules are also based on the following  
1232 concepts as defined in ISO 11179:

- 1233 • **Object Class**. This represents the logical data grouping or aggregation (in a  
1234 logical data model) to which a data element belongs. The *Object Class*  
1235 thus is the part of a *Business Information Entity's Dictionary Entry Name*  
1236 that represents an activity or object in a specific context.
- 1237 • **Property Term**. This represents the distinguishing characteristic or  
1238 property of the dominant area of interest and shall occur naturally in the  
1239 definition.
- 1240 • **Representation Term**. This defines the type of valid values for an  
1241 information entity.
- 1242 • **Qualifier Term**. A word or words which help define and differentiate a  
1243 *Business Information Entity* from its associated core component and other  
1244 *Business Information Entities*. *Qualifier Terms* may be derived from  
1245 *Structure Sets* specific to a context.

#### 1246 6.1.3.2.2 Business Information Entity General Rules

- 1247 [B6] The dictionary content shall be in English Language following the primary  
1248 Oxford Dictionary English spellings to assure unambiguous spelling.

#### 1249 6.1.3.2.3 Business Information Entity Rules for Definitions

- 1250 [B7] The definition shall be consistent with the requirements of ISO 11179-4  
1251 Section 4.4 and will provide an understandable meaning, which should also be  
1252 translatable to other languages.
- 1253 [B8] The definition shall take into account the fact that the users of the *Business*  
1254 *Information Entity Dictionary* are not necessarily native English speakers. It  
1255 shall therefore contain short sentences, using normal words. Wherever  
1256 synonym terms are possible, the definition shall use the preferred term as  
1257 identified in the *Business Information Entity Glossary of Terms*.

1258 [B9] The definition of a *Basic Business Information Entity* shall use a structure that  
1259 is based on the existence of the *Object Class*, the *Property Term*, and its  
1260 *Representation Term*, and enhanced by business related *Qualifier Terms*.

1261 [B10] Whenever both the definite (i.e. *the* ) and indefinite article (i.e. *a* ) are  
1262 possible in a definition, preference shall be given to the indefinite article (i.e.  
1263 *a* ).

#### 1264 6.1.3.2.4 Rules for Business Information Entity Dictionary Entry Names

1265 [B11] The *Dictionary Entry Name* shall be unique.

1266 [B12] The *Dictionary Entry Name* shall be extracted from the *Business Information*  
1267 *Entity* definition.

1268 [B13] The *Dictionary Entry Name* shall be concise and shall not contain consecutive  
1269 redundant words.

1270 [B14] The *Dictionary Entry Name* and all its components shall be in singular form  
1271 unless the concept itself is plural.

1272 [B15] The *Dictionary Entry Name* shall not use non-letter characters unless required  
1273 by language rules.

1274 [B16] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e.  
1275 no words like *and*, *of*, *the*, etc.). This rule shall be applied to the English  
1276 language, and may be applied to other languages as appropriate.

1277 [B17] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall  
1278 be expanded or explained in the definition.

1279 [B18] The *Dictionary Entry Name* of a *Basic Business Information Entity* shall  
1280 consist of the names of the *Object Class*, *Property Term*, and *Representation*  
1281 *Term* of its associated *Core Component*, and additional *Qualifier Term(s)*.

1282 [B19] *Qualifier Terms* shall be applied to the *Object Class* and/or the *Property Term*.

1283 [B20] *Qualifier Terms* shall not be applied to the *Representation Term*.

1284 [B21] The components of a *Dictionary Entry Name* shall be separated by dots. The  
1285 space character shall separate words in multi-word *Object Classes* and/or  
1286 multiword *Property Terms*, including their *Qualifier Terms*. Every word shall  
1287 start with a capital letter. *Qualifier Terms* shall not be separated from their  
1288 associated *Object Class* or *Property Term*. To allow spell checking of the  
1289 words in the *Dictionary Entry Name*, a space character shall follow the dots  
1290 after *Object Class* and *Property Term(s)*.

- 1291 [B22] *Qualifier Terms* shall precede the associated *Object Class* or *Property Term*.  
 1292 The order of qualifiers shall not be used to differentiate *Dictionary Entry*  
 1293 *Names*.

1294 [Example]

1295 In the *Business Information Entity* entitled *Cost. Budget Period Total. Amount*, the  
 1296 component *Budget Period* is a *Qualifier Term* for the *Property Term* of *Total*. This is  
 1297 derived from the *Core Component* of *Cost. Total. Amount*.

- 1298 [B23] The name of a qualified *Object Class* refers to an activity or object within a  
 1299 *Business Context*. It shall be unique throughout the dictionary and may consist  
 1300 of more than one word.

- 1301 [B24] If the name of the *Property Term* uses the same word as the *Representation*  
 1302 *Term* (or an equivalent word), and the *Property Term* is not qualified, the  
 1303 *Property Term* shall be removed from *Dictionary Entry Name*.

- 1304 [B25] The *Dictionary Entry Name* of an *Aggregate Business Information Entity* shall  
 1305 consist of the name of the *Object Class* of its associated *Aggregate Core*  
 1306 *Component* and additional *Qualifier Term(s)* to represent its specific *Business*  
 1307 *Context*, followed by a dot, a space character, and the term *Details*.

#### 1308 6.1.3.2.5 Rules for Business Information Entity Business Terms

1309 *Business Information Entity Business Terms* are those terms that are commonly used  
 1310 for day-to-day information exchanges within a given domain. As such, no specific  
 1311 naming rules apply to *Business Terms*. Interoperability of *Business Terms* will be  
 1312 given by linking them to the *Business Information Entity* dictionary entries.

#### 1313 6.1.3.3 List of Representation Terms

1314 The *Representation Term* is the part of a *Core Component* name that describes the  
 1315 form of valid values in which the business information is expressed in a data item. For  
 1316 instance all basic *Core Components* representing a monetary amount shall be named  
 1317 *[Name]. Amount* where *[Name]* represents a specialisation of the generic amount and  
 1318 *Amount* is the *Representation Term*. Table 6-1 lists the permissible *Representation*  
 1319 *Terms*.

- 1320 [C30] The *Representation Term* shall be one out of the list of permissible  
 1321 *Representation Terms*



1322 **Table 6-1 Permissible Representation Terms**

<b>Representation Term</b>	<b>Definition</b>	<b>Links to Core Component Type</b>
<b>Amount</b>	A number of monetary units specified in a currency where the unit of currency is explicit or implied.	Amount. Type
<b>Code</b>	A character string (letters, figures or symbols) that for brevity and / or language independence may be used to represent or replace a definitive value or text of an attribute. Codes usually are maintained in code lists per attribute type (e.g. colour).	Code. Type
<b>Date</b>	A day within a particular calendar year (ISO 8601).	Date Time. Type
<b>Date Time</b>	A particular point in the progression of time (ISO 8601).	Date Time. Type
<b>Graphic</b>	A diagram, graph, mathematical curves, or similar representation	Graphic. Type
<b>Identifier</b>	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme.  [Note: Type shall not be used when a person or an object is identified by its name. In this case the <i>Representation Term</i> “Name” shall be used.]	Identifier. Type
<b>Indicator</b>	A list of two, and only two, values that indicate a condition such as on/off; true/false etc. (synonym: “Boolean”).	Indicator. Type
<b>Measure</b>	A numeric value determined by measuring an object. Measures are specified with a unit of measure. The applicable unit of measure is taken from UN/ECE Rec. 20.	Measure. Type
<b>Name</b>	A word or phrase that constitutes the distinctive designation of a person, place, thing or concept.	Text. Type
<b>Percent</b>	A rate expressed in hundredths between two values that have the same unit of measure.	Numeric. Type
<b>Picture</b>	A visual representation of a person, object, or scene	Picture. Type
<b>Quantity</b>	A number of non-monetary units. It is associated with the indication of objects. Quantities need to be specified with a unit of quantity.	Quantity. Type
<b>Rate</b>	A quantity or amount measured with respect to another measured quantity or amount, or a fixed	Numeric. Type

<b>Representation Term</b>	<b>Definition</b>	<b>Links to Core Component Type</b>
	or appropriate charge, cost or value e.g. US Dollars per hour, US Dollars per Euro, kilometre per litre, etc.	
<b>Text</b>	A character string generally in the form of words of a language.	Text. Type
<b>Time</b>	The time within a (not specified) day (ISO 8601).	Date Time. Type
<b>Value</b>	Numeric information that is assigned or is determined by calculation, counting or sequencing. It does not require a unit of quantity or a unit of measure	Numeric. Type

1323 In addition to permissible representation terms for *Core Components*, there are also  
 1324 permissible representation terms for *Aggregate Core Components* and *Core*  
 1325 *Component Types*. Table 6-2 contains the permissible representation terms that apply  
 1326 to *Aggregate Core Components* or *Core Component Types*.

1327 [C31] The *Representation Term* for *Aggregate Core Components* or *Core*  
 1328 *Component Types* shall be one of the list of permissible *Aggregate Core*  
 1329 *components* or *Core Component Type Representation Terms*

1330 **Table 6-2 Permissible Representation Terms for Aggregate Core Components or**  
 1331 **Core Component Types**

<b>Representation Term</b>	<b>Definition</b>	<b>Links to Core Component Type</b>
<b>Details</b>	The expression of the aggregation of <i>Core Components</i> to indicate higher levelled information entities	Not Applicable
<b>Type</b>	The expression of the aggregation of <i>Core Components</i> to indicate the aggregation of lower levelled information entities to become <i>Core Component Types</i> . All <i>Core Component Types</i> shall use this <i>Representation Term</i>	Not Applicable
<b>Content</b>	The actual content of an information entity. <i>Content</i> is the first information entity in a <i>Core Component Type</i>	Used with the <i>Content Components</i> of <i>Core Component Types</i>

#### 1332 6.1.4 Catalogue of Core Components

1333 As originally articulated in the ebXML architecture concept and perpetuated in the  
 1334 developing UN/CEFACT architecture concept, all *Core Components* will be recorded  
 1335 in an ebXML compliant registry and stored in a related repository. However, small

1336 and medium enterprise (SME) organisations may not be able to readily access such  
 1337 architecture. As such, it is important that the full range of *UN/CEFACT Core*  
 1338 *Components* be published in a freely available catalogue. This catalogue must convey  
 1339 the full details of each *Core Component* consistent with how those components are  
 1340 stored as UML objects in the repository. Table 6-3 identifies a proper format for the  
 1341 catalogue and contains representative entries from the existing *UN/CEFACT Core*  
 1342 *Components Catalogue*.

1343 **Table 6-3. Core Component Catalogue**

UID	Dictionary Entry Name	CCT Used	Basic or Aggregate	Definition	Remarks	Object Class	Property Term	Representation Term	Business Terms	Core Component Children
000024	Address. Type. Code	Code. Type	Basic	The type of the address.	For example a business address or a home address. Not the Role of the address.	Address	Type	Code		
000147	Base Charge Price. Quantity	Quantity. Type	Basic	The base quantity of the charge/price unit amount.	For example, for a charge of \$5/day for 10 days, the charge base quantity is 1 day.	Base Charge Price	Quantity*	Quantity		
000139	Base. Currency. Code	Code. Type	Basic	The currency that is on the 'one unit' side of the rate of exchange.	The base currency amount divided by the currency exchange rate gives the second currency amount.	Base	Currency	Code		
000012	Birth. Date	Date Time. Type	Basic	The date on which a person was born.	Applies only to parties being natural persons.	Birth	Date*	Date		

1344 The catalogue is intended to be part of a larger *Core Component Library*. The *Core*  
 1345 *Component Library* will consist of the following parts:

- 1346 • *Core Component Types*

- 1347       • *Core Component Catalogue*, including *Basic Core Components*, *Basic*  
1348       *Business Information Entities*, and *Aggregate Core Components*
- 1349       • *Catalogue of Aggregate Business Information Entities*

### 1350   **6.1.5 Catalogue of Business Information Entities**

1351   For the same reasons that a *Core Components Catalogue* is necessary, a *Catalogue of*  
1352   *Business Information Entities* is also required. Predefined BIEs are not provided.  
1353   Rather, the working registries and the groups defining business messages will be  
1354   responsible for developing a *Catalogue of Business Information Entities*.

## 1355   **6.2 Context**

1356   This section fully describes applicable rules and applications for the use of context in  
1357   *Core Component* discovery, analysis, and use to include context categories and their  
1358   values, and the *Constraint Language*.

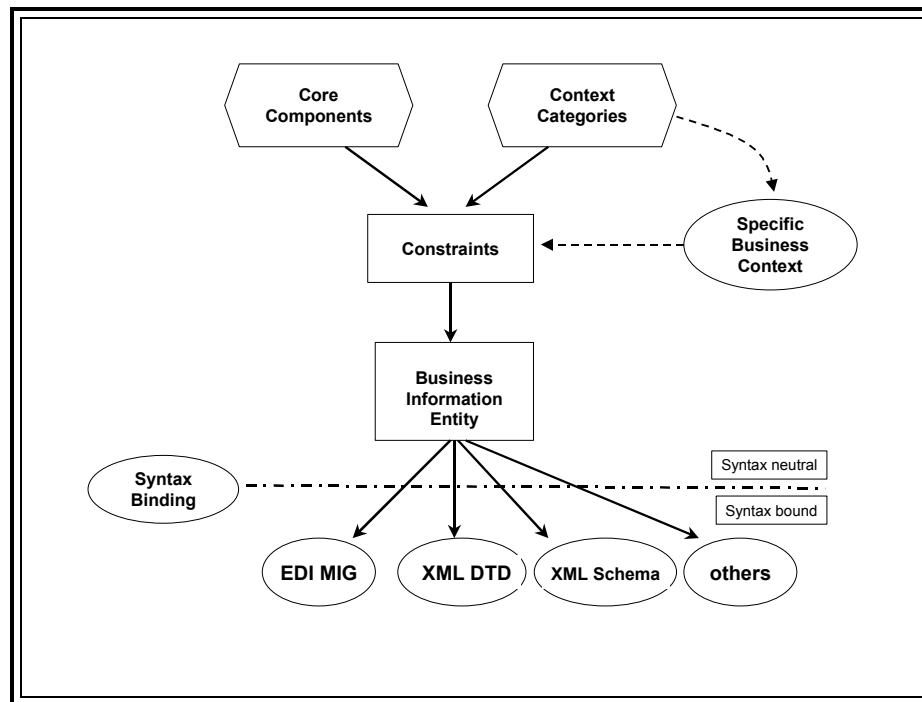
### 1359   **6.2.1 Overview of Context Specification**

1360   Whenever business collaboration takes place between specific trading partners, data is  
1361   exchanged in the form of business messages. That data exists in a particular business  
1362   context. In its simplest form, this is the idea of *context* as used in ebXML. The context  
1363   in which the business collaboration takes place can be specified by a set of categories  
1364   and their associated values.

1365   The *Core Components* have no context independent of their use. The *Context*  
1366   mechanism provides a full semantic qualification for the *Core Component* used in a  
1367   business process. Figure 6-3 shows how the constraint language applies *Business*  
1368   *Context* categories and specific business context(s) to *Core Components* to create  
1369   *Business Information Entities*. The *Business Information Entity* resulting from this  
1370   process can be manifested as a model, which in turn can be used as the basis of a  
1371   syntax-bound business message description (an EDI message implementation guide,  
1372   an XML schema, etc.)

1373   The following sections address the context categories, and the constraint language  
1374   more closely.

1375 **Figure 6-3. Operation of The Context Mechanism**



### 1377 6.2.1.1 Context Categories

1378 *Context Categories* exist to allow users to uniquely identify and distinguish between  
 1379 different business contexts. Eight context categories have been identified (Table 6-4).  
 1380 Each of the identified categories, unless otherwise stated, uses a standard  
 1381 classification to provide values for the category. Constraint rules, and therefore  
 1382 *Business Information Entities*, are tied to a particular set of standard classifications for  
 1383 identifying and distinguishing contexts.

### 1384 6.2.1.2 Constraint Language

1385 A *Constraint Language* is used to express the relationship between specific *Business*  
 1386 *Contexts* and how semantics are applied to the *Core Components* to produce *Business*  
 1387 *Information Entities*. The scope of this language covers two functional parts:

- 1388 • *Assembly* of a large aggregate (the *Document*);
- 1389 • Refinement of the assembly as appropriate. Refinement is both the  
 1390 addition of semantics specific to the business process, and the restriction  
 1391 and extension of the semantic model.

1392 This separation is a convenience for implementation (it simplifies the creation of  
 1393 processing tools) and creation of *standard* assemblies that can then be refined by  
 1394 specific users (a process that resembles how EDI standards and message  
 1395 implementation guides function today).

1396 Both *Constraint Language* parts allow, for example, simple commands indicating  
 1397 how *Core Components* will be used, how they will be named for these specific uses,  
 1398 and how to refine the cardinality (if necessary). Further, conditional relationships can  
 1399 be expressed. Specific context values or sets of values can be tied to the actions  
 1400 performed on *Core Components* to produce *Business Information Entities*.

1401 [Example]

1402 If the *Geopolitical Process Context* has a value of *Anywhere in the European Union*,  
 1403 and the specific *Business Context Value* indicates that the business process occurs in  
 1404 France, then the context-appropriate *Business Information Entity* can be assembled by  
 1405 modifying the correct *Core Component*.

1406 The constraint language would say—If the *Geopolitical Process Context* equals the  
 1407 *European Union*, then take the core *NameAddress* component and rules to provide the  
 1408 correct names, cardinality, and arrangement to the fields. To do business in France,  
 1409 the specific context value for that process will trigger this rule, giving a set of  
 1410 appropriate business semantics (*Business Information Entities*).

### 1411 6.2.1.3 Syntax Binding

1412 The *Business Information Entity* is a model that has no relationship to a specific  
 1413 syntax. It is intended that any given *Business Information Entity* can be expressed in  
 1414 any number of syntaxes. This process is called *syntax binding*, and is independent of  
 1415 (has no relationship to) a specific syntax. It may be possible to express *syntax binding*  
 1416 in an algorithm.

1417 The *Business Information Entity* in its standard form is a model that has no specific  
 1418 relationship to any given syntax. A given *Business Information Entity* can  
 1419 subsequently be expressed in any of a number of syntaxes through a binding process.  
 1420 The *Syntax Binding* process does not alter the semantics of the *Business Information*  
 1421 *Entity*, but simply instantiates the *Business Information Entity* for use in syntax  
 1422 specific documents.

1423 [B26] Syntax binding shall not change the semantics of a Business Information  
 1424 Entity.

### 1425 6.2.2 Approved Context Categories

1426 Table 6-4 contains the eight approved *Context Categories*.

1427 [C32] When describing a specific *Business Context*, a set of values will be assigned  
 1428 to the business situation being formally described.

1429 [C33] Applied *Business Context* will be from the list of approved context categories.

1430 **Table 6-4. Approved Context Categories**

Context Category	Description
Business Process	The business process as described using the ebXML Catalogue of Common Business Processes as extended by the user.
Product Classification	Factors influencing semantics that are the result of the goods or services being exchanged, handled, or paid for, etc. (e.g. the buying of consulting services as opposed to materials)
Industry Classification	Semantic influences related to the industry or industries of the trading partners (e.g., product identification schemes used in different industries).
Geopolitical	Geographical factors that influence business semantics (e.g., the structure of an address).
Official Constraints	Legal and governmental influences on semantics (e.g. hazardous materials information required by law when shipping goods).
Business Process Role	The actors conducting a particular business process, as identified in the Catalogue of Common Business Processes.
Supporting Role	Semantic influences related to non-partner roles (e.g., data required by a third-party shipper in an order response going from seller to buyer.)
System Capabilities	This context category exists to capture the limitations of systems (e.g. an existing back office can only support an address in a certain form).

1431 **6.2.2.1 Business Process Context**

1432 In describing a business situation, generally the most important aspect of that situation  
1433 is the business activity being conducted. *Business Process Context* provides a way to  
1434 unambiguously identify the business activity. To ensure consistency with business  
1435 process activities, it is important to use a common point of reference. The definitive  
1436 point of reference for international standards is the UN/CEFACT *Catalogue of*  
1437 *Common Business Processes*.

1438 [C34] Assigned *Business Process Contexts* shall be from the standard hierarchical  
1439 classification: provided as part of the UN/CEFACT *Catalogue of Common*  
1440 *Business Processes*.

1441 [C35] *Business Process Context* values may be expressed as a single business  
1442 process at any level, or may be expressed as a set of business processes at any  
1443 level.

1444 [C36] *Business Process Context* values may be taken from extensions to the business  
1445 processes described in the *Catalogue of Common Business Processes* as  
1446 provided for in that document.

1447 [C37] When business process extensions are used, they shall include full information  
1448 for each value sufficient to unambiguously identify which extension is  
1449 providing the value used.

#### 1450 6.2.2.2 Product Classification Context

1451 The *Product Classification Context* describes those aspects of a business situation  
1452 related to the goods or services being exchanged by, or otherwise manipulated, or  
1453 concerned, in the business process. Recognised code lists exist that provide  
1454 authoritative sources of product classification contexts.

1455 [C38] A single value or set of values may be used in a *Product Classification*  
1456 *Context*.

1457 [C39] If a hierarchical system of values is used for *Product Classification Context*,  
1458 then these values may be at any level of the hierarchy.

1459 [C40] If more than one classification system is being employed, an additional value  
1460 specifying which classification scheme has supplied the values used shall be  
1461 conveyed.

1462 [C41] Product classification context code values shall be taken from recognised code  
1463 lists to include:

- 1464 • *Universal Standard Product and Service Specification* (UNSPSC)
  - 1465 - Custodian: Electronic Commerce Code Management Association
  - 1466 (ECCMA)
- 1467 • *Standard International Trade Classification* (SITC Rev .3)
  - 1468 - Custodian: United Nations Statistics Division (UNSD)
- 1469 • *Harmonised Commodity Description and Coding System* (HS)
  - 1470 - Custodian: World Trade Organization (WTO)
- 1471 • *Classification Of the purposes of non Profit Institutions serving*  
1472 *households* (COPI)



1473 - Custodian: UNSD (This provides a mapping between the first three.)

#### 1474 6.2.2.3 Industry Classification Context

1475 The *Industry Classification Context* provides a description of the industry or sub-  
1476 industry in which the business process takes place.

1477 [C42] An *Industry Classification Context* may contain a single value or set of values  
1478 at any appropriate level of the value hierarchy.

1479 [C43] The *Industry Classification Context* value hierarchy must be identified.

1480 [C44] *Industry Classification Context* code values shall be taken from recognised  
1481 code lists to include:

1482 • *International Standard Industrial Classification* (ISIC)

1483 - Custodian: UNSD

1484 • *Universal Standard Product and Service Specification* (UNSPSC) Top-  
1485 level Segment [digits 1 and 2] used to define industry.

1486 - Custodian: ECCMA

1487 [Note]

1488 There are many other industry classification schemes that may be used for *Industry*  
1489 *Classification Context*.

#### 1490 6.2.2.4 Geopolitical Context

1491 *Geopolitical Contexts* allow description of those aspects of the business context that  
1492 are related to region, nationality, or geographically based cultural factors.

1493 [C45] *Geopolitical Context* shall consist of appropriate continent, economic region,  
1494 country, and region identifiers.

1495 [C46] *Geopolitical Regional Classification* may associate one or more values with  
1496 any business message or component.

1497 [C47] *Geopolitical Regional Classification* shall employ the following hierarchical  
1498 structure:

1499 Global  
1500 [Continent]  
1501 [Economic Region]  
1502 [Country] - ISO 3166.1  
1503 [Region] - ISO 3166.2

1504 [C48] At any level of the *Geopolitical Regional Classification* hierarchy, a value  
1505 may be a single value, a named aggregate, or cross-border value.

1506 [C49] *Geopolitical Regional Classification* hierarchy values shall structured as  
1507 follows:

- 1508 • **Single Value:** A single value indicating a single continent, economic  
1509 region, country, or region, depending on position within the hierarchy.
- 1510 • **Named Aggregate:** A related group of values (which may themselves be  
1511 single values, named aggregates, or cross-border pairs of values), which  
1512 have been related and assigned a name. A named aggregate contains at  
1513 least two values.
- 1514 • **Cross-Border:** One or more pairs of values, designated *To*, *From*, or *Bi-*  
1515 *directional*, indicating the direction of cross-border context. Values may  
1516 be named aggregates or single values.

1517 [Example]

1518 The following example shows an extract of the basic, single-value hierarchy of  
1519 recommended values, based on the common ISO 3166.1 *Country Codes*. (The value  
1520 at the top of any hierarchy is always understood to be *Global*.)

1521 Europe  
1522 Eastern Europe  
1523 AL – ALBANIA  
1524 AM – ARMENIA

1525 [C50] Points in the *Geopolitical Regional Classification* hierarchy shall be specified  
1526 by the use of the node value, or by the full or partial path.

1527 [C51] The full path of the *Geopolitical Regional Classification* hierarchy must be  
1528 used to understand the hierarchy when complex constructs are employed.

1529 [C52] A single-point specification is understood to inherit all of the properties of the  
1530 single-value hierarchy except where otherwise specified.

1531 [C53] *Geopolitical Values* will be taken from ISO 3166.1 and 3166.2

#### 1532 6.2.2.5 Official Constraints Context

1533 The *Official Constraints Context* category describes those aspects of the business  
1534 situation that result from legal or regulatory requirements and similar *official*  
1535 categories. This category contains two distinct parts:

1536 • Regulatory and Legislative. These are normally unilateral in nature and  
1537 include such things as customs.

1538 • Conventions and Treaties. These are normally bi- or multilateral  
1539 agreements and as such are different from regulatory and legislative  
1540 constraints.

1541 [C54] The *Official Constraints Context* will consist of at least two values:

1542 • Identification of the legal or other classification used to identify the  
1543 context values.

1544 • Identification of the official constraint itself. These values may represent a  
1545 hierarchical structure depending on the official constraints system being  
1546 referenced.

1547 Because there is no known global classification of all *Official Constraints Contexts* as  
1548 used here, any implementation must provide a set of recognised official constraints  
1549 classifications for use within the appropriate *Core Components* Registry  
1550 implementation.

1551 [C55] Individual *Core Component* implementations shall register used official  
1552 constraint classification schemes with the appropriate supporting *Core*  
1553 *Components* Registry implementation.

#### 1554 6.2.2.6 Business Process Role Context

1555 The *Business Process Role Context* describes those aspects of a business situation that  
1556 are specific to an actor or actors within the business process. Its values are taken from  
1557 the set of *Role* values provided by the *Catalogue of Common Business Processes*. A  
1558 *Business Process Role Context* is specified by using a value or set of values from this  
1559 source.

1560 [C56] *Business Process Role Context* values shall be taken from an approved list  
1561 provided by the business process model library being employed.

1562 [C57] The UN/CEFACT *Catalogue of Common Business Processes* shall be the  
1563 definitive source of *Business Process Role Context* values for all  
1564 UN/CEFACT *Business Information Entities*.

### 1565 6.2.2.7 Supporting Role Context

1566 The *Supporting Role Context* identifies those parties that are not active participants in  
1567 the business process being conducted but who are interested in it. A *Supporting Role*  
1568 *Context* is specified with a value or set of values from a standard classification.

1569 [C58] *Supporting Role Context* values shall be taken from the UN/EDIFACT *Code*  
1570 *List for DE 3035 Party Roles*.

1571 [Note]

1572 Users are cautioned that duplication exists in the current version of the required code  
1573 list. UN/CEFACT will review this code list to clarify duplicates and identify non-  
1574 *Supporting Role Context* values.

### 1575 6.2.2.8 System Capabilities Context

1576 This category identifies a system, a class of systems or standard in the business  
1577 situation. The *System Capabilities Context* requires a least one pair of values: an  
1578 identification of the classification scheme being used and a value from that scheme. A  
1579 valid *System Capabilities Context* may include more than one such pair of values.

1580 [C59] *Systems Capabilities Context* values shall consist of pairs of values. Each pair  
1581 shall be comprised of an identification of the referenced classification scheme  
1582 and the value(s) being employed.

1583 [Note]

1584 There is no known classification of all types of information systems and standards. It  
1585 is recommended that a mechanism for the registration of system and standard names  
1586 be provided by the ebXML registry, as valid values for the *System Capabilities*  
1587 *Context*.

## 1588 6.2.3 Context Values

1589 A specific business context is formally described using a set of context values. Every  
1590 context category must have a valid value, even if this value is *In All Contexts* or *None*.  
1591 The value *None* is appropriate for *Official Constraints Context* because there will be  
1592 instances where there are no official constraints.

1593 [C60] The *In All Contexts* value shall be a valid value for every context category  
1594 except for *Official Constraints Context*.

1595 [C61] The value *None* shall be a valid value for *Official Constraints Context*.

## 1596 6.2.4 Core Components Context Constraints Language

1597 The *Core Components Context Constraints Language* consists of a set of values (See  
 1598 Table 6-5) that allow users to express the relationships between specific business  
 1599 situations and the specific structure and meaning of business data used in that  
 1600 situation. The constraints language refers to specific contexts as described in the  
 1601 *Context Categories* specification and uses UIDs to refer to *Core Components*  
 1602 semantic models. The constraints applied to *Core Components* in specific business  
 1603 contexts to generate *Business Information Entities* are expressed using the constraints  
 1604 language.

1605 [C62] The *Core Components Context Constraints Language* shall be used to describe  
 1606 the constraints being applied to *Core Components* to create *Business*  
 1607 *Information Entities*.

1608 An *Assembly* is the overall expression of a single set of *Assembly Rules*, which groups  
 1609 a set of unrefined *Business Information Entities* in to a larger structure. When working  
 1610 with pre-assembled standard document sets, it should not be necessary for users to  
 1611 create *Assembly* constraints.

1612 [C63] *Assembly* shall be the top-level construct in any set of *Assembly Rules*.

1613 The *ContextRules* construct is the overall expression of a single set of rules that are  
 1614 used to apply context to *Core Components*. The *ContextRules* add the full semantic  
 1615 and structural refinement to the *Core Components* to produce *Business Information*  
 1616 *Entities*. This mechanism supports specifying cardinality, addition and subtraction of  
 1617 child core components, renaming of those children, assigning *Business Information*  
 1618 *Entity* names to the context-specific instances of the *Core Components*, and adding  
 1619 structure to create *Aggregate Business Information Entities*.

1620 [C64] A single set of context rules shall be described using the *ContextRules*  
 1621 expression.

1622 **Table 6-5 Core Components Context Constraints Language**

Construct	Component Constructs	Description
<b>Assembly</b>		An Assembly contains at least one Assemble, optionally either an @id or an @idref, and optionally one @version <b>Note:</b> An Assembly is the top level construct in a set of Assembly Rules
	Assemble	List of assembled <i>Core Components</i> to be grouped together to form BIEs
	@id	ID of an Assembly
	@idref	Reference to an Assembly id
	@version	Version of the Assembly Rules document.

Construct	Component Constructs	Description
<b>Assemble</b>		An Assemble contains at least either a CreateBIE or a CreateGroup, optionally either an @id or an @idref, and one @name
	CreateBIE	List of <i>Core Components</i>
	CreateGroup	Create a group of BIEs
	@name	Name of the highest-level BIE being assembled
	@id	ID of an Assemble rule
	@idref	Reference to an Assemble id
<b>CreateGroup</b>		A CreateGroup contains at least one of CreateGroup or CreateBIE or UseBIE or Annotation, optionally an @id or an @idref, and one @type
	@type	Type of group to be created (the only permitted values are 'sequence' and 'choice')
	@id	ID of a CreateGroup rule
	@idref	Reference to CreateGroup id
	CreateGroup	Create a group of BIEs
	CreateBIE	Create a BIE
	UseBIE	Use the named BIE from among the children of the BIE being created.
	Annotation	Insert Annotation
<b>CreateBIE</b>		A CreateBIE rule contains an optional Name followed by an optional Type followed by a MinOccurs followed by a MaxOccurs followed by zero or more CreateGroup or Rename, or UseBIE, or Condition or Annotation, optionally an @id or an @idref, and an optional @location
	Type	Type of BIE to be created – a reference to a <i>Core Component</i>
	MinOccurs	Minimum occurrences for the BIE created
	MaxOccurs	Maximum occurrences for the BIE created. One possible value (other than integer) is 'unbounded'.
	@id	Id of the created BIE
	@idref	Reference to the ID of another created BIE
	Name	Name of the BIE to be assembled
	@location	Location of the BIE to be assembled (i.e. query to the registry)
	Rename	Renames children of the created BIE
	Condition	Condition under which this rule should

Construct	Component Constructs	Description
		apply
	Annotation	Insert Annotation
<b>Name</b>		A Name contains only a string of characters
<b>Type</b>		A Type contains only a string of characters. It represents a type in the output – representation class or <i>Core Component</i> , depending on where used.
<b>Rename</b>		A Rename rule contains optionally an @id or an @idref, and one @from and one @to
	@id	Id of the Rename rule
	@idref	Reference to the ID of another Rename rule
	@from	Original name of the child BIE being renamed
	@to	New name of the child being renamed
<b>ContextRules</b>		ContextRules contains one or more Rules <b>Note:</b> A ContextRules is the top level construct in a set of Context Rules
	Rule	List of refinement and qualification rules to be applied
	@id	Id of the ContextRules rule
	@idref	Reference to the ID of another ContextRules rule
	@version	Version of the ContextRules document.
<b>Rule</b>		A Rule contains one or more Taxonomy, followed by one or more Condition, one @apply, and an optional @order.
	@apply	(See note below)
	Condition	When rule should be run
	@order	Defines order for running rules. Rules with lower value for order are run first
	Taxonomy	List of taxonomies used in a Rule that employs hierarchical conditions.
<b>Taxonomy</b>		A Taxonomy contains a @context and a @ref, and optionally an @id or an @idref
	@ref	Pointer to a taxonomy.
	@context	Name of the context category to which this Taxonomy applies
	@id	Id of the Taxonomy rule
	@idref	Reference to the ID of another Taxonomy rule
<b>Condition</b>		A Condition contains at least one of Action or Condition or Occurs, one

Construct	Component Constructs	Description
		@test, and optionally an @id or an @idref
	Action	What happens when rule is run
	Condition	A nested condition
	Occurs	Specify number of occurrences
	@id	Id of the Condition rule
	@idref	Reference to the ID of another Condition rule
	@test	Boolean expression testing whether the rule should be run.
<b>Action</b>		An Action contains at least one of Add or Occurs or Subtract or Condition or Comment or Rename, one @applyTo and optionally an @id or an @idref
	@applyTo	Node to apply action to
	Add	Add a component to the content model
	Subtract	Subtract a component from the content model
	Occurs	Constrain or expand the number of occurrences of the component
	Condition	When rule should be run
	Comment	Add a comment
	Rename	Rename a component
	@id	Id of the Condition rule
	@idref	Reference to the ID of another Condition rule
	@applyTo	Name of the component to apply this rule to
<b>Add</b>		Add contains a MinOccurs followed by a MaxOccurs followed by at least one of an optional BIE or an optional Attribute, or a CreateGroup or an Annotation, optionally an @id or an @idref, an optional @before or an optional @after
	MinOccurs	Minimum number of times that the new instance must occur
	MaxOccurs	Maximum number of times that the new instance can occur
	@before	Specifies before which component the addition should occur.
	@after	Specifies after which component the subtraction should occur.
	CreateGroup	Create a group of BIEs
	BIE	Adds a new BIE to the content model.
	Attribute	Adds a new non-BIE property to the content model
	Annotation	Insert Annotation



Construct	Component Constructs	Description
	@id	Id of the Add rule
	@idref	Reference to the ID of another Add rule
<b>Subtract</b>		Subtract contains one or more of BIE or Attribute, and optionally an @id or an @idref
	BIE	Removes a BIE from the content model.
	Attribute	Removes a non-BIE property from the content model
	@id	Id of the Subtract rule
	@idref	Reference to the ID of another Subtract rule
<b>Occurs</b>		Occurs contains a MinOccurs, followed by a MaxOccurs, followed by one or more BIEs, and optionally an @id or an @idref
	BIE	Changes an optional BIE to required.
	MinOccurs	Overrides the minimum number of occurrences for this BIE
	MaxOccurs	Overrides the maximum number of occurrences for this BIE
	@id	Id of the Occurs rule
	@idref	Reference to the ID of another Occurs rule
<b>BIE</b>		A BIE contains a Name, followed by an optional Type, followed by zero or more Attribute, followed by zero or more Annotation, and optionally an @id or an @idref
	Name	Name of BIE to be modified
	Type	Type of BIE – the <i>Core Component</i> - required only if contained in an Add tag
	Attribute	Attribute(s) of this BIE
	Annotation	Insert Annotation
	@id	Id of the BIE rule
	@idref	Reference to the ID of another BIE rule
<b>Attribute</b>		An Attribute contains an optional Name followed by an optional Type, followed by an optional Use, followed by an optional Value, followed by zero or more Annotation, and optionally an @id or an @idref, and an optional @applyTo
	Name	Name of attribute to be modified
	Type	Type of the attribute (representation class)
	Use	Indicates whether required or optional, and if the latter whether fixed or defaulted

<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
	Value	Indicates a fixed or defaulted value, or a value to be modified
	@id	Id of the Attribute rule
	@idref	Reference to the ID of another Attribute rule
<b>UseBIE</b>		A UseBIE contains zero or more of Annotation or CreateGroup or UseBIE, and optionally an @id or an @idref
	@name	Name of the BIE being used
	CreateGroup	Create a group of BIEs
	UseBIE	Use the named BIE from among the children of the BIE being created.
	Annotation	Insert Annotation
	@id	Id of the UseBIE rule
	@idref	Reference to the ID of another UseBIE rule
<b>Comment</b>		Ubiquitous. Records comments about the rules document at the location it appears. It is not intended to be output in the resulting semantic model.
<b>MinOccurs</b>		Minimum number of occurrences in the output
<b>MaxOccurs</b>		Maximum number of occurrences in the output
<b>Annotation</b>		An Annotation contains zero or more of either Documentation or Appinfo, and optionally an @id or an @idref
	Documentation	Used to include documentation
	Appinfo	Used to include application specific information
	@id	Id of the Annotation
	@idref	Reference to the ID of another Annotation
<b>Documentation</b>		Documentation contains optionally an @id or an @idref
	@id	Id of the Documentation
	@idref	Reference to the ID of another Annotation
<b>Appinfo</b>		Documentation contains optionally an @id or an @idref
	@id	Id of the Appinfo
	@idref	Reference to the ID of another Appinfo

1624 [Note]

1625 Table Key: @ indicates properties of the construct being defined. For example, @id,  
1626 @idref and @version are properties of Assembly.

#### 1627 6.2.4.1 Assembly Construct

1628 The *MinOccurs* and *MaxOccurs* constructs in the *CreateBIE* construct specify the  
1629 occurrence that the created *Business Information Entity* will have in the resulting  
1630 semantic model.

1631 [C65] A *Business Information Entity* created with *MinOccurs* = 1 and *MaxOccurs* =  
1632 1 shall be specified in the resulting semantic model as occurring only once.

1633 [C66] An *Assembly* may contain more than one assembled top-level semantic model.

#### 1634 6.2.4.2 ContextRule Construct

1635 Several built-in variables are used to access context information. These variables  
1636 correspond to the identified context categories. All of these variables have string  
1637 values.

1638 [C67] The *Apply* attribute of the *ContextRule* construct type shall be used for  
1639 determining the behaviour of rules that use hierarchical values.

1640 [C68] Allowed *Apply* attribute values are:

- 1641 • *exact* - match only if the value in the provided context is precisely the  
1642 same as that specified in the rule
- 1643 • *hierarchical* - match if the value provided is the same or a child of that  
1644 specified in the rule.

1645 [Example]

1646 If the *ContextRule* specifies the region *Europe*, the value *France* would match only if  
1647 the *Apply* attribute is set to *hierarchical* (*exact* being the default).

1648 [C69] The *Attribute* construct has four optional children in its content model, of  
1649 which at least one must be present.

1650 [C70] When the *Attribute* construct is used to refine an existing *Attribute*, then a  
1651 value must be specified for @applyTo on that *Attribute* construct.

1652 [C71] *ContextRules* must refer to the names of the *Core Components*, and not the  
1653 names given to the resulting *Business Information Entities* elsewhere in the  
1654 Rules.

1655 [Example]

1656 Given a source that contains an optional child type named 'X', a rule can be applied  
1657 to rename 'X' to 'Y', but a rule to make 'Y' required, rather than 'X', would be  
1658 illegal.

#### 1659 6.2.4.3 Output Constraints

1660 [C72] Semantic models and document definitions produced through the application  
1661 of *Assembly* and *Context Rules* must contain the metadata about the rules and  
1662 context that produced them.

#### 1663 6.2.4.4 Ordering and Application

1664 There is an explicit *Order* property on the *Rule* construct that applies a sequence to  
1665 the application of a set of rules. It is an error for two *Rule* constructs to have the same  
1666 value for the property *Order*. In a single set of *ContextRules*, users should be careful  
1667 not to sequence rules in a way that would preclude their execution—such as adding an  
1668 attribute to a business Information Entity that has not been added yet by the rules.

1669 [C73] The *Order* property on the *Rule* construct will determine the sequence for the  
1670 application of the applicable set of rules.

1671 [C74] Two *Rule* constructs must not have the same value for the property *Order*.

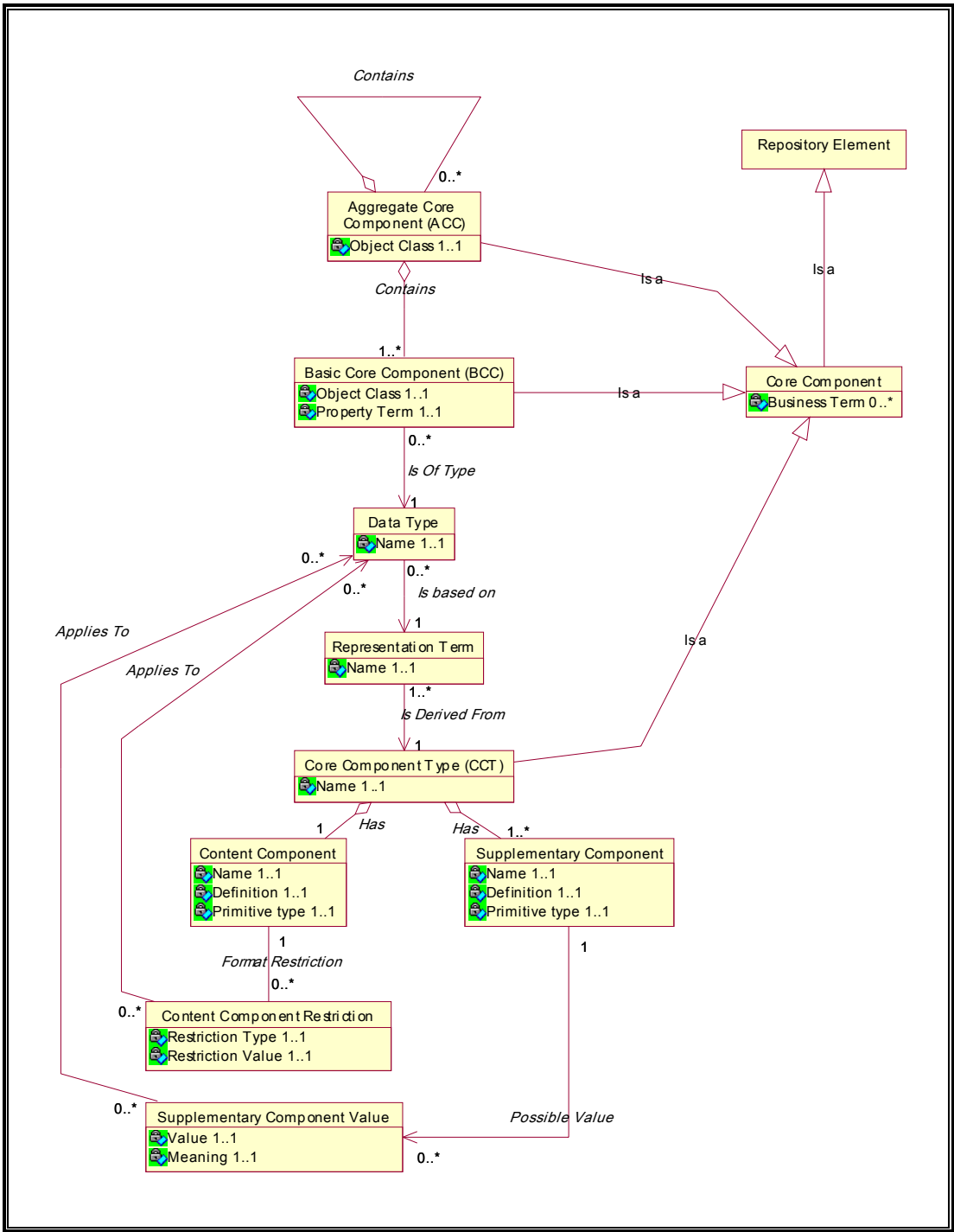
## 7 Technical Details - Core Component Repository Storage

Section 6 specifies the basic definitions for *Core Components*, *Business Information Entities* and *Context*. This section details exact information required for creation of *Unified Modelling Language* objects to store *Core Components*, *Business Information Entities*, *Context* and relevant associated metadata in the repository. Both parts contain requirements that must be addressed by developers of *Core Components* and users of *Core Components*. Further, both parts contain requirements that must be satisfied in the supported registry and repository suite of technical specifications and any corresponding overarching information technology framework that uses *Core Components* as the linchpin between process modelling and trade.

### 7.1 Storing Core Components

This section fully describes *Core Component* storage details. Figure 7-1 is the *Unified Modelling Language* model of all aspects of *Core Components* and fully describes the types of *Core Components* and their relationships as a requirement of storage.

1687 **Figure 7-1. Core Components - Full Definition**



### 1689 7.1.1 Stored Core Components

- 1690 [S1] Stored *Core Components* shall always be defined as one of the three recognised  
1691 types—*Basic Core Component*, *Aggregate Core Component*  
1692 or *Core Component Type*.
- 1693 [S2] Stored *Core Components* shall include the following attributes:
- 1694 • **Business Term (optional, repetitive):** A synonym term under which the *Core*  
1695 *Component* is commonly known and used in a business. A *Core Component*  
1696 may have several business terms or synonyms.

### 1697 7.1.2 Stored Basic Core Components

- 1698 [S3] Stored *Basic Core Components* shall always be based on three elements: (1) an  
1699 *Object Class*, (2) a *Property Term* and (3) a *Data Type*.
- 1700 [S4] Stored *Basic Core Components* shall include the following Attributes:
- 1701 • **Object Class (mandatory):** Basis for the *Dictionary Entry Name*.
  - 1702 • **Property Term (mandatory):** Basis for the *Dictionary Entry Name*.

### 1703 7.1.3 Stored Core Component Types

- 1704 [S5] Stored *Core Component Types* shall include one *Content Component* that defines  
1705 the primitive type and one or more *Supplementary Components* that give meaning  
1706 to the *Content Component*.
- 1707 [S6] Stored *Core Component Types* shall not reflect business meaning.
- 1708 [S7] Stored *Core Component Types* shall include the following attributes:
- 1709 • **Name (mandatory):** A meaningful type name, as basis for the *Dictionary*  
1710 *Entry Name*.

### 1711 7.1.4 Stored Aggregate Core Components

- 1712 [S8] Stored *Aggregate Core Components* shall consist of two or more *Basic Core*  
1713 *Components*, or at least one *Basic Core Component* plus one or more *Aggregate*  
1714 *Core Components*.
- 1715 [S9] Stored *Aggregate Core Components* shall identify all relationships with the *Basic*  
1716 *Core Components* and *Aggregate Core Components* from which they are  
1717 constructed.

1718 [S10] *Stored Aggregate Core Components* shall include the following Attributes:

- 1719 • **Object Class (mandatory):** Basis for the *Dictionary Entry Name*.

#### 1720 7.1.5 Stored Data Types

1721 [S11] *Stored Data Types* shall define the full range of valid values that can be used for a  
1722 particular *Basic Core Component* and will include the following attribute:

- 1723 • **Name (mandatory):** Name in the Repository of a *Data Type*.

1724 [S12] *Stored Data Types* shall be based on a *Representation Term* derived from a *Core*  
1725 *Component Type*.

1726 [S13] Restrictions on *Stored Content Components* and *Supplementary Components* shall  
1727 be identified when the *Core Component Type* is used as basis for a particular *Data*  
1728 *Type* and when it is necessary to restrict the format of the *Content Component*  
1729 and/or to restrict the possible values of a *Supplementary Component*.

#### 1730 7.1.6 Stored Representation Term

1731 [S14] *Stored Representation Terms* will include the following attribute:

- 1732 • **Name (mandatory):** Name in the Repository of a *Representation Term*

1733 [S15] *Stored Representation Terms* shall be derived from a *Core Component Type*.

#### 1734 7.1.7 Stored Supplementary Components

1735 [S16] *Stored Supplementary Components* shall be associated with the *Content*  
1736 *Component* in the overarching *Core Component Type* and shall include the  
1737 following attributes:

- 1738 • **Name (mandatory):** Name in the Repository of a *Supplementary Component*  
1739 of a *Core Component Type*.
- 1740 • **Definition (mandatory):** A clear, unambiguous and complete explanation of  
1741 the meaning of a *Supplementary Component* and its relevance for the related  
1742 *Core Component Type*.
- 1743 • **Primitive type (mandatory):** Primitive type to be used for the representation  
1744 of the value of a *Supplementary Component*.

1745 [Example]

1746 Possible values for primitive type are String, Decimal, Integer, Boolean, Date and Binary.



### 1747 **7.1.8 Stored Supplementary Component Value**

1748 [S17] A stored *Supplementary Component Value* shall define one possible value of a  
1749 *Supplementary Component*.

1750 [S18] A stored *Supplementary Component Value* shall only be stored if all possible  
1751 values can be defined by an enumeration (e.g. list of quantity units).

1752 [Note]

1753 The list of possible stored *Supplementary Component Values* can be further restricted  
1754 when a *Core Component Type* is used for a particular *Basic Component*.

1755 [Example]

1756 The *Core Component Type Quantity* has a *Supplementary Component Quantity Unit* with  
1757 possible values like gram and second. A *Basic Component* like *Person. Weight. Quantity*  
1758 will not accept *second* as quantity unit.

1759 [S19] Stored *Supplementary Component Values* shall contain the following Attributes:

- 1760 • **Value (mandatory):** A possible value of a *Supplementary Component*.
- 1761 • **Meaning (mandatory):** The meaning of the *Supplementary Component* when  
1762 it has a particular Value.

### 1763 **7.1.9 Stored Content Components**

1764 [S20] Stored *Content Components* shall contain the following attribute:

- 1765 • **Name (mandatory):** Name in the Repository of a *Content Component* of a  
1766 *Core Component Type*.
- 1767 • **Definition (mandatory):** A clear, unambiguous and complete explanation of  
1768 the meaning of a *Content Component*.
- 1769 • **Primitive type (mandatory):** Primitive type to be used for the expression of  
1770 the value of a *Basic Core Component* based on the associated *Core*  
1771 *Component Type*.

### 1772 **7.1.10 Stored Content Component Restrictions**

1773 [S21] Stored *Content Component Restrictions* shall only exist if the values can be  
1774 defined by a format restriction such as string pattern, minimum or maximum  
1775 length, or enumeration.

1776 [S22] Stored *Content Component Restrictions* shall contain the following attributes:

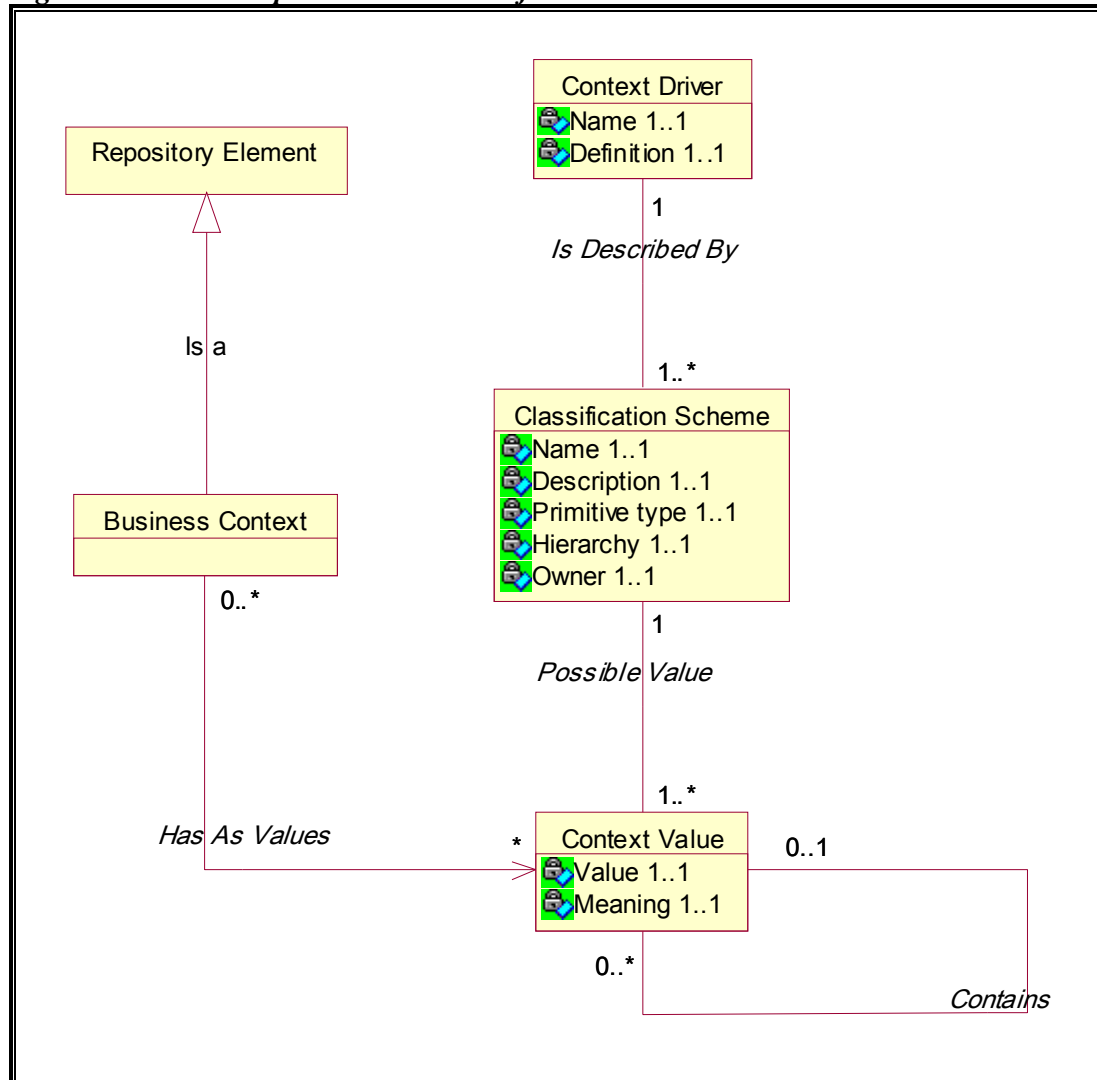
- 1777       • **Restriction Type (mandatory)**: Defines the type of format restriction that  
1778       must be applied to the *Content Component*.
- 1779       • **Restriction Value (mandatory)**: The actual value of the *Restriction Type* that  
1780       applies to a *Content Component*.

1781 [Example]

1782 Possible values include pattern, length, minimum length, maximum length, enumeration,  
1783 and others to be identified. The restriction values depend on the restriction type (e.g.  
1784 integer for a **length** restriction type, list of possible values for an **enumeration** restriction  
1785 type.).

## 1786 7.2 Stored Context

1787 This section fully describes *Context* storage details. Figure 7-2 is the *Unified Modelling*  
1788 *Language* model of all aspects of *Context*. It shows that there are a number of *Context*  
1789 *Categories* (e.g. Region, Product), which can each be described by one or more  
1790 Classification Schemes (e.g. United Nations scheme for products, World Trade  
1791 Organization scheme for products). For each Classification Scheme the list of possible  
1792 values (and their meaning) is defined. A *Business Context* is then defined as a unique and  
1793 meaningful combination of context values.

1794 **Figure 7-2 Core Components Context Definition Model**

1795

1796 [S23] Stored *Business Context* shall contain the combination of values for context  
 1797 categories so as to define a unique and meaningful *Business Context*.

### 1798 7.2.1 Context categories

1799 [S24] Stored *Context Categories* shall be in conformance with the officially accepted  
 1800 categories of *Core Component* contexts.

1801 [S25] Stored *Context Categories* shall contain the following attributes:

- 1802 • **Name (mandatory):** Name in the Repository of a *Context Category*.
- 1803 • **Definition (mandatory):** The meaning of the *Context Category*.

### 1804 7.2.2 Classification Scheme

1805 [S26] A *Context Category* may be described by one or more *Classification Schemes*.

1806 [S27] Stored *Classification Schemes* shall contain the following attributes:

- 1807 • **Name (mandatory):** Name under which the *Classification Scheme* is known.
- 1808 • **Definition (mandatory):** Definition of the *Classification Scheme*.
- 1809 • **Primitive type (mandatory):** Primitive type that is used for the representation  
1810 of a context value in the *Classification Scheme*.
- 1811 • **Hierarchy (mandatory):** Indicator describing whether the *Classification*  
1812 *Scheme* supports a hierarchical description of the context.
- 1813 • **Owner (mandatory):** Organisation that is responsible for the *Classification*  
1814 *Scheme*.

### 1815 7.2.3 Context Value

1816 [S28] Stored *Context Value* shall describe a particular context in a given Context  
1817 Category according to a particular Classification Scheme. If the *Classification*  
1818 *Scheme* allows a hierarchy, the value obtained via the association **contains**  
1819 describes this hierarchy.

1820 [S29] Stored *Context Value(s)* shall contain the following attributes:

- 1821 • **Value (mandatory):** Value describing a particular context.
- 1822 • **Meaning (mandatory):** Description of the meaning of the corresponding  
1823 value.

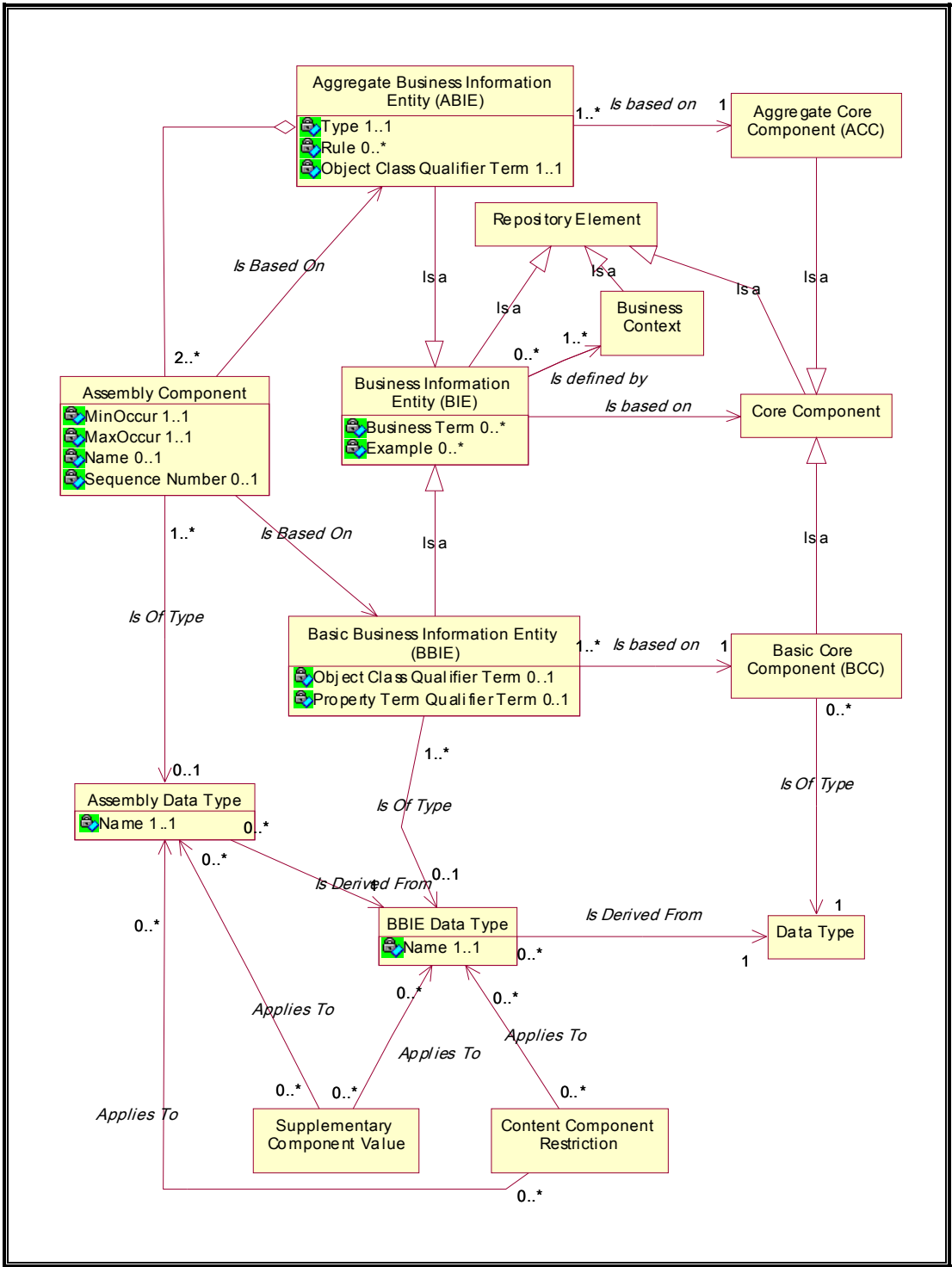
1824 [Note]

1825 The context value is derived from a business process model which presumably uses  
1826 values that have their meaning defined somewhere. For example, if the value is taken  
1827 from a code list (specified in the *Classification Scheme*), then the meaning of the code  
1828 should be provided by the code list specification. As an alternative solution, Meaning  
1829 could be an optional Uniform Resource Identifier that points to the definition.

## 1830 7.3 Stored Business Information Entities

1831 This section fully describes *Business Information Entity* storage details. Figure 7-3 is the  
1832 *Unified Modelling Language* model of all aspects of *Business Information Entity* and  
1833 fully describes the types of *Business Information Entities* and their relationships as a  
1834 requirement of storage.

1835 **Figure 7-3. Business Information Entities – Full Definition**



1837 An *Aggregate Business Information Entity* is either a sequence or a choice and will  
1838 consist of two or more *Assembly Components*, which are either *Basic Business*  
1839 *Information Entities* or *Aggregate Business Information Entities*. Each *Assembly*  
1840 *Component* has a certain cardinality (i.e. it is mandatory, optional and/or repetitive) and -  
1841 in case of a sequence - a sequence number. When used as an *Assembly Component*, it is  
1842 possible to change the name of the composing *Aggregate Business Information Entity* or  
1843 *Basic Business Information Entity* and to restrict the data type of a composing *Basic*  
1844 *Business Information Entity*.

### 1845 7.3.1 Stored Aggregate Business Information Entities

1846 [S30] Stored *Aggregate Business Information Entities* shall contain the following  
1847 attributes:

- 1848 • **Type (mandatory)**: Indicates whether the composing components of the  
1849 *Aggregate Business Information Entity* form a sequence (i.e. all composing  
1850 components may occur when the *Aggregate Business Information Entity* is  
1851 used) or a choice (i.e. only one of the composing components may occur when  
1852 the *Aggregate Business Information Entity* is used).
- 1853 • **Rule (optional, repetitive)**: Describes a restriction that relates to various  
1854 *Assembly Components* of the *Aggregate Business Information Entity*.
- 1855 • **Object Class Qualifier Term (mandatory)**: Qualifies the *Object Class* of the  
1856 associated *Aggregate Core Component*.

### 1857 7.3.2 Stored Assembly Component

1858 [S31] A stored *Assembly Component* shall be either an *Aggregate Business Information*  
1859 *Entity* or a *Basic Business Information Entity*.

1860 [S32] Stored *Assembly Components* shall contain the following attributes:

- 1861 • **MinOccur (mandatory)**: Minimum number of occurrences that a composing  
1862 *Business Information Entity* must occur when used in an *Aggregate Business*  
1863 *Information Entity*. If the minimum is zero, the component is optional. If the  
1864 minimum is one or more, the component is mandatory.
- 1865 • **MaxOccur (mandatory)**: Maximum number of occurrences that a composing  
1866 *Business Information Entity* may occur when used in an *Aggregate Business*  
1867 *Information Entity*. If the maximum is zero, the component is not allowed. If  
1868 the maximum is more than one, the component is repetitive. The defined  
1869 maximum must always be greater than or equal to the defined minimum.
- 1870 • **Name (optional)**: Alternative name to be used for a *Business Information*  
1871 *Entity* when used in an *Aggregate Business Information Entity*.
- 1872 • **Sequence Number (optional)**: Position of the *Assembly Component* in an  
1873 *Aggregate Business Information Entity* of type Sequence.

### 1874 7.3.3 Stored Assembly Data Type

1875 [S33] *Stored Assembly Data Types* shall be defined by specifying additional restrictions  
1876 on the *Content Component* and *Supplementary Components* of the *Basic Business*  
1877 *Information Entity Data Type* from which it is derived.

1878 [S34] *Stored Assembly Data Types* shall contain the following attribute:

- 1879 • **Name (mandatory):** Name in the Repository of a *Assembly Data Type*.

### 1880 7.3.4 Stored Basic Business Information Entities

1881 [S35] *Stored Basic Business Information Entities* shall contain the following attributes:

- 1882 • **Object Class Qualifier Term (optional):** Qualifies the *Object Class* of the  
1883 associated *Basic Core Component*.
- 1884 • **Property Term Qualifier Term (optional):** Qualifies the *Property Term* of the  
1885 associated *Basic Core Component*.

1886 [S36] *Stored Basic Business Information Entities* shall contain at least one of the  
1887 *Qualifier Terms* specified in Rule S35.

### 1888 7.3.5 Stored Basic Business Information Entity Data Types

1889 [S37] A *Basic Business Information Entity Data Type* will be defined by specifying  
1890 additional restrictions on the *Content Component* and *Supplementary Components*  
1891 of the *Data Type* from which it is derived.

1892 [S38] *Stored Basic Business Information Entity Data Types* shall contain the following  
1893 attributes:

- 1894 • **Name (mandatory):** Name in the Repository of a *Basic Business Information*  
1895 *Entity Data Type*.

### 1896 7.3.6 Stored Business Information Entities

1897 [S39] *Stored Business Information Entities* shall always be defined as either a *Basic*  
1898 *Business Information Entity* or an *Aggregate Business Information Entity*.

1899 [S40] A *stored Business Information Entity* shall contain the following attributes:

- 1900 • **Business Term (optional, repetitive):** A synonym term under which the  
1901 *Business Information Entity* is commonly known and used in the business. A  
1902 *Business Information Entity* may have several business terms or synonyms.

- 1903 • **Example (optional, repetitive):** Example of a possible value of a *Business*  
1904 *Information Entity*

## 1905 7.4 Core Component Storage Metadata

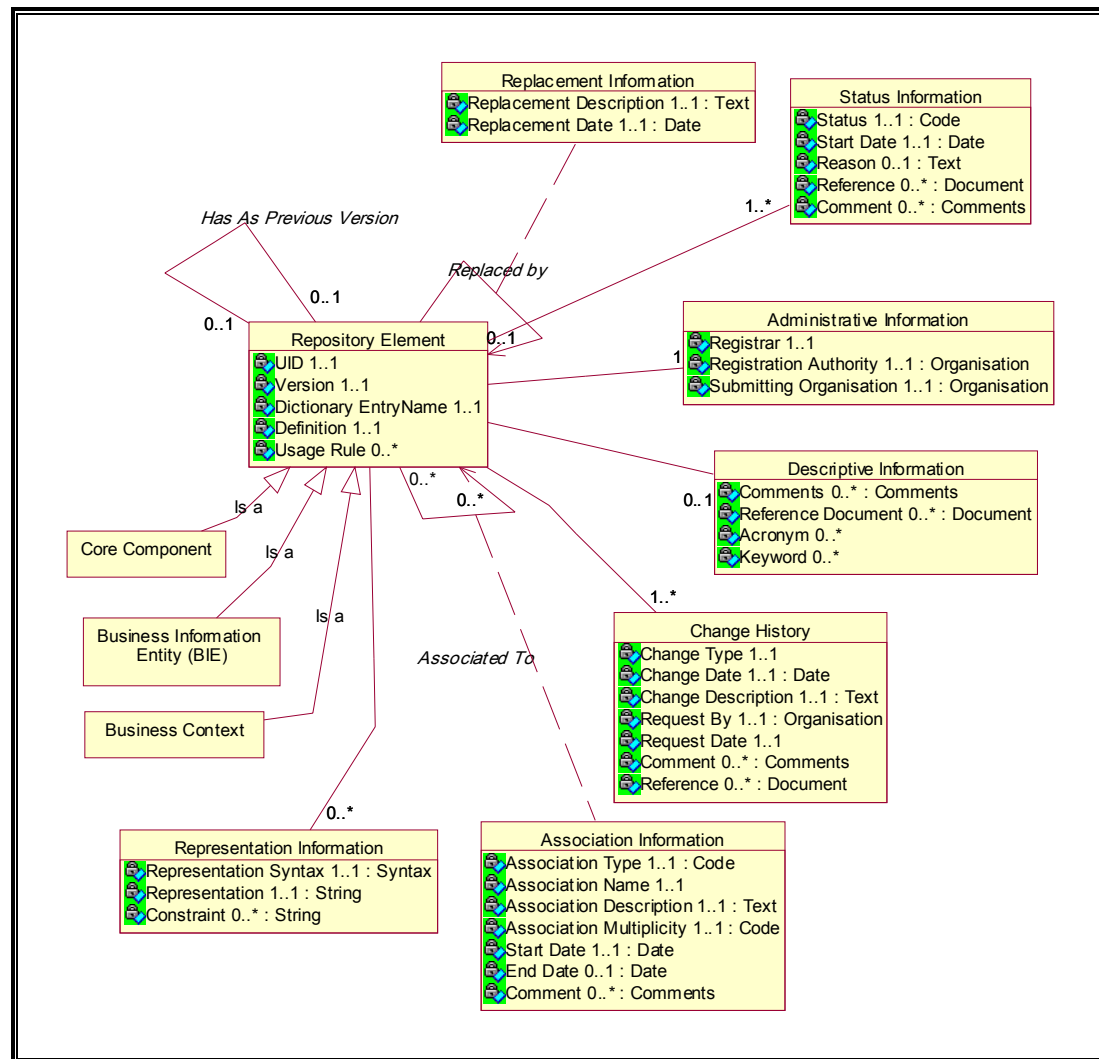
1906 *Core Components* and *Business Information Entities* are the linchpins for developing  
1907 standardised process models and business documents. Storing these artefacts so that they  
1908 are able to meet this role requires rich metadata storage as well.

1909 Figure 7-4 focuses on the meta-information that needs to be defined for a *Repository*  
1910 *Element* (i.e. all information needed to store for *Core Components* and for *Business*  
1911 *Information Entities*). To simplify the diagram all information regarding the structure of a  
1912 *Core Component* and a *Business Information Entity* has been hidden.

1913 As shown in Figure 7-4, the following metadata categories will be required:

- 1914 • **Version Information:** even though at any given point in time only one  
1915 version of a *Repository Element* can be valid, multiple previous versions may  
1916 have existed and a future version may be in preparation. The **Version**  
1917 association makes it possible to link the consecutive versions of a *Repository*  
1918 *Element*. There will not be branches in the versioning; only a linear versioning  
1919 will be supported.
- 1920 • **Replacement Information:** a *Repository Element* may be replaced by another  
1921 *Repository Element* at some point in time (e.g. because a duplicate is  
1922 discovered). The **Replaced by** association makes it possible to do this and  
1923 *Replacement Information* makes it possible to document the date of and reason  
1924 for replacement.
- 1925 • **Status Information:** information about the live status of a *Repository Element*
- 1926 • **Administrative Information:** information about the registration of the  
1927 *Repository Element*.
- 1928 • **Descriptive Information:** additional descriptive information about a  
1929 *Repository Element*, giving further clarification about its meaning.
- 1930 • **Change History:** information about all changes that are made to a *Repository*  
1931 *Element*.
- 1932 • **Association Information:** a *Repository Element* may be associated to multiple  
1933 other *Repository Elements*.
- 1934 • **Representation Information:** information about the physical representation  
1935 of a *Repository Element* in a particular syntax (e.g. to document the XML-  
1936 tag).



1937 **Figure 7-4. Repository Metadata**

1938

1939 **7.4.1 General Metadata Storage Rules**1940 [S41] Stored *Repository Elements* shall include a universally unique identifier.1941 [S42] Stored *Repository Elements* shall include a version number to keep track of the  
1942 evolution over time of a *Repository Element*.1943 [S43] Stored *Repository Elements* shall include a *Dictionary Entry Name*.1944 [S44] Stored *Repository Elements* shall include a Definition.1945 [S45] Stored *Repository Elements* may include one or more *Usage Rules*, describing  
1946 how and/or when to use the *Repository Element*.

1947 [S46] Except for the first version of a *Repository Element*, each stored version shall be  
1948 linked to its previous version.

1949 [S47] Except for the last version of a *Repository Element*, each stored version shall be  
1950 linked to its next version.

1951 [S48] Stored *Repository Elements* shall include the history of the status lifecycle of each  
1952 version.

## 1953 **7.4.2 Management Information**

### 1954 7.4.2.1 Administrative Information

1955 [S49] Stored *Repository Elements* shall contain administrative information and will  
1956 include the following attributes:

- 1957 • **Registrar (mandatory):** Name of the responsible person who has created the  
1958 *Repository Element* in the repository
- 1959 • **Registration Authority (mandatory):** Organisation authorised to register the  
1960 *Repository Element*.
- 1961 • **Submitting Organisation (mandatory):** The organisation that has submitted /  
1962 requested the *Repository Element*

### 1963 7.4.2.2 Status Information

1964 [S50] Stored *Repository Elements* shall contain status information to include the  
1965 following attributes:

- 1966 • **Status (mandatory):** Status of the *Repository Element* (i.e. draft,  
1967 provisionally registered, registered, to be retired, retired, ...)
- 1968 • **Start Date (mandatory):** Date on which the status comes into effect
- 1969 • **Reason (optional):** Description of why the *Repository Element* status has been  
1970 changed.
- 1971 • **Reference (optional, repetitive):** External Document(s) containing relevant  
1972 information about the status change.
- 1973 • **Comment (optional, repetitive):** Remark about the *Repository Element*  
1974 status.

### 1975 7.4.2.3 Change History

1976 [S51] Stored *Repository Elements* shall include the history of all modifications related to  
1977 each version to include the following attributes:

- 1978 • **Change Type (mandatory):** Purpose of the Change—such as *new element*,  
1979 *new version*, *element modification*, *status modification*, *element replacement*.
- 1980 • **Change Date (mandatory):** Date on which the modification has been made.
- 1981 • **Change Description (mandatory):** Description of why and how the  
1982 *Repository Element* has been modified.
- 1983 • **Request By (mandatory):** Name of the organisation that has requested the  
1984 modification of the *Repository Element*
- 1985 • **Request Date (mandatory):** Date on which the modification was requested.
- 1986 • **Comment (optional, repetitive):** Remark about the *Repository Element*  
1987 modification.
- 1988 • **Reference (optional, repetitive):** External Document(s) containing relevant  
1989 information about the modification.
- 1990 7.4.2.4 Replacement Information
- 1991 [S52] For each stored pair of *Repository Elements* where one *Repository Element*  
1992 replaces the other, the stored information shall specify replacement information to  
1993 include the following attributes:
- 1994 • **Replacement Description (mandatory):** Reason for the *Repository Element*  
1995 being replaced
- 1996 • **Replacement Date (mandatory):** Date from which the replacement is  
1997 effective.
- 1998 [S53] If another *Repository Element* has replaced a *Repository Element*, it shall be  
1999 linked to the *Repository Element* by which it has been replaced.
- 2000 [S54] If a *Repository Element* replaces one or more other *Repository Element*, it shall be  
2001 linked to the *Repository Element(s)* it replaces
- 2002 **7.4.3 Content Information**
- 2003 7.4.3.1 Descriptive Information
- 2004 [S55] Stored *Repository Elements* may optionally include additional descriptive  
2005 information to include the following attributes:
- 2006 • **Comments (optional, repetitive):** Comments is additional information about  
2007 a *Repository Element*, which is not part of the definition but that is considered  
2008 relevant for clarification.

- 2009      • **Reference Document (optional, repetitive):** Reference Document is a  
 2010      reference (e.g. a Uniform Resource Locator) to external documentation that  
 2011      contains relevant additional information about a *Repository Element*.
- 2012      • **Acronym (optional, repetitive):** Acronym is an abbreviation or code under  
 2013      which the *Repository Element* is commonly known.
- 2014      • **Keyword (optional, repetitive):** Keyword is one or more significant words  
 2015      used for the search and retrieval of a *Repository Element*.
- 2016      7.4.3.2    Representation Information
- 2017      [S56]    Stored *Repository Elements* may optionally include information about the  
 2018      representation of the *Repository Element* in one or more syntaxes to include the  
 2019      following attributes.
- 2020      • **Representation Syntax (mandatory):** Identification of the representation  
 2021      syntax
- 2022      • **Representation (mandatory):** Physical representation of the *Repository*  
 2023      *Element* (e.g. *Extensible Markup Language* tag)
- 2024      • **Constraint (optional, repetitive):** Description of additional constraints that  
 2025      apply to the representation of the *Repository Element* in the given syntax (e.g.  
 2026      maximum length, ...)
- 2027      7.4.3.3    Association Information
- 2028      [S57]    Stored *Repository Elements* shall include all associations they have with other  
 2029      stored *Repository Elements* and shall include the following attributes:
- 2030      • **Association Name (mandatory):** Name of the association
- 2031      • **Association Description (mandatory):** Descriptive text explaining the  
 2032      meaning of the association
- 2033      • **Association Type (mandatory):** Type of association (e.g. aggregation,  
 2034      specialisation, generalisation, simple association ...)
- 2035      • **Association Multiplicity (mandatory):** Cardinality of the association (i.e.  
 2036      optional/mandatory and repetition)
- 2037      • **Start Date (mandatory):** Date at which the association becomes valid
- 2038      • **End Date (optional):** Date from which the association is no longer valid
- 2039      • **Comment (optional, repetitive):** Relevant information about the association  
 2040      (e.g. reason why it has been removed, ...)

## 2041 8 Approved Core Component Type, Content, and 2042 Supplementary Components

2043 The Following subsections contain tables that convey the currently approved *Core*  
2044 *Component Types* (Section 8.1) and the *Core Component Type Content* and  
2045 *Supplementary Components* (Section 8.2).

### 2046 8.1 Approved Core Component Types

2047 [Note]

2048 The UIDs in Table 8-1 are interim in nature and will be finalised prior to release of  
2049 this document in specification status.

2050 *Table 8-1 Core Component Types (CCT)*

2051

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000105	Amount. Type	A number of monetary units specified in a currency where the unit of currency is explicit or implied.		Amount	Type	<ul style="list-style-type: none"> <li>Amount. Content (000106)</li> <li>Amount Currency. Identification. Code (000107)</li> </ul>
000089	Code. Type	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute together with relevant supplementary information.		Code	Type	<ul style="list-style-type: none"> <li>Code. Content (000091)</li> <li>Code List. Identifier (000092)</li> <li>Code List. Agency. Identifier (000093)</li> <li>Code List. Version. Identifier (000099)</li> <li>Code. Name (000100)</li> <li>Language. Code (000075)</li> </ul>
000066	Date Time. Type	A particular point in the progression of time together with relevant supplementary information.	Can be used for a date and/or time.	Date Time	Type	<ul style="list-style-type: none"> <li>Date Time. Content (000067)</li> <li>Date Time. Format. Text (000068)</li> </ul>
000200	Graphic. Type	A diagram, graph, mathematical curves, or similar representation.		Graphic	Type	<ul style="list-style-type: none"> <li>Graphic. Content</li> <li>Graphic. Format. Text</li> </ul>

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000101	Identifier. Type	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme together with relevant supplementary information.		Identifier	Type	<ul style="list-style-type: none"> <li>Identifier. Content (000102)</li> <li>Identification Scheme. Name (000103)</li> <li>Identification Scheme Agency. Name (000104)</li> <li>Language. Code (000075)</li> <li>Identification Scheme. Data. Uniform Resource Identifier (000209)</li> <li>Identification Scheme. Uniform Resource Identifier (000208)</li> </ul>
000180	Indicator. Type	A list of two, and only two, values, which indicate a condition such as on/off; true/false etc. (synonym: "Boolean").		Indicator	Type	<ul style="list-style-type: none"> <li>Indicator. Content (000181)</li> <li>Indicator. Format. Text</li> </ul>
000152	Measure. Type	The size, volume, mass, amount or scope derived by performing a physical measure together with relevant supplementary information.		Measure	Type	<ul style="list-style-type: none"> <li>Measure. Content (000153)</li> <li>Measure Unit. Code (000154)</li> </ul>
000182	Numeric. Type	A representation of a number.	May or may not be decimal	Numeric	Type	<ul style="list-style-type: none"> <li>Numeric. Content (000183)</li> <li>Numeric. Format. Text</li> </ul>
000201	Picture. Type	A visual representation of a person, object, or scene.		Picture	Type	<ul style="list-style-type: none"> <li>Picture. Content</li> <li>Picture. Format. Text</li> </ul>
000108	Quantity. Type	A number of non-monetary units together with relevant supplementary information.		Quantity	Type	<ul style="list-style-type: none"> <li>Quantity. Content (000109)</li> <li>Quantity. Unit. Code (000110)</li> <li>Quantity Unit. Code List. Identifier (000111)</li> <li>Quantity Unit. Code List Agency. Identifier (000112)</li> </ul>
000090	Text. Type	A character string with or without a specified language.		Text	Type	<ul style="list-style-type: none"> <li>Text. Content (000094)</li> <li>Language. Code (000075)</li> </ul>

2052

## 2053 8.2 Approved Core Component Type Content and 2054 Supplementary Components

2055 Table 8-2 presents the currently approved set of *Core Component Type Content* and  
 2056 *Supplementary Components*. The asterisk (\*) in the property term column indicates  
 2057 cases where the property term is the same as either the representation term or object  
 2058 class, and is consequently dropped from the dictionary entry name.

2059

2059

**Table 8-2. Core Component Type Content and Supplementary Components**

2060

UID	Name	Data-type	Definition	Remarks
000106	Amount. Content	decimal	A number of monetary units specified in a currency where the unit of currency is explicit or implied	
000107	Amount Currency. Identification. Code	string	The currency of the amount	Reference ISO 4217.
000091	Code. Content	string	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute	
000093	Code List. Agency. Identifier	string	An agency that maintains one or more code lists	
000092	Code List. Identifier	string	The name of a list of codes	Can be used to identify the URL of a source that defines the set of currently approved permitted values
000099	Code List. Version. Identifier	string	The version of the code list	
000100	Code. Name	string	The textual equivalent of the code content	If no code content exists, the code name can be used on its own
000067	Date time. Content	string	The particular point in the progression of time	
000068	Date Time. Format. Text	string	The format of the date/time content	Reference ISO 8601
000202	Graphic. Content	binary	A diagram, graph, mathematical curves, or similar representation	
000203	Graphic. Format. Text	string	The format of the graphic content	
000104	Identification Scheme Agency. Name	string	The agency that maintains the identification scheme	
000103	Identification Scheme. Name	string	The name of the identification scheme	
000209	Identification Scheme. Data. Uniform Resource Identifier	string	The Uniform Resource Identifier that identifies where the Identification Scheme Data is located	
000208	Identification Scheme. Uniform Resource Identifier	string	The Uniform Resource Identifier that identifies where the Identification Scheme is located	
000102	Identifier. Content	string	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme	
000181	Indicator. Content	string	The value of the indicator	For example on, off, true, false
000207	Indicator. Format. Text	String	Whether the indicator is numeric, textual or binary	
000075	Language. Code	string	The identifier of the language used in the corresponding text string	Reference ISO 639: 1998
000153	Measure. Content	decimal	The size, volume, mass, amount or scope derived by performing a physical measure	For example, 20 kilograms (20 is the measure content)

UID	Name	Data-type	Definition	Remarks
000154	Measure Unit. Code	string	The type of unit of measure	Reference UN/ECE Recommendation #20 and X12 355. For example, for \$10/100 km use CCT quantity type and for a measured distance of 20 kilometres use CCT measure type
000183	Numeric. Content	As defined by Numeric. Format.	The representation of a number	May be decimal
000204	Numeric. Format. Text	string	Whether the number is an integer, decimal, real number or percentage	
000205	Picture. Content	binary	A visual representation of a person, object, or scene	
000206	Picture. Format. Text	string	The acronym of the coding scheme used to record the picture	
000109	Quantity. Content	decimal	A number of non-monetary units	
000110	Quantity. Unit. Code	string	The unit of the quantity	May use UN/ECE Recommendation #20 and X12 355, but for actual measurements use the CCT measure type. For example, for \$10/100 km use CCT quantity type and for a measured distance of 20 kilometres use CCT measure type
000112	Quantity Unit Code List Agency. Identifier	string	The agency which maintains the quantity unit code list	
000111	Quantity Unit Code List. Identifier	string	The quantity unit code list	
000094	Text. Content	string	A character string generally in the form of words	

2061



## 2061 9 Definition of Terms

2062 **Aggregate Business Information Entity (ABIE)**– A collection of related pieces of  
2063 business information that together convey a distinct business meaning in a specified  
2064 business context.

2065 **Aggregate Core Component - (ACC)** – A collection of *Core Components* that convey  
2066 a distinct business meaning. An *Aggregate Core Component* will consist of two or  
2067 more *Basic Core Components*, or at least one *Basic Core Component* plus one or  
2068 more *Aggregate Core Components*.

2069 **Assembly Component** – An *Assembly Component* is a *Business Information Entity*  
2070 that is a component part of an *Aggregate Business Information Entity*.

2071 **Assembly Data Type** - The formal definition of a set of valid values that can be used  
2072 for a particular *Basic Business Information Entity* when used as an assembly  
2073 component in a particular *Aggregate Business Information Entity*.

2074 **Basic Business Information Entity (BBIE)** – A *Core Component* used in a specific  
2075 business context. A *Basic Business Information Entity* is derived from a *Basic Core*  
2076 *Component*.

2077 **Basic Business Information Entity Data Type** - The formal definition of a set of  
2078 valid values that can be used for a particular *Basic Information Entity*.

2079 **Basic Core Component (BCC)** – A *Core Component* that represents a singular  
2080 business concept with a unique business semantic definition. A *Basic Core*  
2081 *Component* is constructed by using a *Core Component Type*. *Basic Core Components*  
2082 are used in developing *Aggregate Core Components*.

2083 **Business Context** – The formal description of a specific business circumstance as  
2084 identified by the values of a set of context categories, allowing different business  
2085 circumstances to be uniquely distinguished.

2086 **Business Information Entity (BIE)** – A *Business Information Entity* is a piece of  
2087 business data or a group of pieces of business data with a unique business semantic  
2088 definition. A *Business Information Entity* can be either a *Basic Business Information*  
2089 *Entity* or an *Aggregate Business Information Entity*.

2090 **Business Term** – This is a synonym term under which the *Core Component* or  
2091 *Business Information Entity* is commonly known and used in the business. A *Core*  
2092 *Component* or *Business Information Entity* may have several business terms or  
2093 synonyms.

2094 **Cardinality** – The number of elements in a particular set or other grouping.

- 2095 **Child Core Component** – A *Core Component* that is used in a larger aggregate  
2096 construct.
- 2097 **Classification Scheme**– This is an officially supported scheme to describe a given  
2098 *Context Category*.
- 2099 **Constraint Language** – A formal expression of actions occurring in specific contexts  
2100 to assemble, structurally refine, and semantically qualify *Core Components*. The  
2101 result of applying the constraint language to a set of *Core Components* in a specific  
2102 context is a set of *Business Information Entities*.
- 2103 **Content Component** - Defines the primitive type used to express the content of a  
2104 *Core Component Type*.
- 2105 **Content Component Restrictions** - The formal definition of a format restriction that  
2106 applies to the possible values of a *Content Component*.
- 2107 **Context** – Defines the circumstances in which a business process can be used. This is  
2108 specified by a set of context categories known as a *Business Context*.
- 2109 **Context Category** – A group of one or more related values used to express one  
2110 characteristic of a business circumstance.
- 2111 **Context Information Entity** – The influence of a particular context on the restriction  
2112 on a reusable semantic building block for the exchange of business-related  
2113 information.
- 2114 **Controlled Vocabulary** - A supplementary vocabulary to define uniquely any words  
2115 or business terms that are potentially ambiguous. This is to ensure that every word  
2116 within any of the *Core Component* names and definitions is used in a consistent and  
2117 unambiguous way and this will also aid accurate language translations.
- 2118 **Core Component (CC)** – A building block for the creation of a semantically correct  
2119 and meaningful information exchange ‘parcel’. It contains only the information pieces  
2120 necessary to describe a specific concept.
- 2121 **Core Component Administrative Information** – Administrative information  
2122 regarding a *Core Component*
- 2123 **Core Component Association Information** – Information about the association  
2124 between two *Core Components*.
- 2125 **Core Component Catalogue** – The temporary collection of all metadata about each  
2126 *Core Component* that has been discovered during the development and initial testing  
2127 of this Core Component Technical Specification, pending the establishment of a  
2128 permanent Repository.

- 2129 **Core Component Change History** – History of the modifications applied to a *Core*  
2130 *Component* version.
- 2131 **Core Component Library** – The *Core Component Library* is that part of the  
2132 Repository (the elusive ARK of ebXML mythology) in which *Core Components* shall  
2133 be stored as *Repository Elements*. The *Core Component Library* will contain all the  
2134 *Core Component Types*, *Basic Core Components*, *Aggregate Core Components*, *Basic*  
2135 *Business Information Entities* and *Aggregate Business Information Entities*.
- 2136 **Core Component Replacement Information** – Information about the replacement of a  
2137 *Core Component* by another.
- 2138 **Core Component Representation Information** – Information about the physical  
2139 representation of a *Core Component* in a particular syntax.
- 2140 **Core Component Status Information** – History of the lifecycle of a particular version  
2141 of a *Core Component*.
- 2142 **Core Component Type (CCT)**– A *Core Component* that consists of one and only one  
2143 *Content Component* that carries the actual content plus one or more supplementary  
2144 components giving an essential extra definition to the content component. *Core*  
2145 *Component Types* do not have business meaning.
- 2146 **Data Type** – Defines the set of valid values that can be used for a particular *Basic*  
2147 *Core Component*. It is defined by specifying restrictions on the *Core Component Type*  
2148 that forms the basis of the *Representation Term* from which the *Data Type* is derived.
- 2149 **Definition** - This is the unique semantic business meaning of a *Core Component*,  
2150 *Business Information Entity* or *Business Context*.
- 2151 **Dictionary Entry Name** – This is the unique official name of a *Core Component*,  
2152 *Business Information Entity* or *Business Context* in the dictionary.
- 2153 **Information Entity** – A reusable semantic building block for the exchange of  
2154 business-related information.
- 2155 **Object Class** – The logical data grouping (in a logical data model) to which a data  
2156 element belongs (ISO11179). The *Object Class* is the part of a *Core Component*'s  
2157 *Dictionary Entry Name* that represents an activity or object in a specific context.
- 2158 **Primitive Type** – Primitive type used for the representation of the value of a  
2159 *Supplementary Component*. Possible values are String, Decimal, Integer, Boolean,  
2160 Date.
- 2161 **Property Term** – This identifies one of the characteristics belonging to the *Object*  
2162 *Class*.

- 2163 **Qualifier Term** - A word or words which help to define and differentiate a *Business*  
2164 *Information Entity* from its associated *Core Component* and other *Business*  
2165 *Information Entities*.
- 2166 **Repository Element** - The formal definition of all the information that is needed to be  
2167 recorded in the Repository about a *Core Component* or a *Business Information Entity*.
- 2168 **Representation Term** – The type of valid values for a *Basic Core Component*.
- 2169 **Supplementary Component** – Gives meaning to the *Content Component* in the *Core*  
2170 *Component Type*.
- 2171 **Usage Rules** - *Usage Rules* describe how and/or when to use the *Repository Element*.
- 2172 **User Community** – A user community is a group of practitioners, with a publicised  
2173 contact address, who may define context profiles relevant to their area of business.  
2174 Users within the community do not create, define or manage their individual context  
2175 needs but conform to the community's standard. Such a community should liaise  
2176 closely with other communities and with general standards-making bodies to avoid  
2177 overlapping work and to avoid creating multiple *Towers of Babel*. A community may,  
2178 of course, be as small as two consenting organisations.
- 2179 **XML schema** – A generic term used to identify the family of grammar based XML  
2180 document structure validation languages to include the more formal W3C XML  
2181 Schema Technical Specification, Document Type Definition, Schematron, Regular  
2182 Language Description for XML (RELAX), and the OASIS RELAX NG.

## 2183 10 References

- 2184 — *ebXML Technical Architecture Specification* v1.04
- 2185 — *ebXML Business Process Specification Schema* v1.01
- 2186 — *ebXML Registry Information Model* v1.0
- 2187 — *ebXML Registry Services Specification* v1.0
- 2188 — *ebXML Requirements Specification* v1.06
- 2189 — *ebXML Collaboration-Protocol Profile and Agreement Specification* v1.0
- 2190 — *ebXML Message Service Specification* v1.0
- 2191 — *ebXML Technical Report, Business Process and Business Information Analysis*
- 2192 *Overview* v1.0
- 2193 — *ebXML Technical Report, Business Process Analysis Worksheets & Guidelines*
- 2194 *v1.0 -*
- 2195 — *ebXML Technical Report, E-Commerce Patterns* v1.0
- 2196 — *ebXML Technical Report, Catalog of Common Business Processes* v1.0
- 2197 — *ebXML Technical Report, Core Component Overview* v1.05
- 2198 — *ebXML Technical Report, Core Component Discovery and Analysis* v1.04
- 2199 — *ebXML Technical Report, Context and Re-Usability of Core Components* v1.04
- 2200 — *ebXML Technical Report, Guide to the Core Components Dictionary* v1.04
- 2201 — *ebXML Technical Report, Naming Convention for Core Components* v1.04
- 2202 — *ebXML Technical Report, Document Assembly and Context Rules* v1.04
- 2203 — *ebXML Technical Report, Catalogue of Context Categories* v1.04
- 2204 — *ebXML Technical Report, Core Component Dictionary* v1.04
- 2205 — *ebXML Technical Report, Core Component Structure* v1.04
- 2206 — *Information Technology — Metadata registries: Framework for the*
- 2207 *Specification and Standardization of Data Elements*, International
- 2208 *Standardization Organization, ISO 11179-1*
- 2209 — *Information Technology — Metadata registries: Classification of Concepts for*
- 2210 *the Identification of Domains*, International Standardization Organization, ISO
- 2211 *11179-2*
- 2212 — *Information Technology — Metadata registries: Registry Metamodel,*
- 2213 *International Standardization Organization, ISO 11179-3*
- 2214 — *Information Technology — Metadata registries: Rules and Guidelines for the*
- 2215 *Formulation of Data Definitions*, International Standardization Organization,
- 2216 *ISO 11179-4*
- 2217 — *Information Technology — Metadata registries: Naming and Identification*
- 2218 *Principles for Data Elements*, International Standardization Organization, ISO
- 2219 *11179-5*

- 2220 — Information Technology — *Metadata registries: Framework for the*  
2221 *Specification and Standardization of Data Elements*, International  
2222 Standardization Organization, ISO 11179-6
- 2223 — *Information Technologies – Open-edi Reference Model*, ISO/IEC 14662
- 2224 — *Information Technologies – Business Agreement Semantic Descriptive*  
2225 *Techniques – Part 1: Operational Aspects of Open-edi for Implementation* ,  
2226 ISO/IEC 15944-1
- 2227 — *UN/CEFACT Modelling Methodology*, UN/CEFACT TMWG N090
- 2228 — *Information Technologies – IT Enablement for Widely Used Coded Domains*,  
2229 ISO/IEC 18022
- 2230 — *Information Technologies – Identification and Mapping of Various Categories*  
2231 *of Jurisdictional Domains*, ISO/IEC 18038

## 2232 **11 Disclaimer**

2233 The views and specification expressed in this document are those of the authors and  
2234 are not necessarily those of their employers. The authors and their employers  
2235 specifically disclaim responsibility for any problems arising from correct or incorrect  
2236 implementation or use of this design.

**2237 12 Contact Information**

2238 Team Leader

2239 Name Hartmut Hermes

2240 Company Siemens AG

2241 Street Richard Strauss Strasse 76

2242 City, state, zip/other 81679 Munich

2243 Nation Germany

2244

2245 Phone: (089) 92 21-4564

2246 Email: hartmut.hermes@mch11.siemens.de

2247

2248 Editor

2249 Name Mark Crawford

2250 Company Logistics Management Institute

2251 Street 2000 Corporate Ridge

2252 City, state, zip/other McLean, Virginia 22102

2253 Nation USA

2254

2255 Phone: +01 703 917 7177

2256 Email: mcrawford@lmi.org

2257



## Copyright Statement

Copyright © UN/CEFACT 2002. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to UN/CEFACT except as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by UN/CEFACT or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and UN/CEFACT DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.