



**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  
15

**UN/CEFACT – ebXML Core Components Technical  
Specification**

**30 September 2002  
Version 1.85**



## 16 **1 Status of This Document**

17 This *UN/CEFACT – ebXML Technical Specification* is being developed in accordance  
18 with the UN/CEFACT/TRADE/22 Open Development Process for Technical  
19 Specifications. It has been approved by the United Nations Centre for Trade  
20 Facilitation and Electronic Business (UN/CEFACT) Techniques and Methodology  
21 Group (TMG) for public review as defined in Step 5 of the Open Development  
22 Process.

23 This document contains information to guide in the interpretation or implementation  
24 of ebXML concepts.

25 Distribution of this document is unlimited.

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

27 This version: *UN/CEFACT – ebXML Core Components Technical Specification*,  
28 Version 1.85 of 26 September 2002

29 Previous version: *UN/CEFACT – ebXML Core Components Technical Specification*,  
30 Version 1.8 of 8 February 2002

## 31 **2 UN/CEFACT – ebXML Core Components Technical** 32 **Specification Project Team Participants**

33 We would like to recognise the following for their significant participation to the  
34 development of this Technical Specification.

35	Project Team Leader:	Hartmut Hermes	Siemens
36	Lead Editor:	Mark Crawford	Logistics Management Institute
37	Editing Team	Mike Adcock	APACS
38		Mary Kay Blantz	AIAG
39		Arofan Gregory	CommerceOne
40		Alan Stitzer	Marsh, Inc.
41		Frank Vandamme	SWIFT
42		James Whittle	e Centre

43

44 Contributors:

45		Bernd Boesler	DIN
46		Todd Boyle	NetAccount
47		Kerstien Celis	Seagha c.v.
48		Jean-Luc Champion	Enterprise Integration Partners
49		Marianne Cockle	APACS
50		Scott Colthurst	State Farm
51		Alain Dechamps	CEN/ISSS
52		Eduardo Gutentag	Sun Microsystems
53		Paula Heilig	Worldspan
54		Stig Korsgaard	Danish Bankers Association
55		Melanie McCarthy	General Motors
56		Sue Probert	Commerce One
57		Andreas Schultz	DKV
58		Lisa Seaburg	AEON Consulting
59		Gunther Stuhec	SAP AG
60		Hisanao Sugamata	ECOM-Japan
61		Herbert Thomas	AustriaPro
62		Fred Van Blommestein	Berenschot
63		Nigel Wooden	ACORD

64

### 65 **3 Table of Contents**

66	1	Status of This Document .....	2
67	2	UN/CEFACT – ebXML Core Components Technical Specification	
68		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	Conformance .....	9
75	4.4	Related Documents .....	9
76	4.5	Overview .....	10
77	4.6	Key Concepts .....	11
78	4.6.1	Key Core Component Concepts .....	11
79	4.6.2	Key Business Information Entity Concepts .....	15
80	4.7	Relationship between UN/CEFACT Modelling Methodology and Core	
81		Components .....	20
82	5	Working Process and Methodology .....	21
83	5.1	Overview .....	21
84	5.1.1	Discovery .....	21
85	5.1.2	How to use UN/CEFACT Core Components .....	23
86	5.1.2.1	Core Components and Semantic Interoperability .....	23
87	5.1.2.2	Overall Discovery and Document Design .....	23
88	5.2	Core Components Discovery .....	26
89	5.2.1	Core Component Discovery – Preparation Steps .....	26
90	5.2.2	Core Component Discovery – Search Registry/Repository .....	28
91	5.2.3	Core Component Discovery – Basic and Association Business	
92		Information Entities .....	30
93	5.2.4	Data Types, Property, and Identifying Similarities .....	30
94	5.3	Preparation for Submission .....	30
95	5.3.1	Applying the Naming Convention to a New Item .....	31
96	5.3.2	Preparation for Submitting New Items .....	34
97	5.3.2.1	New Aggregate Core Components .....	34
98	5.3.2.2	New Basic Core Components .....	36
99	5.3.2.3	New Aggregate Business Information Entities which re-use Existing	
100		Aggregate Core Components .....	36
101	5.4	Harmonization .....	37
102	5.5	Technical Assessment and Approval .....	38
103	5.6	Context in the Discovery Process .....	39
104	5.6.1	Context Categories .....	39
105	5.6.2	Guidelines for Analysing Business Information Entities in Context ...	40
106	6	Technical Details .....	44
107	6.1	Core Components, Data Types and Business Information Entities ....	44
108	6.1.1	Core Components .....	44
109	6.1.2	Data Types .....	47
110	6.1.3	Business Information Entities .....	48
111	6.1.4	Naming Convention .....	50
112	6.1.4.1	Core Component Naming Rules .....	50

113	6.1.4.1.1	Core Component Dictionary Information	50
114	6.1.4.1.2	Core Component General Rules	51
115	6.1.4.1.3	Core Component Rules for Definitions	52
116	6.1.4.1.4	Core Component Rules for Dictionary Entry Names	52
117	6.1.4.1.5	Rules for Core Component Business Terms	55
118	6.1.4.2	Rules for Business Information Entities	55
119	6.1.4.2.1	Business Information Entity Dictionary Information	55
120	6.1.4.2.2	Business Information Entity General Rules	56
121	6.1.4.2.3	Business Information Entity Rules for Definitions	56
122	6.1.4.2.4	Rules for Business Information Entity Dictionary Entry Names	57
123	6.1.4.2.5	Rules for Business Information Entity Business Terms	59
124	6.1.4.3	Rules for Data Types	59
125	6.1.4.3.1	Data Type Dictionary Information	59
126	6.1.4.3.2	Data Type General Rules	59
127	6.1.4.3.3	Data Type Rules for Definitions	59
128	6.1.4.3.4	Rules for Data Type Dictionary Entry Names	60
129	6.1.4.3.5	List of Permissible Representation Terms	61
130	6.1.5	Catalogue of Core Components	61
131	6.1.6	Catalogue of Business Information Entities	63
132	6.2	Context	63
133	6.2.1	Overview of Context Specification	63
134	6.2.1.1	Context Categories	64
135	6.2.1.2	Constraint Language	64
136	6.2.1.3	Syntax Binding	65
137	6.2.2	Approved Context Categories	65
138	6.2.2.1	Business Process Context	66
139	6.2.2.2	Product Classification Context	67
140	6.2.2.3	Industry Classification Context	68
141	6.2.2.4	Geopolitical Context	68
142	6.2.2.5	Official Constraints Context	69
143	6.2.2.6	Business Process Role Context	70
144	6.2.2.7	Supporting Role Context	70
145	6.2.2.8	System Capabilities Context	71
146	6.2.3	Context Values	71
147	6.2.4	Core Components Context Constraints Language	71
148	6.2.4.1	Assembly Construct	78
149	6.2.4.2	ContextRules Construct	78
150	6.2.4.3	Output Constraints	79
151	6.2.4.4	Ordering and Application	79
152	7	Technical Details - Core Component Registry/Repository Storage	80
153	7.1	Storing Core Components	80
154	7.1.1	Stored Core Components	81
155	7.1.2	Stored Aggregate Core Components	82
156	7.1.3	Stored Core Component Properties	83
157	7.1.4	Stored Basic Core Component Properties	83
158	7.1.5	Stored Association Core Component Properties	83
159	7.1.6	Stored Basic Core Components	83
160	7.1.7	Stored Association Core Components	84
161	7.1.8	Stored Core Component Types	84
162	7.1.9	Stored Supplementary Components	84

163	7.1.10	Stored Content Components.....	85
164	7.2	Storing Data Types.....	85
165	7.2.1	Stored Data Types.....	86
166	7.2.2	Stored Content Component Restrictions.....	86
167	7.2.3	Stored Supplementary Component Restrictions.....	89
168	7.3	Stored Context.....	89
169	7.3.1	Stored Business Contexts.....	90
170	7.3.2	Stored Classification Schemes.....	90
171	7.3.3	Stored Context Values.....	91
172	7.4	Stored Business Information Entities.....	91
173	7.4.1	Stored Aggregate Business Information Entities.....	92
174	7.4.2	Stored Aggregate Business Information Entities.....	94
175	7.4.3	Stored Business Information Entity Properties.....	94
176	7.4.4	Stored Basic Business Information Entity Properties.....	95
177	7.4.5	Stored Association Core Component Properties.....	95
178	7.4.6	Stored Basic Business Information Entities.....	95
179	7.4.7	Stored Association Business Information Entities.....	95
180	7.5	Core Component Storage Metadata.....	96
181	7.5.1	General Metadata Storage Rules.....	97
182	7.5.2	Management Information.....	98
183	7.5.2.1	Administrative Information.....	98
184	7.5.2.2	Status Information.....	98
185	7.5.2.3	Change History.....	98
186	7.5.2.4	Replacement Information.....	99
187	7.5.3	Content Information.....	99
188	7.5.3.1	Descriptive Information.....	99
189	7.5.3.2	Representation Information.....	100
190	7.5.3.3	Association Information.....	100
191	8	Approved Core Component Type, Content, and Supplementary Components; and Permissible Representation Terms.....	101
192			
193	8.1	Approved Core Component Types.....	101
194	8.2	Approved Core Component Type Content and Supplementary Components.....	103
195			
196	8.3	Permissible Representation Terms.....	105
197	9	Definition of Terms.....	108
198	10	References.....	114
199	11	Disclaimer.....	116
200	12	Contact Information.....	117
201		Copyright Statement.....	118
202			

## 203 4 Introduction

204 This *UN/CEFACT – ebXML Core Components Technical Specification* describes and  
205 specifies a new approach to the well-understood problem of the lack of information  
206 interoperability between applications in the e-business arena. Traditionally, standards for  
207 the exchange of business data have been focused on static message definitions that have  
208 not enabled a sufficient degree of interoperability or flexibility. A more flexible and  
209 interoperable way of standardising business semantics is required. The UN/CEFACT  
210 (United Nations Centre for Trade Facilitation and Electronic Business) – *ebXML Core*  
211 *Component* solution described in this specification presents a methodology for developing  
212 a common set of semantic building blocks that represent the general types of business  
213 data in use today and provides for the creation of new business vocabularies and  
214 restructuring of existing business vocabularies.

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

### 219 4.1 Scope and Focus

220 This *UN/CEFACT – ebXML Core Components Technical Specification* can be employed  
221 wherever business information is being shared or exchanged amongst and between  
222 enterprises, governmental agencies, and/or other organisations in an open and worldwide  
223 environment. The *Core Components User Community* consists of business people,  
224 business document modellers and business data modellers, *Business Process* modellers,  
225 and application developers of different organisations that require interoperability of  
226 business information. This interoperability covers both interactive and batch exchanges of  
227 business data between applications through the use of Internet and Web based  
228 information exchanges as well as traditional Electronic Data Interchange (EDI) systems.

229 This specification will form the basis for standards development work of business  
230 analysts, business users and information technology specialists supplying the content of  
231 and implementing applications that will employ the UN/CEFACT *Core Component*  
232 *Library* (CCL). The CCL will be stored in a UN/CEFACT repository and identified in an  
233 ebXML compliant registry.

234 Due to the evolving nature of the UN/CEFACT *Core Component Library*, the  
235 specification includes material that focuses on the business community doing further  
236 discovery and analysis work. Some of the contents of this specification are not typical of  
237 this type of technical document. However, they are critical for successful adoption and  
238 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>

## 239 4.2 Structure of this Specification

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

- 242 • Section 5: Working Process and Methodology for Business Users—Discovery,  
243 Harmonization, Assessment and How to Use [informative]
- 244 • Section 6: Technical Details—*Core Components* and *Context* [normative]
- 245 • Section 7: Technical Details—Storage and Metadata [normative]
- 246 • Section 8: Technical Details— Permissible *Representation Terms* and  
247 Approved *Core Component Type, Content, and Supplementary Components*  
248 [normative]
- 249 • Section 9: Definition of Terms [normative]

250 Sections 5, 6, 7 and 8 are complementary, but may also be used independently of each  
251 other. Section 5 is informative. A business audience may choose to read through the  
252 working process and methodology section (Section 5) and only reference the Technical  
253 Details (Sections 6, 7 and 8) as needed. Sections 6, 7 and 8 are normative. A technical  
254 audience may choose to focus on the technical details (Sections 6, 7, and 8), referring to  
255 the methodology (Section 5) and example (published as a supplemental document)  
256 sections as appropriate, using the current permissible *Representation Terms* and approved  
257 *Core Component Type, Content, and Supplementary Components* (Section 8) and the  
258 glossary (Section 9).

259 In addition, the UN/CEFACT Forum will prepare supplemental documents that may be  
260 used in conjunction with this *Core Components Technical Specification*. These  
261 supplemental documents will include:

- 262 ◆ *Message Assembly* – expands on the *Assembly* principles and  
263 *Constraints Language* contained in the *Core Components Technical*  
264 *Specification* and provides specific methodology for assembling higher  
265 level *Business Information Entities* for electronic messages.
- 266 ◆ *Core Components Primer* – details how the contents of Sections 5, 6,  
267 and 7 would be used in practice to create a library of *Core Components*  
268 and *Business Information Entities*.
- 269 ◆ *Catalogue of Core Components* – represents the work of various  
270 organisations working in a joint endeavour to develop and publish  
271 semantically correct and meaningful information exchange parcels.

### 272 4.2.1 Notation

273 [Definition] – A formal definition of a term. Definitions are normative.



274 [Example] – A representation of a definition or a rule. Examples are informative.

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

276 [Rn] – Identification of a rule that requires conformance to ensure discovered *Core*  
277 *Components* are properly discovered, named and stored. The value R is a prefix to  
278 categorise the type of rule where R=A for Conformance rule, R=B for *Business*  
279 *Information Entity* rule, R=C for *Core Component* rule, R=D for *Data Type* rule, or R=S  
280 for *Storage* rule; and n (1..n) indicates the sequential number of the rule]. Rules are  
281 normative.

282 *Italics* – All words appearing in italics, when not titles or used for emphasis, are special  
283 terms defined in Section 9.

### 284 **4.3 Conformance**

285 Applications will be considered to be in full conformance with this technical specification  
286 if they comply with the content of normative sections, rules and definitions.

287 [A1] Conformance shall be determined through adherence to the content of normative  
288 sections, rules and definitions.

### 289 **4.4 Related Documents**

290 The following documents provided significant levels of influence in the development of  
291 this document:

- 292 — ebXML Technical Architecture Specification v1.04
- 293 — ebXML Business Process Specification Schema v1.01
- 294 — OASIS/ebXML Registry Information Model v2.0
- 295 — OASIS/ebXML Registry Services Specification v2.0
- 296 — ebXML Requirements Specification v1.06
- 297 — OASIS/ebXML Collaboration-Protocol Profile and Agreement Specification v2.0
- 298 — OASIS/ebXML Message Service Specification v2.0
- 299 — ebXML Technical Report, Business Process and Business Information Analysis  
300 Overview v1.0
- 301 — ebXML Business Process Analysis Worksheets & Guidelines v1.0
- 302 — ebXML Technical Report, E-Commerce Patterns v1.0
- 303 — ebXML Technical Report, Catalog of Common Business Processes v1.0
- 304 — ebXML Technical Report, *Core Component* Overview v1.05
- 305 — ebXML Technical Report, *Core Component* Discovery and Analysis v1.04
- 306 — ebXML Technical Report, *Context* and Re-Usability of *Core Components* v1.04
- 307 — ebXML Technical Report, Guide to the *Core Components* Dictionary v1.04

- 308 — ebXML Technical Report, Naming Convention for *Core Components* v1.04
- 309 — ebXML Technical Report, Document Assembly and *Context* Rules v1.04
- 310 — ebXML Technical Report, Catalogue of *Context Categories* v1.04
- 311 — ebXML Technical Report, *Core Component* Dictionary v1.04
- 312 — ebXML Technical Report, *Core Component* Structure v1.04
- 313 — Information Technology - Metadata registries: Framework for the Specification
- 314 and Standardization of Data Elements, International Standardization Organization,
- 315 ISO 11179-1
- 316 — Information Technology - Metadata registries: Classification of Concepts for the
- 317 Identification of Domains, International Standardization Organization, ISO 11179-
- 318 2
- 319 — Information Technology - Metadata registries: Registry Metamodel, International
- 320 Standardization Organization, ISO 11179-3
- 321 — Information Technology - Metadata registries: Rules and Guidelines for the
- 322 Formulation of Data Definitions, International Standardization Organization, ISO
- 323 11179-4
- 324 — Information Technology - Metadata registries: Naming and Identification
- 325 Principles for Data Elements, International Standardization Organization, ISO
- 326 11179-5
- 327 — Information Technology - Metadata registries: Framework for the Specification
- 328 and Standardization of Data Elements, International Standardization Organization,
- 329 ISO 11179-6

## 330 **4.5 Overview**

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

335 The *Core Components* approach described in this document is more flexible than current  
336 standards in this area because the semantic standardisation is done in a syntax-neutral  
337 fashion. Using *Core Components* as part of the ebXML framework will help to ensure  
338 that two trading partners using different syntaxes [e.g. Extensible Markup Language  
339 (XML) and United Nations/EDI for Administration, Commerce, and Transport  
340 (UN/EDIFACT)] are using business semantics in the same way on condition that both  
341 syntaxes have been based on the same *Core Components*. This enables clean mapping  
342 between disparate message definitions across syntaxes, industry and regional boundaries.

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

## 350 4.6 Key Concepts

351 The *Core Components Technical Specification* key concepts cover two focus areas—*Core*  
352 *Components* and *Business Information Entities*. Each of these focus areas is discussed in  
353 the following subsections. In each subsection, concepts are introduced, followed by a  
354 normative definition and where appropriate an example for each.

### 355 4.6.1 Key Core Component Concepts

356 The central concept of this specification is the *Core Component*. The *Core Component* is  
357 a semantic building block which is used as a basis to construct all electronic business  
358 messages.

359 [Definition] *Core Component* (CC)

360 A building block for the creation of a semantically correct and meaningful information  
361 exchange package. It contains only the information pieces necessary to describe a specific  
362 concept.

363 There are four different categories of *Core Components*: *Basic Core Component*,  
364 *Association Core Component*, *Core Component Type* and *Aggregate Core Component*.  
365 The following definitions explain each of these:

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

367 A *Core Component* which constitutes a singular business characteristic of a specific  
368 *Aggregate Core Component* that represents an *Object Class*. It has a unique business  
369 semantic definition. A *Basic Core Component* represents a *Basic Core Component*  
370 *Property* and is therefore of a *Data Type*, which defines its set of values. *Basic Core*  
371 *Components* function as the *Properties* of *Aggregate Core Components*.

372

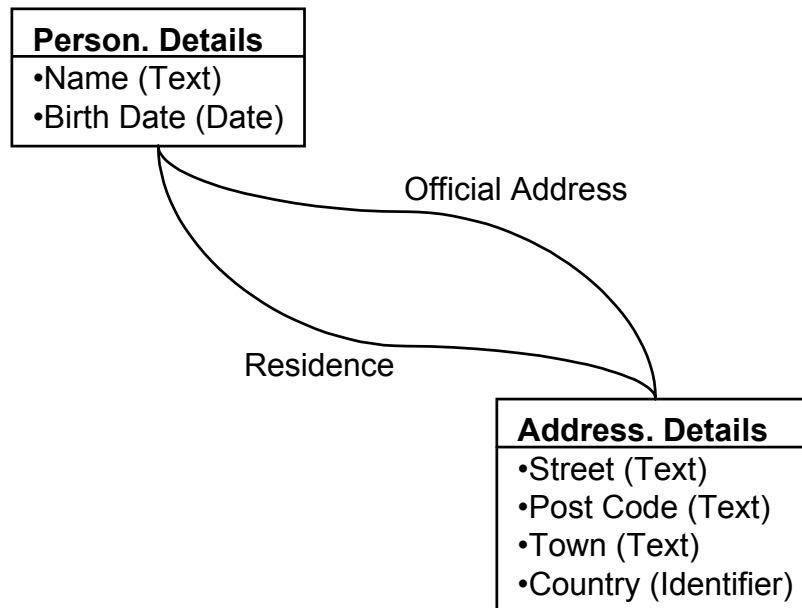
373 [Definition] *Association Core Component* (ASCC)

374 A *Core Component* which constitutes a complex business characteristic of a specific  
375 *Aggregate Core Component* that represents an *Object Class*. It has a unique business  
376 semantic definition. An *Association Core Component* represents an *Association Core*  
377 *Component Property* and is associated to an *Aggregate Core Component*, which describes  
378 its structure.

379

380 [Example] *Association Core Component*

381



382

383

384 The example shows two *Aggregate Core Components*, **Person. Details** and  
 385 **Address. Details**. Each *Aggregate Core Component* has a number of *Properties* (i.e.  
 386 business characteristics). **Person. Details** has four *Properties*, namely Name, Birth  
 387 Date, Residence and Official Address. **Address. Details** also has four *Properties*,  
 388 namely Street, Post Code, Town and Country.

389 Most of these *Properties* are *Basic Core Component Properties*. These *Properties*  
 390 represent a singular business characteristic and their set of allowed values is defined by a  
 391 *Data Type*. Name, Street, Post Code and Town are of the *Data Type* Text, Birth Date is of  
 392 the *Data Type* Date and Country is of the *Data Type* Identifier.

393 The other *Properties* are *Association Core Component Properties*. They represent  
 394 complex business characteristics and their structure is therefore defined by another  
 395 *Aggregate Core Component*. Residence and Official Address are both *Association Core*  
 396 *Component Properties* and their structure is described by “Address. Details”.

397 This example will therefore result in following set of *Core Components*:

398 • **Person. Details** (*Aggregate Core Component*)

399 • **Person. Name. Text** (*Basic Core Component*)

400 [Example] *Association Core Component* (Continued)

401 • **Person. Birth. Date** (*Basic Core Component*)

402 • **Person. Residence. Address** (*Association Core Component*)

403 • **Person. Official. Address** (*Association Core Component*)

404 • **Address. Details** (*Aggregate Core Component*)

405 • **Address. Street. Text** (*Basic Core Component*)

406 • **Address. Post Code. Text** (*Basic Core Component*)

407 • **Address. Town. Text** (*Basic Core Component*)

408 • **Address. Country. Identifier** (*Basic Core Component*)

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

410 A *Core Component*, which consists of one and only one *Content Component*, that carries  
411 the actual content plus one or more *Supplementary Components* giving an essential extra  
412 definition to the *Content Component*. *Core Component Types* do not have business  
413 semantics.

414

415 [Example] *Core Component Types*

416 For a *Core Component Type* of Amount. Type, the *Content Component* carries the value  
417 of 12. This value has no meaning on its own. But 12 Kilometres or 12 Euro, where  
418 Kilometres or Euro are the *Supplementary Component* that gives essential extra definition  
419 to the *Content Component*, do have meaning.

420

421 [Definition] *Aggregate Core Component*

422 A collection of related pieces of business information that together convey a distinct  
423 business meaning, independent of any specific *Business Context*. Expressed in modelling  
424 terms, it is the representation of an *Object Class*, independent of any specific *Business*  
425 *Context*.

426 [Example] – *Aggregate Core Component*  
427 Aggregate: **Financial Account. Details**<sup>2</sup>  
428 Definition: A service through a bank or other organisation through which funds are held  
429 on behalf of a client or goods or services are supplied on credit.  
430 *Basic Core Components:*  
431       • **Financial Account. Identifier**  
432       • **Financial Account. Name**  
433       • **Financial Account. Country. Identifier**  
434       • **Financial Account. Product Type. Identifier**  
435       • **Financial Account. Nickname. Name**

436 *Core Components (and Business Information Entities) have Properties that are defined by*  
437 *Data Types.*

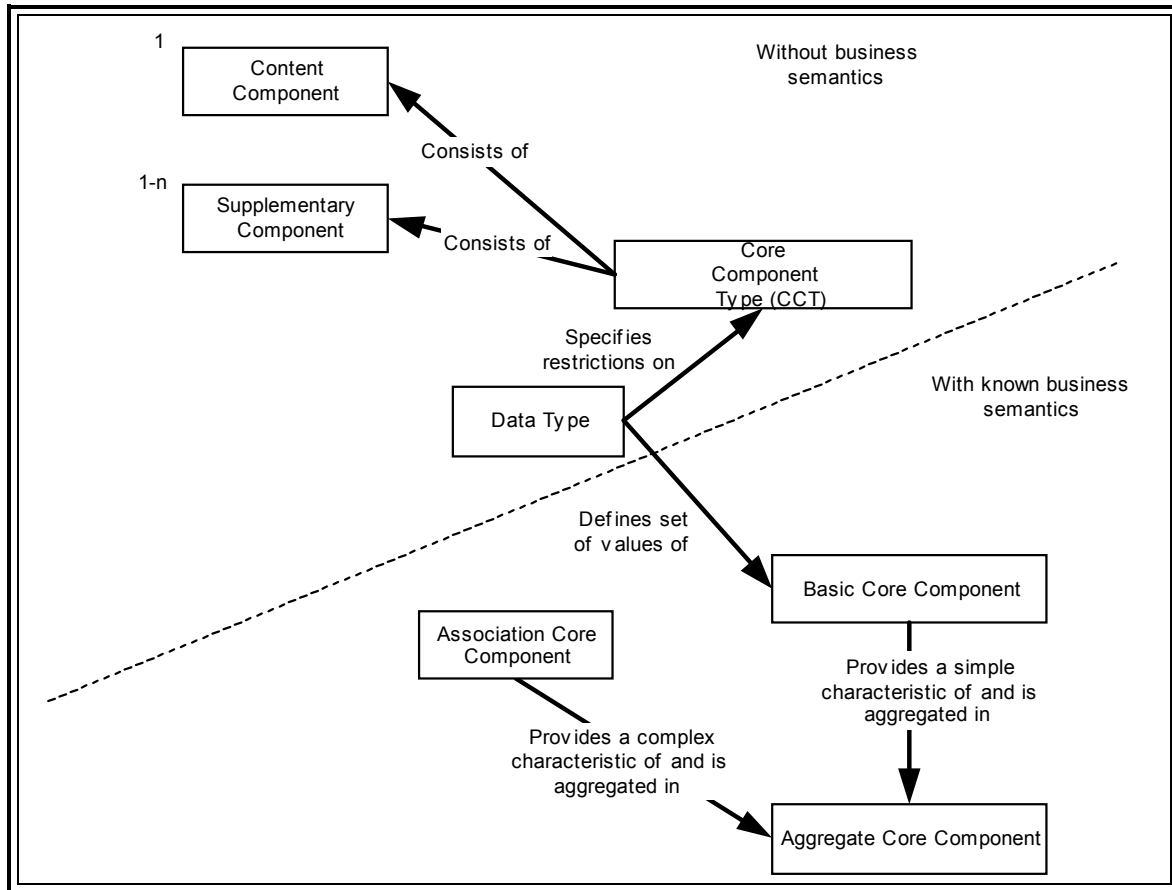
438 A *Data Type* represents the full range of values that shall be used for the representation of  
439 a particular *Core Component Property*. A *Data Type* must be based on one of the *Core*  
440 *Component Types*, but may include restrictions of the set of values of that *Core*  
441 *Component Type's Content Component* and/or *Supplementary Component(s)*.

442 [Definition] – *Data Type*  
443 Defines the set of valid values that can be used for a particular *Basic Core Component*  
444 *Property* or *Basic Business Information Entity Property*. It is defined by specifying  
445 restrictions on the *Core Component Type* that forms the basis of the *Data Type*.

446 The simple diagram in Figure 4-1 shows the relationships between the various *Core*  
447 *Component* elements.

---

<sup>2</sup> See section 6.1.4 for detailed rules for developing Core Component names.

448 **Figure 4-1. Core Component Overview**449  
450451 **4.6.2 Key Business Information Entity Concepts**

452 The key differentiator between *Core Components* and *Business Information Entities* is the  
 453 concept of *Business Context*. *Business Context* is a mechanism for qualifying and refining  
 454 *Core Components* according to their use under particular business circumstances. Once  
 455 *Business Contexts* are identified, *Core Components* can be differentiated to take into  
 456 account any necessary qualification and refinement needed to support the use of the *Core*  
 457 *Component* in the given *Business Context*. The *Business Process* definition provides a  
 458 high level description of the use of a message and its contents.<sup>3</sup>

459 [Definition] *Business Context*

460 The formal description of a specific business circumstance as identified by the values of a  
 461 set of *Context Categories*, allowing different business circumstances to be uniquely  
 462 distinguished.

<sup>3</sup> The *Core Components*' *Context* mechanism provides the more detailed linkage between specific business data and the exact circumstances of its business use.

463 When a *Core Component* is used in a real business circumstance it serves as the basis of a  
464 *Business Information Entity*. The *Business Information Entity* is the result of using a *Core*  
465 *Component* within a specific *Business Context*.

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

467 A piece of business data or a group of pieces of business data with a unique business  
468 semantic definition. A *Business Information Entity* can be a *Basic Business Information*  
469 *Entity (BBIE)*, an *Association Business Information Entity (ASBIE)*, or an *Aggregate*  
470 *Business Information Entity (ABIE)*.

471 A specific relationship exists between *Core Components* and *Business Information*  
472 *Entities*. *Core Components* and *Business Information Entities* are complementary in many  
473 respects. *Core Components* are intended to be the linchpin for creating interoperable  
474 *Business Process* models and business documents using a *Controlled Vocabulary*.

475 There are three different categories of Business Information Entities: *Basic Business*  
476 *Information Entity*, *Association Business Information Entity*, and *Aggregate Business*  
477 *Information Entity*. The most primitive of these is the *Basic Business Information Entity*.  
478 A *Basic Business Information Entity* is a *Basic Core Component* used in a specific  
479 *Business Context*.

480 [Definition] *Basic Business Information Entity (BBIE)*

481 A *Business Information Entity* that represents a singular business characteristic of a  
482 specific *Object Class* in a specific *Business Context*. It has a unique business semantic  
483 definition. A *Basic Business Information Entity* represents a *Basic Business Information*  
484 *Entity Property* and is therefore linked to a *Data Type*, which describes its values. A *Basic*  
485 *Business Information Entity* is derived from a *Basic Core Component*.

486 An *Association Business Information Entity* is an *Aggregate Business Information Entity*  
487 serving as the *Property* of another *Aggregate Business Information Entity*. It is based on  
488 an *Association Core Component*, but exists in a *Business Context*.

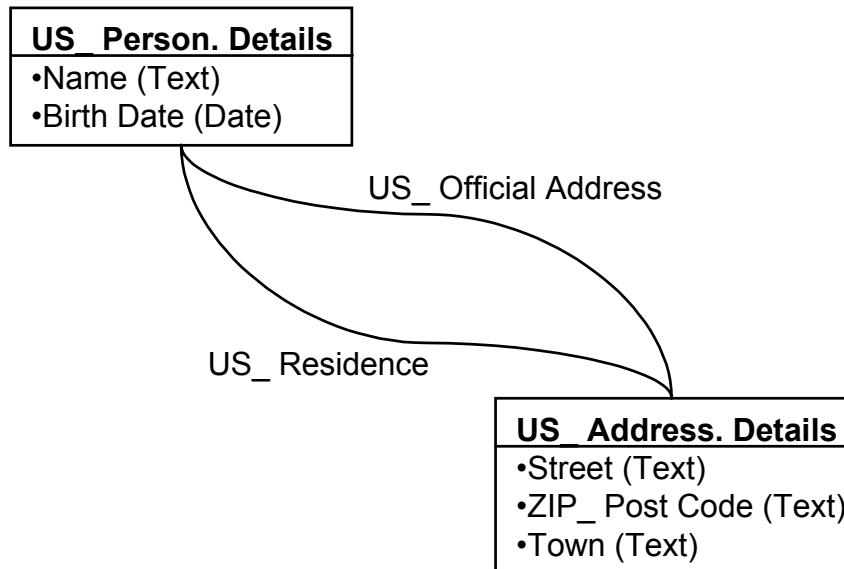
489 [Definition] *Association Business Information Entity (ASBIE)*

490 A *Business Information Entity* that represents a complex business characteristic of a  
491 specific *Object Class* in a specific *Business Context*. It has a unique business semantic  
492 definition. An *Association Business Information Entity* represents an *Association Business*  
493 *Information Entity Property* and is therefore associated to an *Aggregate Business*  
494 *Information Entity*, which describes its structure. An *Association Business Information*  
495 *Entity* is derived from an *Association Core Component*.

496



497 [Example] *Association Business Information Entity*



498

499 The example shows two *Aggregate Business Information Entities*, **US\_ Person.**  
 500 **Details** and **US\_ Address. Details**. Each *Aggregate Business Information Entity*  
 501 has a number of *Properties* (i.e. business characteristics). **US\_ Person. Details** has  
 502 four *Properties*, namely Name, Birth Date, US\_ Residence and US\_ Official Address.  
 503 **US\_ Address. Details** has three *Properties*, namely Street, ZIP\_ Post Code and  
 504 Town.

505 Most of these *Properties* are *Basic Business Information Entity Properties*. They  
 506 represent a singular business characteristic and their set of allowed values is defined by a  
 507 *Data Type*. Name, Street, ZIP\_ Post Code and Town are of the *Data Type* Text and Birth  
 508 Date is of the *Data Type* Date.

509 The other *Properties* are *Association Business Information Entity Properties*. They  
 510 represent complex business characteristics and their structure is therefore defined by  
 511 another *Aggregate Business Information Entity*. US\_ Residence and US\_ Official Address  
 512 are both *Association Business Information Entity Properties* and their structure is  
 513 described by “**US\_ Address. Details**”.

514 This example will therefore result in following set of *Business Information Entities*:

- 515 • **US\_ Person. Details** (*Aggregate Business Information Entity*)
- 516 • **US\_ Person. Name. Text** (*Basic Business Information Entity*)

- 517 [Example] *Association Business Information Entity* (Continued)
- 518 • **US\_ Person. Birth. Date** (*Basic Business Information Entity*)
  - 519 • **US\_ Person. US\_ Residence. US\_ Address** (*Association Business*  
520 *Information*  
521 *Entity*)
  - 522 • **US\_ Person. US\_ Official. US\_ Address** (*Association Business*  
523 *Information Entity*)
  - 524 • **US\_ Address. Details** (*Aggregate Business Information Entity*)
  - 525 • **US\_ Address. Street. Text** (*Basic Business Information Entity*)
  - 526 • **US\_ Address. ZIP\_ Post Code. Text** (*Basic Business Information Entity*)
  - 527 • **US\_ Address. Town. Text** (*Basic Business Information Entity*)

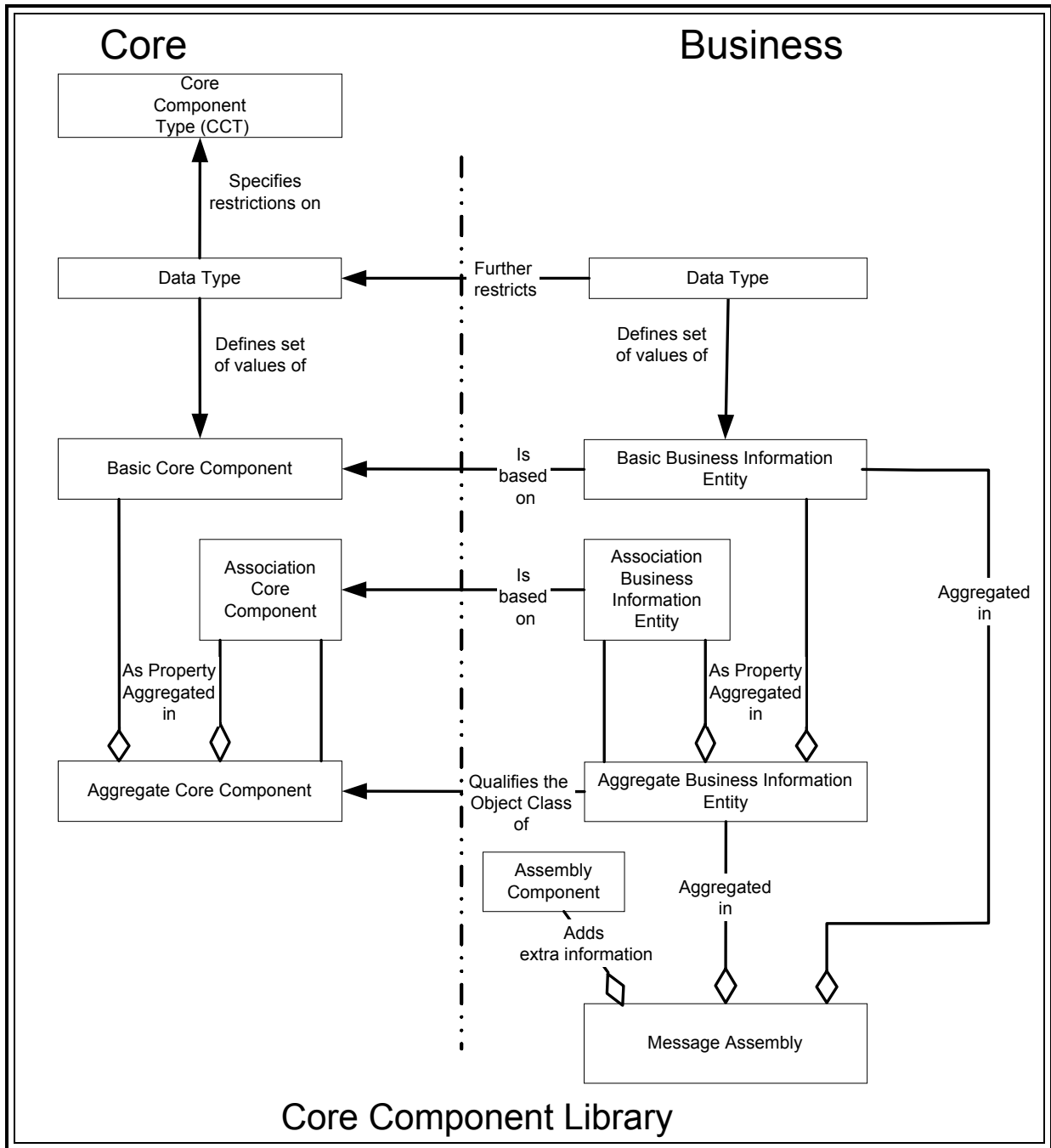
528 An *Aggregate Business Information Entity* is a piece of business data or a group of pieces  
529 of business data with a unique business semantic definition in a specific *Business Context*.

530 [Definition] *Aggregate Business Information Entity*

531 A collection of related pieces of business information that together convey a distinct  
532 business meaning in a specific *Business Context*. Expressed in modelling terms, it is the  
533 representation of an *Object Class*, in a specific *Business Context*.

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

536 **Figure 4-2. Relationships between Core Components and Business Information**  
 537 **Entities**



538  
539

540

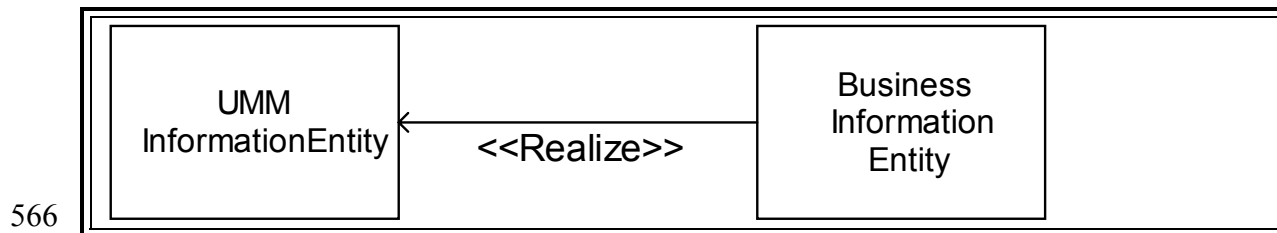
[Note]

541 The term *Core Component* is used as a generic term that encompasses *Basic Core*  
 542 *Components*, *Association Core Components*, *Aggregate Core Components*, and their  
 543 associated *Core Component Types*. Equally the term *Business Information Entity* is used  
 544 as a generic term encompassing *Basic Business Information Entities*, *Association*  
 545 *Business Information Entities*, and *Aggregate Business Information Entities*.

## 546 4.7 Relationship between UN/CEFACT Modelling Methodology 547 and Core Components

548 UN/CEFACT has developed the *UN/CEFACT Modelling Methodology* (UMM). UMM  
549 describes a Unified Modeling Language (UML) based modelling approach to develop  
550 *UMM InformationEntities*.<sup>4</sup> Within UN/CEFACT standards efforts, the *Core Component*  
551 framework of *Core Components* and *Business Information Entities* prescribes the  
552 mechanism for discovery, normalisation, *Context* specialisation, and structure of *UMM*  
553 *InformationEntities*. The *Aggregate Business Information Entity-Basic Business*  
554 *Information Entity* framework provides the structure for components of the body of the  
555 business document. The *Core Component-Business Information Entity-Context* mapping  
556 framework provides the basis for mapping *UMM InformationEntity* realisations to  
557 business entities. The *Business Information Entity to Core Component* relationship  
558 provides the dictionary reference as specified in the information model abstract syntax.  
559 The UN/CEFACT *Core Component Library* is an implementation of the *UN/CEFACT*  
560 *Modelling Methodology* dictionary concept. The *Basic Core Component* is the realization  
561 of a non-aggregate *UMM InformationEntity* and provides the mapping to *Data Types*. The  
562 relationship between the *Core Component Framework* and the *UMM InformationEntity* is  
563 illustrated in Figure 4-3.

564 **Figure 4-3. Relationship between Core Component Framework and UMM**  
565 **InformationEntity**



<sup>4</sup> The UN/CEFACT Modelling Methodology (UMM) is a methodology for Business Process and information modelling that is based on the Object Management Group's Unified Modeling Language.

## 567 **5 Working Process and Methodology**

568 This section identifies aspects of *Core Component* working processes and  
569 methodologies for use. It includes an overview of the discovery and usage  
570 characteristics of *Core Components*. In addition, it includes detailed recommendations  
571 for conducting discovery, storage, approval, and application of *Context*.

### 572 **5.1 Overview**

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

#### 577 **5.1.1 Discovery**

578 A *Business Process* should be modelled using a standard approach. UN/CEFACT  
579 requires the *UN/CEFACT Modelling Methodology* (UMM) as the approach.<sup>5</sup> One of  
580 the results is a model, including a class diagram, which shows the business  
581 information and its inter-relationships. *Business Information Entities* can be identified  
582 from the *ebXML Business Process Analysis Worksheets and Guidelines*<sup>6</sup> that provide  
583 a simplified modelling approach.

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

589 In order to improve interoperability across *Business Contexts*, *Business Information*  
590 *Entities* must be based on a basic library of clearly defined semantic constructs to help  
591 ensure that they will inter-operate. This library must include a set of globally agreed  
592 semantic definitions such as those that will be contained in the *UN/CEFACT Core*  
593 *Components Library*.

594 A *Business Information Entity* is a *Core Component* used in a specific *Business*  
595 *Context* and given its own unique name. As *Basic Core Components* are single pieces  
596 of business information, when they are used directly in specific *Business Contexts* the  
597 structure (components) does not change, but values may be restricted.

---

<sup>5</sup> The UN/CEFACT Modelling Methodology (UMM) is a methodology for Business Process and information modelling that is based on the Unified Modeling Language.

<sup>6</sup> The ebXML Business Process Analysis Worksheets & Guidelines can be found at <http://www.ebxml.org/>

598

[Example]

599

An invoicing *Business Process* uses a piece of information such as

600

**Invoice. VAT\_ Tax. Amount.\* Invoice. VAT\_ Tax. Amount** is a *Basic Business Information Entity* that is based on the *Basic Core Component* of **Invoice.**

601

602

**Tax. Amount.** The invoicing *Business Process* is using **Invoice. Tax. Amount**

603

in a specific *Business Context* where the *Business Process Context* = *Purchasing*, and

604

the *Geopolitical Context* = *EU*. Therefore the application of *Context* adds a

605

specialised definition, but in all other respects the *Basic Business Information Entity* is

606

the same as the associated *Core Component* of **Invoice. Tax. Amount**, i.e. it has

607

the same structure and data type.

608

\*In accordance with rule [B17], VAT would be defined as Value Added Tax in the

609

definition for the BBIE **Invoice. VAT\_ Tax. Amount**.

610

611

Just as each *Basic Business Information Entity* must ultimately be based on a *Basic*

612

*Core Component*, each *Aggregate Business Information Entity* must ultimately be

613

based on an existing *Aggregate Core Component*. The underlying *Aggregate Core*

614

*Component* identifies the generic, standard definition of business information that is

615

being used in the *Aggregate Business Information Entity*. The definition of the

616

*Aggregate Business Information Entity* is based upon the generic description, being

617

then modified and enhanced to be specific to the *Business Context* in which the

618

*Aggregate Business Information Entity* is used. An *Aggregate Business Information*

619

*Entity* is thus directly tied to a specific *Business Process*, or to a *Business Context*.

620

(See Section 5.6 for a fuller understanding of *Context*.)

621

When an *Aggregate Business Information Entity* has a complex *Property*, then that

622

*Property* is represented by an *Association Business Information Entity*. *Association*

623

*Business Information Entities* are specific to their *Business Context*, and relate to

624

*Association Core Components*. This relationship is the same as the relationship

625

between *Aggregate Business Information Entities* and *Aggregate Core Components*

626

and between *Basic Business Information Entities* and *Basic Core Components*. (See

627

Figure 6-2 for a fuller understanding of this concept.)

628

An important aspect of information interoperability is that each *Business Information*

629

*Entity* is based upon a *Core Component* structure and associated semantic definitions

630

derived from the *Core Component Library*. The structure and definition of the

631

*Business Information Entity* may be a refined and/or restricted version of the structure

632

and definition of the *Core Component* upon which it is based.

633

The following section describes the procedures by which the content of the

634

UN/CEFACT ebXML compliant *Core Component Library* may be developed and

635

maintained.

## 636 5.1.2 How to use UN/CEFACT Core Components

637 This section provides a procedure for the technical user who wants to understand how  
638 to use *Core Components*. It assumes the user is dealing with an established set of  
639 *Core Components*, *Context Categories* and metadata/storage. The established set of  
640 *Core Components* being used should be based on those discovered, harmonized, and  
641 published by recognized standards groups. It is further assumed that the recognized  
642 standards group(s) and other business association group(s) have also made available  
643 sets of *Business Information Entities* for use in a published set of *Business Processes*.

### 644 5.1.2.1 Core Components and Semantic Interoperability

645 Today, the e-business community generally agrees on the definition of a standard  
646 message structure expressed as an UN/EDIFACT Message Implementation Guide  
647 (MIG), an XML schema, or similar syntax specific representation. UN/CEFACT will  
648 produce standards based representations of these artefacts for implementation.<sup>7</sup>

649 Under the *Core Components* concept, defining and storing *Core Components* and  
650 associated *Context* mechanisms occur prior to the creation of a MIG or an XML  
651 schema. In this manner, the focus of the user changes from examining the MIG or  
652 XML schema, and moves to an examination of the semantic models. Accordingly,  
653 interoperability between syntaxes no longer depends on analysing specific instances,  
654 but naturally occurs during the *Business Process* model definition phase.

### 655 5.1.2.2 Overall Discovery and Document Design

656 Overall discovery and document design can be thought of as a series of steps that  
657 starts with determining the availability of existing *Business Process* definitions and  
658 ultimately results in standard business documents. Figure 5-1 illustrates this process.  
659 Specific steps to be followed are further described below.  
660

661 Step 1: Search the registry/repository<sup>8</sup> – A search should be made in the registry to  
662 find the *Business Process* that meets the business requirement.

663 Step 1a: If no existing *Business Process* is found to be appropriate, then the new  
664 *Business Process* should be modelled using *UN/CEFACT Modelling*  
665 *Methodology* and submitted to the registry.

666 Step 1b: Conduct a thorough analysis of the business information requirements by  
667 following the *Core Component* discovery steps (Section 5.2)

---

<sup>7</sup> The term XML schema includes XML Schema as defined in World Wide Web Consortium Extensible Markup Language Version 1.0, XML Document Type Definitions, Schematron, SOX, Relax NG, ASN.1, XDR, or any other notation that specifies the form and information content of an XML document.

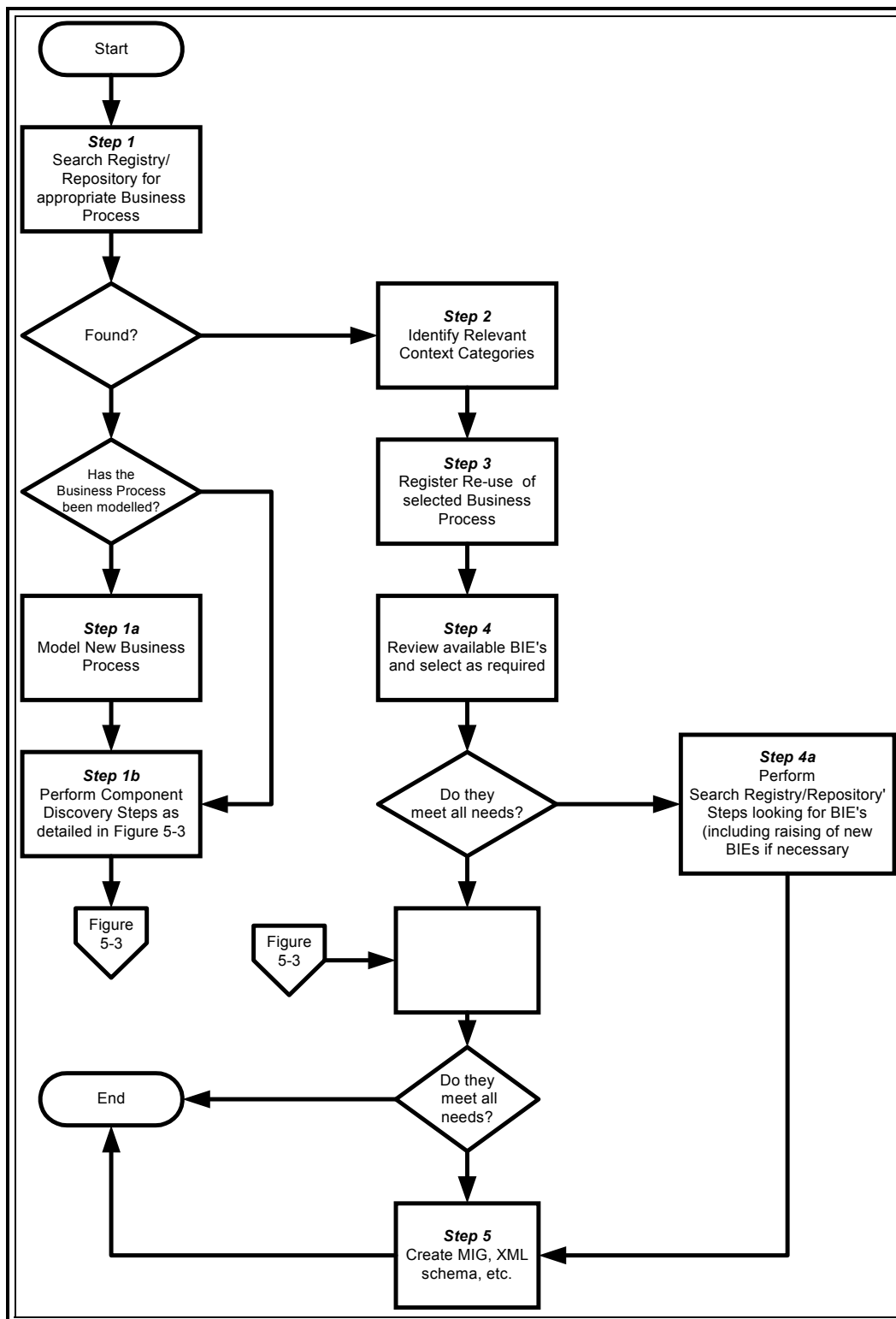
<sup>8</sup> See the list of referred documents for explanation of 'registry/repository' within the ebXML architecture.

- 668 Step 2: Identify relevant *Context Categories* – Access the registry interface and  
669 identify the relevant *Context Categories* of the selected *Business Process* by  
670 determining the following *Context Categories* (See Section 6.2.2):
- 671 • *Business Process Context* – Identify the interaction between trading  
672 partners to achieve a given business objective.
  - 673 • *Product Classification Context* – Determine the goods or services  
674 concerned in the collaboration.
  - 675 • *Industry Classification Context* – Determine the relevant trading partner  
676 industries.
  - 677 • *Geopolitical Context* – Determine where the *Business Process* is to be  
678 conducted. Determine if the *Business Process* crosses regional, national,  
679 or international boundaries.
  - 680 • *Official Constraints Context* – Determine any legal restrictions or  
681 requirements on this *Business Process*.
  - 682 • *Business Process Role Context* – Identify the roles played by the trading  
683 partners. These can be derived from the *Business Process*.
  - 684 • *Supporting Role Context* – Determine what other significant parties will be  
685 using the data in the messages. Determine their role in the overall  
686 process.
  - 687 • *System Capabilities Context* – Determine any major restrictions derived  
688 from system, a class of systems or standard in the business situation.  
689 Identify the type of system.
- 690 The registry will provide a list of pre-defined *Business Information Entities*  
691 that are available to the selected *Business Process*, and which meet the  
692 *Context* criteria specified. These will come with identified relationships to  
693 the *Core Components* upon which they are based, and the *Context*  
694 rules/values that fully qualify them. The registry should also return partial  
695 matches with an indication of how closely they match the specified *Context*.
- 696 Step 3: Register re-use of the selected *Business Process* in the set of *Contexts* in  
697 which it is being used. Registration of each re-use ensures the gradual  
698 development of a library of re-uses that will be available to the widening  
699 user base.
- 700 Step 4: Review the available *Business Information Entities* and select the  
701 appropriate subset for use that meets the needs of the *Business Process*  
702 requirement that is being developed.
- 703 Step 4a: If the *Business Information Entities* available for the specific *Business*  
704 *Process* do not address all of the data requirements, the registry of all  
705 *Business Information Entities* should be searched to see if the appropriate  
706 *Business Information Entities* already exist. The procedure for this is



707 described under Search Registry/Repository (Section 5.2), which includes  
 708 the steps to raise any new *Business Information Entities*, required because no  
 709 appropriate *Business Information Entities* can be found.

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



711

712 Step 4b: If all required *Business Information Entities* are already available, review the  
713 available MIG, XML schema, and/or other syntax-specific message  
714 description and select the appropriate one(s) for use that meet the technical  
715 implementation/solution requirements identified. If no appropriate technical  
716 implementation/solution is already available, continue with Step 5 to create  
717 new ones.

718 Step 5: Create MIG, XML schema, etc. – The resulting semantic model (the set of  
719 *Business Information Entities*) is manually or programmatically rendered  
720 into a syntax-specific message description. The resulting MIG, XML schema  
721 or other syntax specific message description is submitted to the registry  
722 where it is associated with the *Business Information Entities* it represents.

723 [Note]

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

## 731 **5.2 Core Components Discovery**

732 The steps in *Core Component* discovery are preparation and search for candidate  
733 common information building blocks. In order to properly define the *UN/CEFACT*  
734 *Core Component Library*, domain or project groups must follow the prescribed  
735 preparation and search steps as outlined in the following subsections. See the *Core*  
736 *Components Primer* supplemental document for a detailed end-to-end example of  
737 discovering *Core Components*.

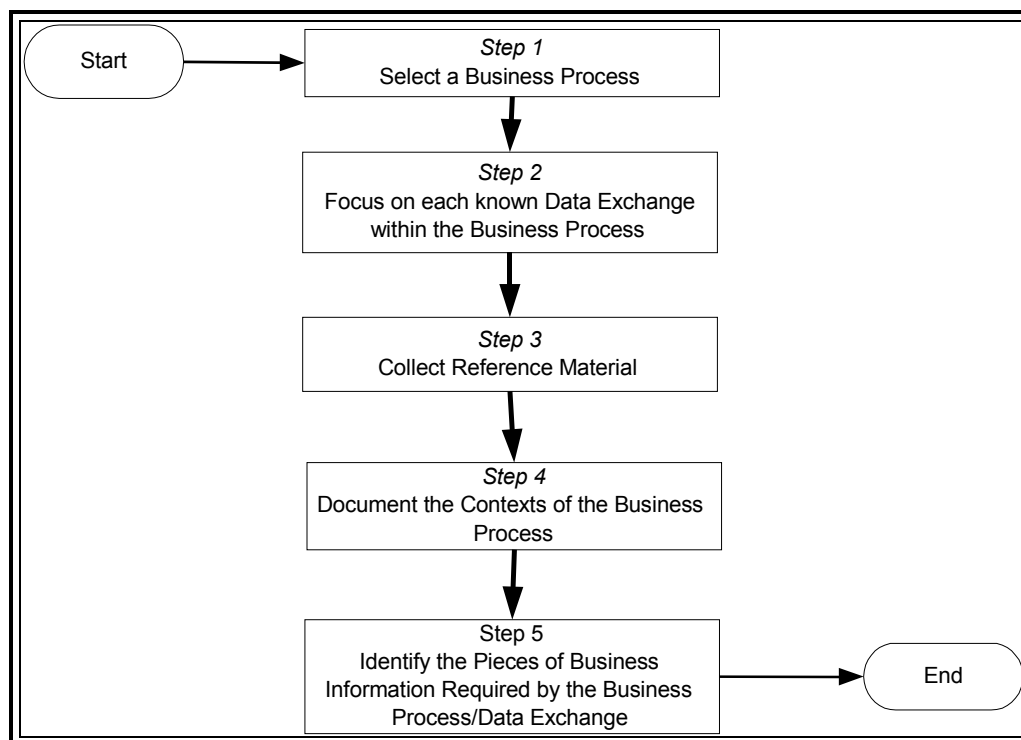
### 738 **5.2.1 Core Component Discovery – Preparation Steps**

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

744 Step 1. Select a *Business Process* that provides a wide range of business information  
745 content within the domain being addressed. The broader the range of the  
746 chosen *Business Process*, the greater the opportunity to discover candidate  
747 *Core Components*. (e.g. *Make a Payment, Place an Order, Issue an Invoice*)

- 748 Step 2. Focus on each known data exchange within the *Business Process* that  
 749 contains key business information (e.g. *Payment Order, Purchase Order,*  
 750 *Invoice*).
- 751 Step 3. Collect all the business information and associated details that are relevant to  
 752 the chosen business exchange for the previously identified *Business Process*.  
 753 Use a cross section of Message Implementation Guides, RosettaNet Partner  
 754 Interface Process (PIP), Business Process Information Models (BPIMs) or  
 755 similar domain-specific artefacts as sources of information about the  
 756 business exchange.
- 757 Step 4. Document the *Context(s)* of the *Business Process* being analysed. Identify  
 758 what is applicable for each category of *Context*, i.e. whether it is none, in all  
 759 *Contexts*, or one or multiple specific *Context* value(s). (See Section 5.6 for a  
 760 more detailed explanation of how to determine *Context*).

761 **Figure 5-2 Preparation Steps**



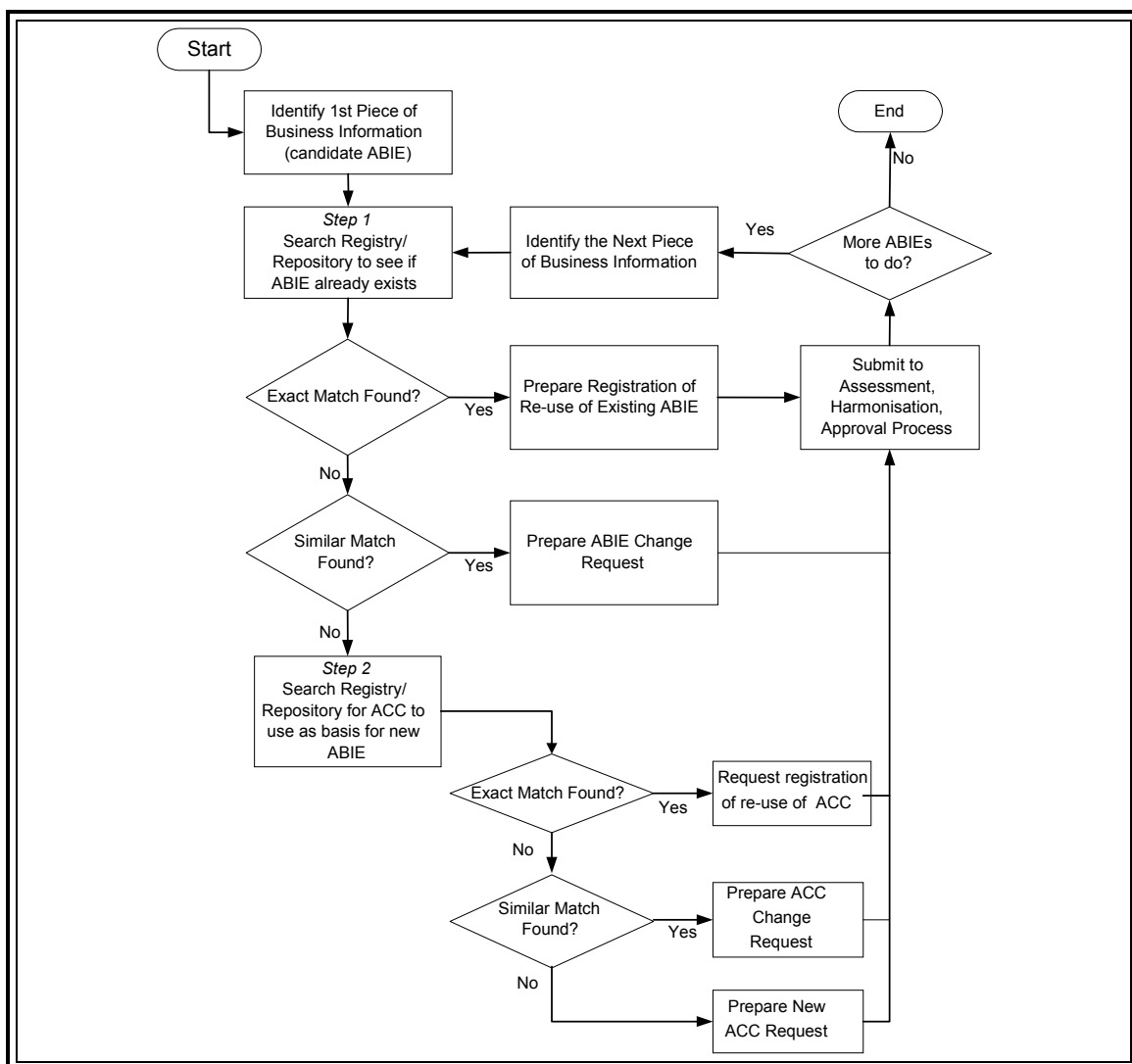
- 762
- 763 Step 5. Compile a list of the pieces of information required for the *Business Process*.
- 764
- 765
- 766
- If starting from a model (UN/CEFACT recommends UMM models of *Business Processes*), identify the objects (*Aggregate Business Information Entities*) that are needed.
- 767
- If not starting from a model, collect the pieces of information into object-like groups (*Aggregate Business Information Entities*). It is important to recognise and avoid pieces of information that are purely used for legacy system or syntax purposes.
- 768
- 769
- 770

- 771           • For each *Aggregate Business Information Entity*, capture its unique  
 772 semantic definition, any *Business Terms* by which it is commonly known,  
 773 and any other information identified in the previous steps.
- 774           • At this point of discovery, and before searching the registry/repository,  
 775 these are candidate *Aggregate Business Information Entities*.

## 776 5.2.2 Core Component Discovery – Search Registry/Repository

777 Having identified the need for a number of candidate *Aggregate Business Information*  
 778 *Entities* in the preparation Step 5 identified in Section 5.2.1 above, repeat the  
 779 following steps for each *Aggregate Business Information Entity*, as shown in Figure 5-  
 780 3.

781 **Figure 5-3 Search Steps**



783

[Note]

784

Exact is 'a precise match in all details'.

785

Similar is 'of the same kind without being identical'.

786

Employment of common sense and good judgement is essential in making these

787

determinations.

788

Step 1 It is recommended to start with *Aggregate Business Information Entities* at the highest level of aggregation. Search the *Catalogue of Business Information Entities* for an existing *Aggregate Business Information Entity* that has the same definition.

789

790

791

792

- Exact Match: If there is an *Aggregate Business Information Entity* with a definition and composition that meets the business need, register the re-use including *Business Context* and any *Business Terms*. (Go to next *Aggregate Business Information Entity*)

793

794

795

796

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

797

798

799

800

801

802

803

804

805

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

806

807

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

808

809

810

811

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

812

813

814

815

816

- If there is an *Aggregate Core Component* with a definition and structure that potentially could be modified to meet the business need, prepare an *Aggregate Core Component* change request for submission to the harmonization and approval process. Include the re-use of the *Aggregate Core Component* as an *Aggregate Business Information Entity*, including the definition and name created according to the naming convention, and the *Business Context* in which it is used. (Go to next *Aggregate Business Information Entity*)

817

818

819

820

821

822

823

- 824           • If there is not an *Aggregate Core Component* with a suitable definition  
825           and structure, prepare a new *Aggregate Core Component* request for  
826           submission to the harmonization and approval process. Include the re-use  
827           of the *Aggregate Core Component* as an *Aggregate Business Information*  
828           *Entity*, including the definition and name created according to the naming  
829           convention, and the *Business Context* in which it is used. (Go to next  
830           *Aggregate Business Information Entity*)

### 831   **5.2.3 Core Component Discovery – Basic and Association Business** 832   **Information Entities**

833   This procedure is exactly the same as that described in Section 5.2.2, except that the  
834   reader should read *Basic* or *Association Business Information Entity* for *Aggregate*  
835   *Business Information Entity* and *Basic* or *Association Core Component* for *Aggregate*  
836   *Core Component*.

### 837   **5.2.4 Data Types, Property, and Identifying Similarities**

838   When looking for similarities between existing *Business Information Entities* and  
839   *Core Components*, and those *Business Information Entities* that are required but not  
840   present, the user should consider *Property* and *Data Types*. If a *Core Component* is  
841   found that has a very similar *Property* to an existing *Core Component*, but a different  
842   *Object Class*, then that *Property* should be used for the new *Basic Business*  
843   *Information Entity* that is to be created where the basic structure and semantics align.  
844   The key to the similarities of *Property* is that they share a *Data type*. If a new *Core*  
845   *Component* is requested, these identified similarities at the level of *Property* should  
846   also be identified.  
847

848   [Example]

849   There is an existing *Basic Business Information Entity* for **Total. Tax. Amount**,  
850   based on a corresponding *Basic Core Component*. The user needs a *Basic Business*  
851   *Information Entity* for **Subtotal. Tax. Amount**, but after searching the  
852   registry/repository determines this does not exist. Because both the existing *Basic*  
853   *Business Information Entity* of **Total. Tax. Amount** and the desired *Basic*  
854   *Business Information Entity* of **Subtotal. Tax. Amount** share strong  
855   similarities—they are the same property and share a specific *Data Type*, but are  
856   applied to different *Object Classes*—the user would identify this similarity, and use it  
857   to take the appropriate action in the discovery process.

## 858   **5.3 Preparation for Submission**

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

- 861           • Preparation of submissions will be carried out by the business domain or  
862           project group making the discovery.

- 863           • Harmonization and approval will be conducted by appropriate Assessment,  
864 Harmonization and Approval teams to be set up as part of the  
865 UN/CEFACT electronic business standards forum.

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

867 The following submissions are simple documented requests, following procedures to  
868 be established by the Assessment, Harmonization and Approval teams.

- 869           • To request registration a Re-use of an existing *Aggregate Business*  
870 *Information Entity*
- 871           • To make a Change Request for an existing *Aggregate Business*  
872 *Information Entity*
- 873           • To make a Change Request for an existing *Aggregate Core Component*

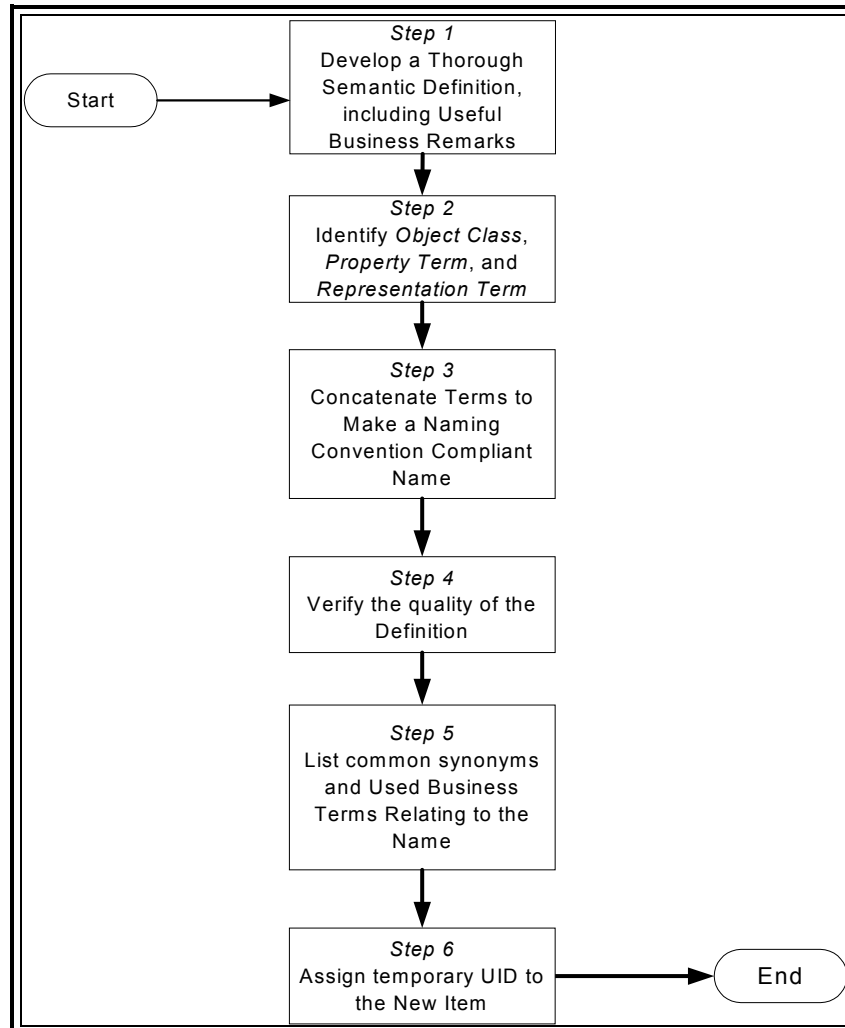
874 The following submissions require more significant preparation, as part of the *Core*  
875 *Component* working methodology, to be carried out by the business domain or project  
876 group conducting the discovery and analysis.

- 877           • Preparation for Requesting a new *Basic Core Component*
- 878           • Preparation for Requesting a new *Association Core Component*
- 879           • Preparation for Requesting a new *Aggregate Core Component*
- 880           • Preparation for Requesting a new *Basic Business Information Entity*
- 881           • Preparation for Requesting a new *Association Business Information Entity*
- 882           • Preparation for Requesting a new *Aggregate Business Information Entity*

883 Each of these needs to initially follow the same steps in applying the *Naming*  
884 *Convention* (Section 6.1.4) to arrive at the name of the new item.

### 885 **5.3.1 Applying the Naming Convention to a New Item**

886 For all new items, the *Naming Convention* and associated rules defined in Section  
887 6.1.4 must be applied. Figure 5-4 shows the steps that must be taken, each of which is  
888 described in the accompanying text.

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

890

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

- 893
- 894 • use words different to those being defined *provided* that no ambiguity is  
thereby introduced,
  - 895 • be globally applicable,
  - 896 • be generic (i.e. able to cover the same business concept for different  
897 products/services),
  - 898 • be applicable across multiple industries or domains, and
  - 899 • be simple and clear to enable unambiguous translation to other languages

900 Step 2. Follow the *Naming Convention for Core Components or Business*  
901 *Information Entities* (Section 6.1.4) to identify as appropriate:

- 902
- *Object Class Term*



- 903           • *Property Term*
- 904           • *Representation Term*
- 905           • *Qualifier Term(s)*

906           [Note]

907           When creating names for *Business Information Entities* that have properties identical  
908           to those of other, existing *Business Information Entities*, the name of the *Property*  
909           should be used to validate the correct naming of the new *Business Information Entity*.  
910           Consistent naming of similar *Business Information Entities* and *Core Components*  
911           contributes to their usability.

912           Step 3.   Concatenate the terms to create a *Naming Convention* compliant *Dictionary*  
913           *Entry Name*.

914           [Note]

915           The resultant name may seem artificial in that it might not be the same as any of the  
916           business terms used for that concept. However, rigor of the *Naming Convention*  
917           enables future translation of the name into other languages.

918           Step 4.   Verify the quality of the definition by adding the words “[*Dictionary Entry*  
919           *Name*] is” to the front of the definition, where [*Dictionary Entry Name*] is  
920           the agreed name.

921           Step 5.   List common synonyms or *Business Term(s)* that are used within the domain  
922           to identify the piece of business information (e.g. *Account Number*, *Account*  
923           *Identifier*).

924           [Note]

925           Some *Business Terms* are used for several different pieces of business information. It  
926           is perfectly acceptable to have the same *Business Term* listed as a synonym for two or  
927           more pieces of business information. For example, as shown in Figure 5-5, *Account*  
928           *Number* is a synonym for *Financial Account Identifier* and for *Sales Account*  
929           *Identifier*.

930           Step 6.   Assign a Temporary Identifier to the new item in the form of a 6 digit  
931           alphanumeric string, chosen at the discretion of the user.

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

Temp Identifier	Definition	Remarks	Business Terms	CCT	Dictionary Entry			
					Name	Object Class	Property Term	RepresentationTerm
C00010	A Financial is a service through bank or other organisation through which funds are on behalf of a or goods or are supplied on	Not a general ledger.	Account	n/a	Financial Account. Details	Financial Account	Details	
F00012	A Sales Account is relationship a vendor and a customer.	Usually includes a contract specifying the terms of	Account	n/a	Sales Account. Details	Sales Account	Details	

Same Business Term

933  
934935 **5.3.2 Preparation for Submitting New Items**

936 This section contains illustrative procedures for submitting new items. The following  
 937 subsections address submitting new *Aggregate Core Components*, new *Basic Core*  
 938 *Components*, and new *Aggregate Business Information Entities* that re-use an existing  
 939 *Aggregate Core Component*. Similar submission procedures will need to be used for  
 940 submitting *Association Core Components*, *Basic Business Information Entities*, and  
 941 *Association Business Information Entities*.

942 **5.3.2.1 New Aggregate Core Components**

943 The development of a new *Aggregate Core Component* requires adherence to the  
 944 *Naming Convention* rules for naming and definition. Once named, the new  
 945 aggregate's constituent parts need to be individually examined. The following  
 946 diagram and text describes the procedure that is to be followed.

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

949 Step 2. Identify all of the *Properties* within the new *Aggregate Core Component*.

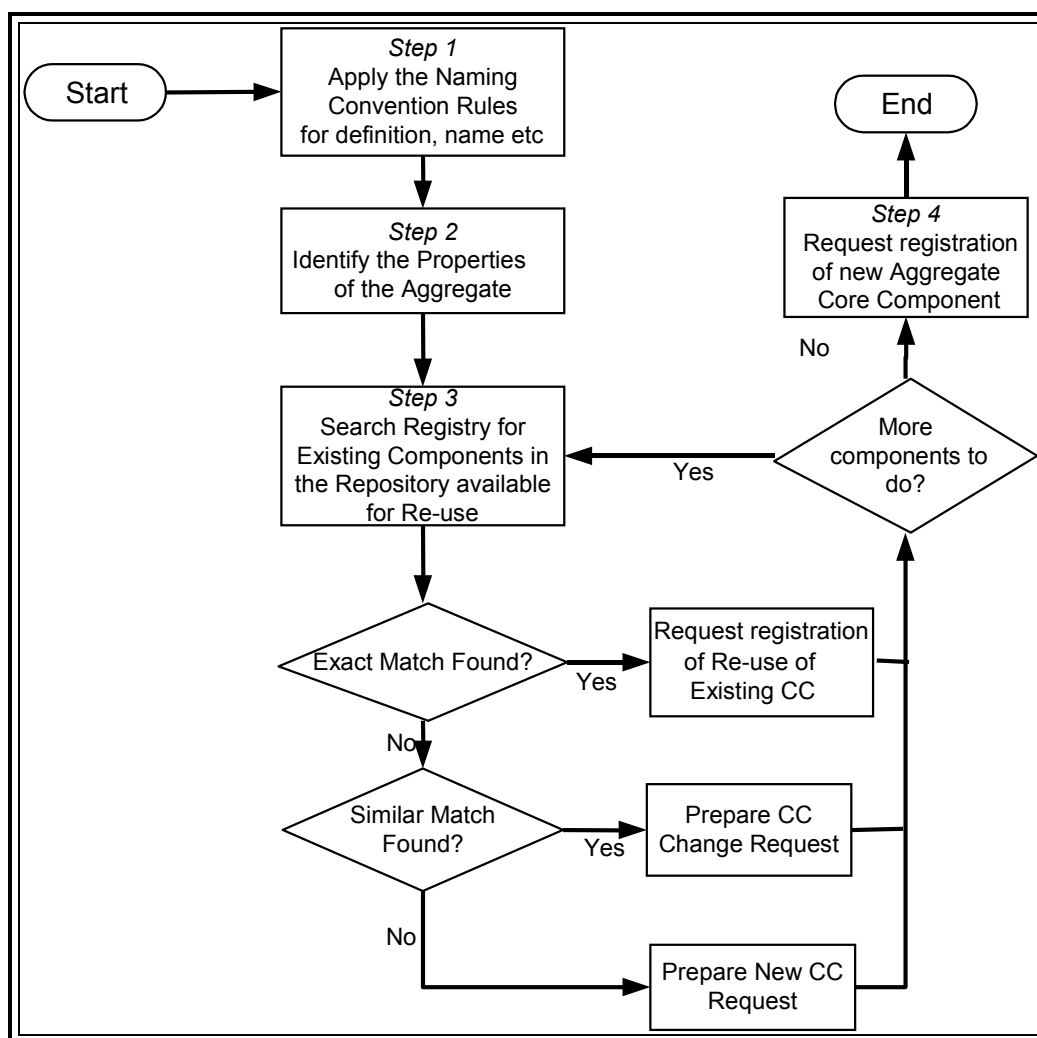
950 Repeat the following step for each constituent property identified in Step 2:

951 Step 3. Search the Registry for an existing *Core Component* or *Data Type* that has  
 952 the appropriate generic definition and structure.

- 953
- If there is an existing *Core Component* or *Data Type* with a definition and  
 954 structure that meets the requirement, request registration of this re-use of

955 the *Core Component* or *Data Type* including the *Context* in which it is  
 956 used.

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



958

- 959 • If there is an existing *Core Component* or *Data Type* with a definition and  
 960 structure that potentially could be modified to meet the requirement,  
 961 prepare an appropriate change request for submission to the harmonization  
 962 and approval process, including the re-use of the *Core Component* or *Data*  
 963 *Type* and the *Context* in which it is used.
- 964 • If there is not an existing *Core Component* or *Data Type* with a suitable  
 965 definition and structure, prepare an appropriate new item request for  
 966 submission to the harmonization and approval process, including the re-  
 967 use of the *Core Component* or *Data Type* and identification of the *Context*.

968 When all the constituent properties identified in Step 2 have been checked as  
 969 described in Step 3, then:

970 Step 4. Request registration of the new *Aggregate Core Component*.

971 Prepare the new *Aggregate Core Component* request and submit to the harmonization  
972 and approval process.

### 973 5.3.2.2 New Basic Core Components

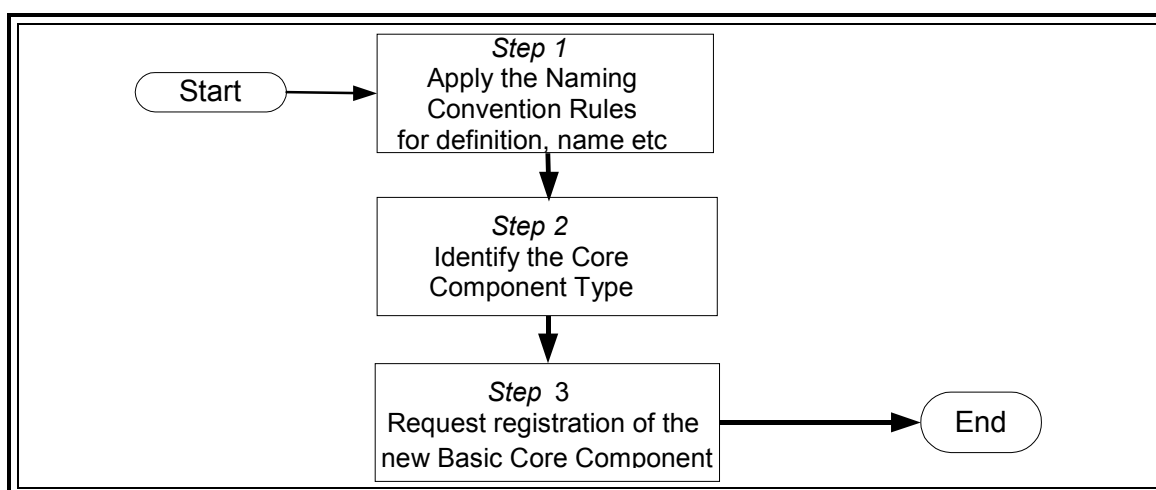
974 As shown in Figure 5-7, there are three steps necessary to prepare for requesting a  
975 new *Basic Core Component*. These three steps are:

976 Step 1. Apply the *Naming Convention Rules* to arrive at the name of the new *Basic*  
977 *Core Component*

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

980 Step 3. Request registration the new *Basic Core Component*

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



### 983 5.3.2.3 New Aggregate Business Information Entities which re-use Existing 984 Aggregate Core Components

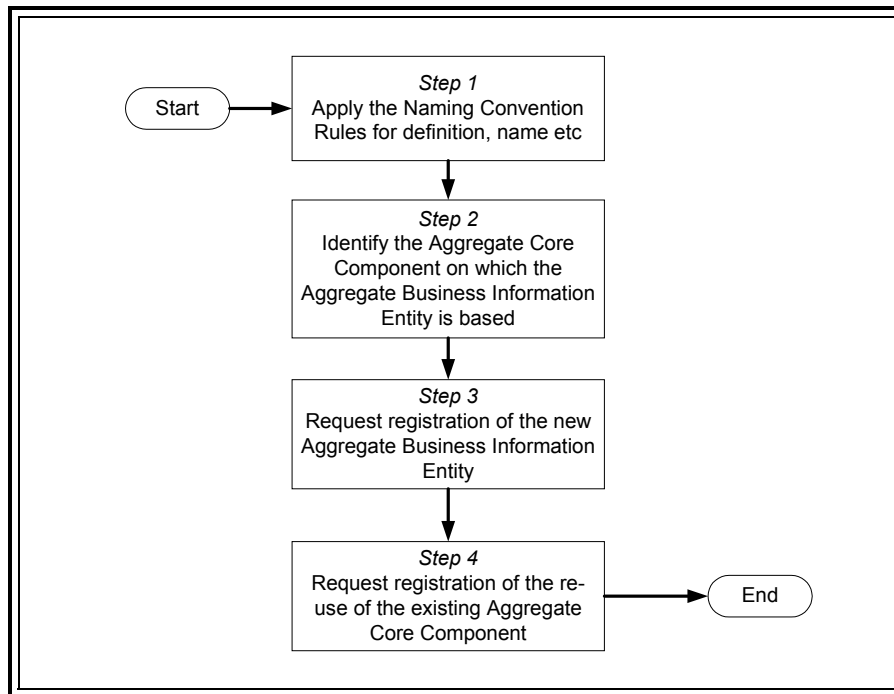
985 As shown in Figure 5-8, there are four steps necessary to prepare for requesting a new  
986 *Aggregate Business Information Entity* that re-uses an existing *Aggregate Core*  
987 *Component*. These four steps are:

988 Step 1. Apply the *Naming Convention Rules* to arrive at the name of the new  
989 *Aggregate Business Information Entity*.

990 Step 2. Identify the *Aggregate Core Component* on which the new *Aggregate*  
991 *Business Information Entity* is based

992 Step 3. Request registration of the new *Aggregate Business Information Entity*

993 Step 4. Request registration the re-use of the existing *Aggregate Core Component*  
994 by this new *Aggregate Business Information Entity*.

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

996

997 **5.4 Harmonization**

998 The purpose of harmonization is to take the candidate *Core Components* or *Business*  
 999 *Information Entities* submitted by different domains, identify differences and  
 1000 similarities between the submissions and existing library entries, and produce a single,  
 1001 complete cross-domain set, i.e. the *Core Component Library*. Harmonization is a  
 1002 critical process in the overall *Core Component* procedures. The following describes  
 1003 the recommended areas that harmonization procedures should cover.

- 1004
- 1005 • Evaluate each submitted *Core Component* for consistent application of the  
 1006 discovery methodology. Resolve any questions or issues by discussion  
 with the submitting groups.
  
  - 1007 • Compare the definition and structure of each submitted *Core Component*  
 1008 with what already exists in the *Core Component Library*.
  
  - 1009 - If the submitted *Core Component* is the same or similar, compare the  
 1010 properties of each to identify any differences. If the submitted *Core*  
 1011 *Component* has properties missing in the existing one, enforce a  
 1012 harmonized form that contains the properties of each. If the submitted  
 1013 *Core Component* is a subset of the existing *Core Component*  
 1014 definition, then recommend the use of the existing one. Similarities  
 1015 between *Core Components* should be judged on whether or not the  
 1016 *Property* of each shares a *Data Type*. A *Data Type* should be reused as  
 1017 much as possible across *Properties* of *Core Components*.

- 1018 - If the definition of the *Core Component* does not match any existing  
1019 ones, then proceed.
- 1020 • Publish the results of harmonization to the submitting groups for review  
1021 and finalisation.

1022 [Note]

1023 In order to ensure that each submission is evaluated on its own merits, and that no  
1024 submission is given precedence over others, all submissions should be processed  
1025 separately and serially against the full cross-domain library.

1026 Once the submitted material has passed the harmonization procedure, it may now be  
1027 submitted for technical assessment and approval.

## 1028 **5.5 Technical Assessment and Approval**

1029 Technical assessment must be done in close coordination with the discovery teams  
1030 and the harmonization process in order to minimise domain re-working after technical  
1031 assessment and harmonization review. This section defines a recommended process  
1032 for conducting technical assessment and approval of all newly submitted and changed  
1033 *Core Components*. A technical assessment and approval process for *Business*  
1034 *Information Entities* should also be developed and applied.

1035 Technical assessment procedures define the processing that shall be followed by the  
1036 joint development groups, the harmonization group, submission entry points, the  
1037 technical assessment group, and the secretariat as related to the review of *Core*  
1038 *Components*. The result of this process is the final publication of approved *Core*  
1039 *Components*.

1040 These procedures are needed in order to facilitate the process of reviewing and  
1041 approving submissions to the *Core Component Library*. In order to minimise the  
1042 requirements for technical assessment and harmonization, and to expedite the review  
1043 and approval process, *Core Component* development groups should work with the  
1044 technical assessment group, and the harmonization group during the early  
1045 development stages of component discovery.

1046 In outline, these procedures should cover:

- 1047 • Submission of *Core Component* work that is ready to be reviewed to a  
1048 designated secretariat.
- 1049 • Recording of all *Core Component* submissions and distribution to the  
1050 harmonization group members.
- 1051 • Review procedures and criteria followed by the harmonization group.

- 1052           • Return of harmonized *Core Component* submissions for technical  
1053           assessment.
- 1054           • Review procedures and criteria followed by the technical assessment  
1055           group.
- 1056           • Registration of the approved *Core Component(s)* in the appropriate *Core*  
1057           *Component* registry.

## 1058   **5.6   Context in the Discovery Process**

1059   Information that is needed by a *Business Process* is used in a *Context* that is defined  
1060   by how and where the *Business Process* can take place. The initial analysis will be  
1061   performed on a set of *Business Information Entities*, i.e. *Basic*, *Association*, and  
1062   *Aggregate Business Information Entities*, and not on a set of *Core Components* (See  
1063   Figure 5-1). The analysis that produces *Core Components* is, among other things, a  
1064   process of identifying the various *Context Categories* and values, to determine the  
1065   underlying context-independent *Properties*.

1066   The guidelines presented here facilitate the analysis of *Business Information Entities*  
1067   to determine core business semantics, or provide a mechanism to describe *Business*  
1068   *Information Entities* when they are entered into a registry and published in a  
1069   repository.

1070   If there are any instances of the *Business Information Entity* in which a *Property* is not  
1071   present, then this raises the issue of identity. Specifically – is the *Business*  
1072   *Information Entity* which lacks that property really the same *Business Information*  
1073   *Entity*, just used in a different *Context*?

1074   If the answer to this question is *yes*, then that property is part of the *Core Component*.  
1075   If the answer is *no*, then it is possible that a second, different *Core Component* has  
1076   been discovered.

### 1077   **5.6.1   Context Categories**

1078   *Context Categories* are introduced here and are followed by a brief description. After  
1079   which the various guidelines used to determine *Context* are introduced:

- 1080           • *Business Process Context* – This is the classification of the *Business*  
1081           *Process*, business collaboration, or business transaction as described in the  
1082           *UN/CEFACT Catalogue of Common Business Processes*. It is the primary  
1083           *Context Category*, and provides many useful distinctions in the analysis of  
1084           *Core Components*.
- 1085           • *Product Classification Context* – There are many types of information that  
1086           are specific to products or services being traded or referred to in a *Business*  
1087           *Process*.

- 1088 • *Industry Classification Context* – Traditionally, business vocabularies are  
1089 divided up into industry verticals. This *Context Category* specifies a  
1090 particular industry vertical.
- 1091 • *Geopolitical Context* – Specifies the semantic and structural variation. This  
1092 is often the result of regional or cultural factors.
- 1093 • *Official Constraints Context* – Specifies the legal or contractual influences  
1094 upon business semantics.
- 1095 • *Business Process Role Context* – Every partner in a *Business Process* data  
1096 exchange has a particular role – buyer, seller, etc. These roles are  
1097 described in the *UN/CEFACT Catalogue of Common Business Processes*  
1098 and in other *Business Libraries* (libraries of *Business Process* models).  
1099 Depending on the *Business Process*, the nature of these roles may require  
1100 that certain semantics and data be employed in the messages exchanged. In  
1101 any *Business Process Role Context*, one must either be a sender or receiver  
1102 of data in that particular exchange – otherwise, role is described by the  
1103 *Supporting Role Context*.
- 1104 • *Supporting Role Context* – Parties in a *Business Process* who are neither  
1105 senders nor receivers of data in a particular exchange, may place  
1106 requirements on the data exchanged by partners who are sending or  
1107 receiving of data in that exchange. These non-sending, non-receiving  
1108 parties in this exchange play a supporting role, and are described by the  
1109 *Supporting Role Context*.
- 1110 • *System Capabilities Context* – When a particular semantic or structure is  
1111 primarily the result of system constraints, or compliance with a standard,  
1112 then it is attributable to the *System Capabilities Context*.

## 1113 **5.6.2 Guidelines for Analysing Business Information Entities in Context**

1114 Using the criteria given in section 5.6.1 for determining that a particular property of a  
1115 *Business Information Entity* is in fact the product of its use in *Context*, the analyst  
1116 must ascertain and document the applicable *Context Categories*. To accomplish this,  
1117 the analyst should list all the *Context Categories*, and assign a value or values to each  
1118 category for that component. If a *Context* category has no particular value or values,  
1119 then the analyst should assign a value of *In All Contexts* (for all *Contexts* except  
1120 *Official Constraints*) or *None* (for *Official Constraints*). As this analysis is conducted,  
1121 different *Context Categories* might appear to be in competition for application. The  
1122 analyst must ascertain which *Context Category* is responsible. This section provides  
1123 some guidelines for answering this question in a systematic and consistent fashion, by  
1124 examining the typical ambiguities that arise.

1125 It is possible that a particular *Property* of a *Business Information Entity* may be the  
1126 result of several *Context* factors. These *Context* factors are identified by analysis of  
1127 differences and similarities across particular *Contexts*. For example, comparing the  
1128 same *Business Information Entity* as used in different regions of the world, variation



1129 will probably be the result of a *Geopolitical Context* or *Official Constraints Context*  
1130 (see below). If a single *Business Information Entity* differs between *Business*  
1131 *Processes*, then the *Business Process Context* is probably the cause.

1132 The following guidelines apply:

1133 1) *Geopolitical Context* versus *Official Constraints Context*

1134 If a property can be traced to a specific body of law or international treaty then it is  
1135 the result of an *Official Constraint*. For example, if a warning about hazardous  
1136 goods is required as part of a goods description, and it is required on all uses of that  
1137 goods description within the United States, then both *Geopolitical* and *Official*  
1138 *Constraints* are involved. The value of an *Official Constraint Context* should  
1139 always be the body of law or treaty that is being cited. The value of a *Geopolitical*  
1140 *Context* always expresses the region or regions that are relevant.

1141 2) *Product Classification Context* versus *Industry Classification Context*

1142 When a particular variation on a given product or service is specific to a particular  
1143 industry, then the *Industry Classification Context* is adequate to specify the  
1144 *Context*. If all examples of the particular product or service are described by the  
1145 same unique set of *Properties* across industries, then only a *Product Classification*  
1146 *Context* is required. In other cases, a value or values should be supplied for both  
1147 *Context Categories*.

1148 3) *Business Process Context* versus *Business Role Context*

1149 *Business Role Context* is employed when one actor in the *Business Process* has an  
1150 information requirement and the other does not. If both actors have the same  
1151 information requirement, then it is a *Business Process Context*.

1152 4) *System Capability Context Categories*

1153 This *Context* is the result of system or classes of systems that *primarily* influence  
1154 data variation. For example, if a specific Enterprise Resource Planning (ERP)  
1155 provider's proprietary data formats use a particular field, and no other applications  
1156 use that field, then the presence of the data can be attributed to the processing  
1157 capabilities of that specific system.

1158 The following detailed example illustrates the process of assigning values for all  
1159 *Context Categories* as part of the *Business Information Entity* analysis process:

1160 [Example]

1161 Case: A buyer address *Business Information Entity* is taken from a standard that is  
1162 used across all industry boundaries and in all processes within the United States. The  
1163 *Business Information Entity* also contains a *Property* that holds the *State* information.  
1164 The following set of values could be ascribed to this *Property* for this *Business*  
1165 *Information Entity*:

1166       *Business Process* = *In All Contexts*  
1167       *Product Classification* = *In All Context*  
1168       *Industry Classification* = *In All Contexts*  
1169       *Geopolitical* = *United States*  
1170       *Official Constraint* = *None*  
1171       *Business Process Role* = *In All Contexts*  
1172       *Supporting Role* = *In All Contexts*  
1173       *System Capabilities* = *In All Contexts*

1174 These values were selected based on the following analysis:

1175 The *Business Information Entity* construct is the same in every *Business Process*  
1176 covered by the standard in question – the address always contains a *State* field.  
1177 Therefore, for the range of *Business Processes* covered by the *Business Information*  
1178 *Entity* being analysed, – the *Business Process Context* category is marked *In All*  
1179 *Contexts*.

1180 The products that might be described in the same business message do not affect the  
1181 address. Since the standard from which the *Business Information Entity* has been  
1182 extracted is horizontal across industry boundaries, it is equally valid in all *Industry*  
1183 *Classification Contexts*.

1184 As a *Property* of *Buyer Address*, it is clear that the *State Property* is intended to hold a  
1185 value specific to United States geopolitical demarcations. Therefore the *Geopolitical*  
1186 *Context Category* is properly assigned the value *United States*.

1187 No specific law can be cited that requires the presence of the *State Property* in the  
1188 address. Therefore, a value of *None* is given to the *Official Constraint Context*  
1189 *Category*.

1190 On inspection of *Business Process Role*, it appears that all addresses in the standard in  
1191 question are required to provide the *State* information, regardless of what role they  
1192 play in the transaction. The fact that a *Buyer Role* is being analysed has no effect on  
1193 this *Property*: all types of addresses have the same semantics. Therefore, all roles  
1194 provide the data equally when giving an address. A value of *In All Contexts* is  
1195 applicable here. The same reasoning holds for the *Supporting Role Context*.

1196 Finally, considering the *System Capabilities Context*, there are no specific systems  
1197 that act as the primary reason for the presence or absence of the semantic. Instead, the  
1198 primary existence of the *Property* can be ascribed to the fact that in common usage,  
1199 US addresses include the *State Property*. Therefore, we can provide the value *In All*  
1200 *Contexts* here. Note that as wide of a range of values as possible should be provided to  
1201 ensure completeness.

1202

1203 If, in the above example, the address was taken from a French standard, it might be  
1204 that some *Properties* are common across a number of countries in the same region,  
1205 and perhaps even in multiple regions. Providing the value *France* as a *Geopolitical*  
1206 *Context* here would be incomplete – every known valid value should be given.

## 1207 **6 Technical Details**

1208 This section provides a detailed technical explanation of the *Core Component*, *Business*  
1209 *Process* integration, storage and metamodel elements of the UN/CEFACT *Core*  
1210 *Components* concept.

1211 The *Core Component* framework prescribes the mechanism for discovery, normalisation,  
1212 *Context* specialisation, and structure of *UMM InformationEntities*. The *Aggregate*  
1213 *Business Information Entity-Basic Business Information Entity* framework provides the  
1214 structure for components of the body of the business document. The *Core Component-*  
1215 *Business Information Entity-Context* mapping framework provides the basis for mapping  
1216 information entity realizations to business entities. The *Business Information Entity* to  
1217 *Core Component* relationship provides the dictionary reference as specified in the  
1218 information model abstract syntax. The *Core Component Library* is an implementation of  
1219 the UMM dictionary concept. The *Basic Core Component* is the realization of a *UMM*  
1220 *InformationEntity* and provides the mapping to *Data Types*.

### 1221 **6.1 Core Components, Data Types and Business Information** 1222 **Entities**

1223 This section defines the following:

- 1224 • *Core Component* rules,
- 1225 • *Data Type* rules,
- 1226 • *Business Information Entity* rules,
- 1227 • *Naming Conventions*,
- 1228 • *Core Component Types*,
- 1229 • *Content* and *Supplementary Components*, and
- 1230 • *Representation Terms*.

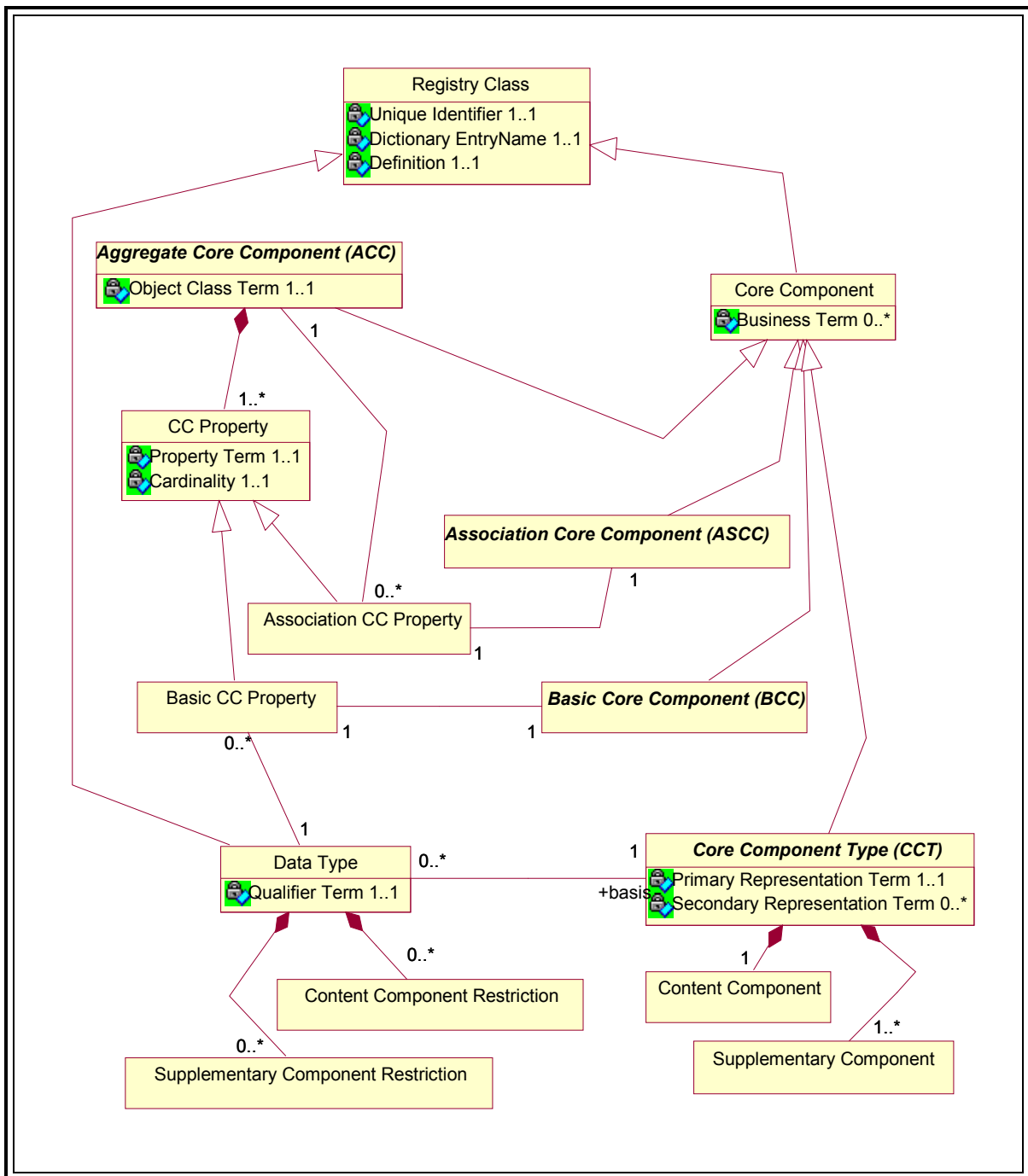
1231 This section also specifies relationships between *Core Components*, *Data Types* and  
1232 *Business Information Entities* and includes details required for constructing the *Core*  
1233 *Components Catalogue* and a larger *Core Component Library*.

#### 1234 **6.1.1 Core Components**

1235 A *Core Component* is a building block for the development of a semantically correct and  
1236 meaningful business information exchange ‘parcel’ containing the information pieces  
1237 needed to describe a specific concept. There are four categories of *Core Components*:  
1238 *Basic Core Component*, *Association Core Component*, *Core Component Type* and  
1239 *Aggregate Core Component*. Figure 6-1 illustrates these four categories and their

1240 relationships. The complete *Core Component* Metamodel is illustrated in Figure 7-1.  
 1241 Models are normative to the level of detail at which they exist.

1242 **Figure 6-1. Core Components and Data Types Metamodel**



1243  
 1244

1245 The following general rules must be followed in discovering and documenting the four  
 1246 types of *Core Components*:

- 1247 [C1] Each *Core Component Type*, *Basic Core Component*, *Association Core*
- 1248 *Component* or *Aggregate Core Component* must have its own unique semantic

1249 definition within the library of which it is a part. The definition shall be developed  
1250 first and the *Dictionary Entry Name* shall be extracted from it. Comments can be  
1251 used to further clarify the definition, to provide examples and/or to reference a  
1252 recognized standard.

1253 [Note]

1254 The *Core Components Dictionary* is one of several ways that *Core Components* are to be  
1255 made available. The purpose of the *Core Components Dictionary* is to provide a ready  
1256 reference of the *Core Component* through its *Dictionary Entry Name*, component parts,  
1257 and definition. The *Core Components Dictionary* will be considered a supplement to the  
1258 *Catalogue of Core Components* which in turn is a documented listing of the contents of  
1259 the *Core Components Registry/Repository*.

1260 [C2] Within an *Aggregate Core Component*, all embedded *Core Component Properties*  
1261 shall be related to the concept of the aggregate.

1262 [C3] There shall be no semantic overlap between the *Core Component Properties*  
1263 embedded within the same *Aggregate Core Component*.

1264 [C4] The representation of the information in a *Core Component* whose *Core*  
1265 *Component Type* is *Code*. *Type* should use a standard issued by a recognized  
1266 standards body, whenever a standard exists. If international standards are not used  
1267 a business driven justification shall be provided.

1268 [C5] An *Aggregate Core Component* shall contain at least one *Core Component*  
1269 *Property*. A *Core Component Property* shall be either a *Basic Core Component*  
1270 *Property* or an *Association Core Component Property*.

1271 [Note]

1272 At the deepest level of nesting an *Aggregate Core Component* shall only contain *Basic*  
1273 *Core Component Properties*.

1274

1275 [Note]

1276 For the purpose of exchanging information a practical compromise on the level of detail  
1277 of a *Basic Core Component* is required. This compromise shall be based on the business  
1278 need. It is not necessary to have absolute detail, which decomposes a piece of information  
1279 down to its lowest level.

1280 [C6] An *Aggregate Core Component* shall never contain - indirectly or at any nested  
1281 level - a mandatory *Association Core Component Property* that references itself.

1282 [Note]

1283 The objective of the above rule is to avoid endless loops in the definition of an *Aggregate*  
1284 *Core Component*. The rule allows an *Aggregate Core Component* to contain an  
1285 *Association Core Component Property* that references itself. The fact that the *Association*  
1286 *Core Component Property* is not mandatory makes it possible to stop the loop after a  
1287 finite number of iterations.

1288 [C7] The *Core Component Type* shall be one of the approved *Core Component Types*  
1289 Table 8-1 provides a complete list of the approved *Core Component Types* as of the date  
1290 of this specification.

1291 [Note]

1292 Table 8-1 may subsequently be published separately to facilitate maintenance outside the  
1293 body of this specification.

1294 Table 8-2 provides a complete list of the approved *Content Components* and  
1295 *Supplementary Components* as of the date of this specification.

1296 [C8] The *Content Component* shall be the approved *Content Component* for the related  
1297 *Core Component Type*

1298 [C9] The *Supplementary Component* shall be one of the approved *Supplementary*  
1299 *Components* for the related *Core Component Type*

1300 [Note]

1301 Table 8-2 may subsequently be published separately to facilitate maintenance outside the  
1302 body of this specification.

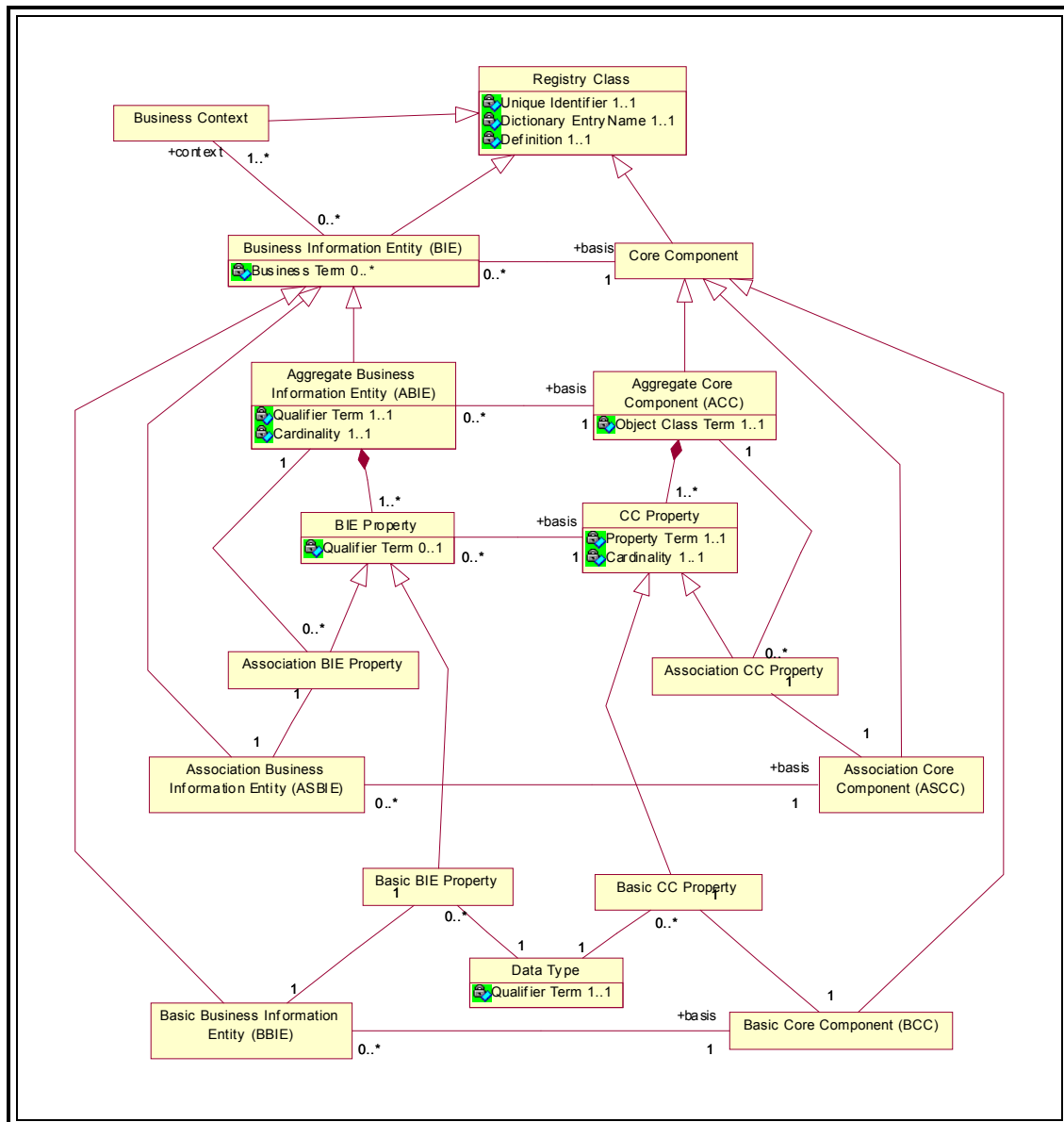
1303

### 1304 **6.1.2 Data Types**

1305 A *Data Type* defines the set of valid values that can be used for a particular *Basic Core*  
1306 *Component Property* or *Basic Business Information Entity Property*. It is defined by  
1307 specifying restrictions on the *Core Component Type* from which the *Data Type* is  
1308 derived. Figure 6-1 describes the *Data Type* and shows relationships to the *Core*  
1309 *Component Type*.

1310 [D1] A *Data Type* shall be based on one of the approved *Core Component Types*.

1311 [D2] Where necessary, a *Data Type* shall restrict the set of valid values allowed by the  
1312 *Core Component Type* on which it is based, by imposing restrictions on the  
1313 *Content Component* and/or the *Supplementary Component*.

1314 **Figure 6-2. Business Information Entities Basic Definition Model**

1315

1316 **6.1.3 Business Information Entities**

1317 A *Business Information Entity* is a piece of business data or a group of pieces of business  
 1318 data with a unique business semantic definition in a specific *Business Context*. A  
 1319 *Business Information Entity* can be a *Basic Business Information Entity* (BBIE), an  
 1320 *Association Business Information Entity* (ASBIE) or an *Aggregate Business Information*  
 1321 *Entity* (ABIE).

1322 • A *Basic Business Information Entity* is based on a *Basic Core Component* (BCC).

1323 • An *Association Business Information Entity* is based on an *Association Core*  
 1324 *Component* (ASCC).



- 1325 • An *Aggregate Business Information Entity* is a re-use of an *Aggregate Core*  
1326 *Component (ACC)* in a specified *Business Context*.
- 1327 Figure 6-2 describes the *Business Information Entity* types and shows relationships to the  
1328 *Core Component* counterparts.
- 1329 [B1] A *Business Information Entity* shall be a *Basic Business Information Entity*, an  
1330 *Association Business Information Entity* or an *Aggregate Business Information*  
1331 *Entity*
- 1332 [B2] A *Business Information Entity* shall be defined by one or more *Business Contexts*
- 1333 [B3] A *Basic Business Information Entity* shall be based on a *Basic Core Component*
- 1334 [B4] An *Association Business Information Entity* shall be based on an *Association*  
1335 *Core Component*
- 1336 [B5] An *Aggregate Business Information Entity* shall be based on an *Aggregate Core*  
1337 *Component*
- 1338 [B6] An *Aggregate Business Information Entity* shall contain at least one *Business*  
1339 *Information Entity Property*. A *Business Information Entity Property* shall either  
1340 be a *Basic Business Information Entity Property* or an *Association Business*  
1341 *Information Entity Property*.
- 1342 [Note]
- 1343 At the deepest nesting level an *Aggregate Business Information Entity* will only contain  
1344 *Basic Business Information Entity Properties*.
- 1345 [B7] A *Business Information Entity Property* of an *Aggregate Business Information*  
1346 *Entity* shall be based on a *Core Component Property* of the corresponding  
1347 *Aggregate Core Component*.
- 1348 [B8] The *Data Type*, on which a *Basic Business Information Entity Property* is based,  
1349 shall itself be similar to the *Data Type* on which the corresponding *Basic Core*  
1350 *Component Property* is based (i.e. it shall either be the same *Data Type* or a more  
1351 restricted one).
- 1352 [B9] The *Aggregate Business Information Entity*, on which an *Association Business*  
1353 *Information Entity Property* is based, shall itself be based on the *Aggregate Core*  
1354 *Component* on which the corresponding *Association Core Component Property* is  
1355 based.

1356 [B10] An *Aggregate Business Information Entity* shall never contain – directly or at any  
1357 nested level – a mandatory *Association Business Information Entity Property* that  
1358 references itself.

1359 [Note]

1360 The objective of the above rule is to avoid endless loops in the definition of an *Aggregate*  
1361 *Business Information Entity*. The rule allows an *Aggregate Business Information Entity* to  
1362 contain an *Association Business Information Entity Property* that references itself. The  
1363 fact that the *Association Business Information Entity Property* is not mandatory makes it  
1364 possible to stop the loop after a finite number of iterations.

#### 1365 **6.1.4 Naming Convention**

1366 A *Naming Convention* is necessary to gain consistency in the naming and defining of all  
1367 *Core Components, Data Types* and *Business Information Entities*. The resulting  
1368 consistency facilitates comparison during the discovery and analysis process, and  
1369 precludes ambiguity, such as the development of multiple *Core Components* with  
1370 different names that have the same semantic meaning.

1371 The *Naming Convention* is derived from the guidelines and principles described in  
1372 document ISO 11179 Part 5 -- *Naming and Identification Principles For Data Elements*.  
1373 In certain instances, these guidelines have been adapted to the *Core Component*  
1374 environment. In particular, the guidelines have been extended to cover the naming and  
1375 defining of *Core Component Types, Data Types* and *Business Information Entities*.

1376 In order to ensure absolute clarity and understanding of the names and definitions it is  
1377 essential to use words from the *Oxford English Dictionary*. A supplementary *Controlled*  
1378 *Vocabulary* will be developed to identify the definition to be used for any words that are  
1379 potentially ambiguous. This *Controlled Vocabulary* shall also be used to identify the  
1380 preferred word in cases where more than one word might be used to cover the same  
1381 definition. The *Controlled Vocabulary* will also contain terms not found in the *Oxford*  
1382 *English Dictionary*. This will ensure that each word within any of the names and  
1383 definitions is used in a consistent and unambiguous way. The resultant semantic integrity  
1384 will also mean that translation into other languages retains the precise original meaning.

##### 1385 **6.1.4.1 Core Component Naming Rules**

1386 The following subsections contain all *Core Component* naming rules.

###### 1387 **6.1.4.1.1 Core Component Dictionary Information**

1388 Each *Core Component* contains the following dictionary information that is impacted by  
1389 the naming rules in subsequent sub-sections:

- 1390 • **Dictionary Entry Name** (Mandatory). This is the unique official name of the  
1391 *Core Component* in the dictionary.

- 1392           • **Definition** (Mandatory). This is the unique semantic business meaning of that  
1393           *Core Component*.
- 1394           • **Business Term** (Optional). This is a synonym term under which the *Core*  
1395           *Component* is commonly known and used in the business. A *Core Component*  
1396           may have several business terms or synonyms.

1397           [Example]

1398           *Dictionary Entry Name* – **Person. Tax. Identifier**

1399           Definition – The registered national tax identification of a person

1400           *Business Term* – Income tax number, national register number, personal tax register  
1401           number, social security number, national insurance number

1402           The naming rules are also based on the following concepts as defined in ISO 11179:

- 1403           • **Object Class**. This represents the logical data grouping or aggregation (in a  
1404           logical data model) to which a *Property* belongs. The *Object Class* is  
1405           expressed by an *Object Class Term*. The *Object Class* thus is the part of a  
1406           *Core Component's Dictionary Entry Name* that represents an activity or object  
1407           in a specific *Context*. *Object Classes* have explicit boundaries and meaning  
1408           and their *Properties* and behaviour follow the same rules.
- 1409           • **Property Term**. This represents the distinguishing characteristic or *Property*  
1410           of the *Object Class* and shall occur naturally in the definition.
- 1411           • **Representation Term**. An element of the *Core Component* name which  
1412           describes the form in which the *Core Component* is represented.

#### 1413   6.1.4.1.2 Core Component General Rules

1414   [C10]           The dictionary content shall be in *English Language* following the primary  
1415           *Oxford English Dictionary* English spellings to assure unambiguous  
1416           spelling.

1417           [Note]

1418           There may be restrictions in specific languages, which need to be applied when  
1419           transforming the *Core Component Dictionary* into other languages. These restrictions  
1420           shall be formulated as additional rules and added as separate language specific annexes to  
1421           this document.

1422

## 1423 6.1.4.1.3 Core Component Rules for Definitions

1424 [C11] The definition shall be consistent with the requirements of ISO 11179-4 Section  
1425 4.4 and will provide an understandable meaning, which should also be  
1426 translatable to other languages.

1427 [C12] The definition shall take into account the fact that the users of the *Core*  
1428 *Component Dictionary* are not necessarily native English speakers. It shall  
1429 therefore contain short sentences, using normal words. Wherever synonym terms  
1430 are possible, the definition shall use the preferred term as identified in the  
1431 *Controlled Vocabulary*.

1432 [C13] The definition of a *Basic Core Component* shall use a structure that is based on  
1433 the existence of the *Object Class Term*, the *Property Term*, and the *Data Type* of  
1434 the corresponding *Basic Core Component Property*.

1435 [C14] The definition of an *Association Core Component* shall use a structure that is  
1436 based on the existence of the *Object Class Term*, the *Property Term* and the  
1437 *Object Class Term* of the *Aggregate Core Component* on which the corresponding  
1438 *Association Core Component Property* is based.

1439 [C15] Whenever both the definite (i.e. *the*) and indefinite article (i.e. *a*) are possible in a  
1440 definition, preference shall be given to the indefinite article (e. *a*).

1441 [Note]

1442 To verify the quality of the definition, place the *Dictionary Entry Name* followed by *is*  
1443 before the definition to ensure that it is not simply a repetition of the *Dictionary Entry*  
1444 *Name*.

## 1445 6.1.4.1.4 Core Component Rules for Dictionary Entry Names

1446 [C16] The *Dictionary Entry Name* shall be unique.

1447 [C17] The *Dictionary Entry Name* shall be extracted from the *Core Component*  
1448 definition.

1449 [C18] The *Dictionary Entry Name* shall be concise and shall not contain consecutive  
1450 redundant words.

1451 [C19] The *Dictionary Entry Name* and all its components shall be in singular form  
1452 unless the concept itself is plural.

1453

[Example]

1454

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

1455

1456

[C20] The *Dictionary Entry Name* shall not use non-letter characters unless required by language rules.

1457

1458

[C21] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e. no words like *and*, *of*, *the*, etc.). This rule shall be applied to the English language, and may be applied to other languages as appropriate.

1459

1460

1461

[C22] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall be expanded or explained in the definition.

1462

1463

[C23] The *Dictionary Entry Name* of a *Basic Core Component* shall consist of the following parts in the order specified:

1464

1465

- the *Object Class Term* of the *Aggregate Core Component* owning the corresponding *Basic Core Component Property*,

1466

1467

- the *Property Term* of the corresponding *Basic Core Component Property*, and

1468

- the *Representation Term* of the *Data Type* on which the corresponding *Basic Core Component Property* is based.

1469

1470

[Example]

1471

**Tax. Description. Text**

1472

[C24] The *Dictionary Entry Name* of an *Association Core Component* shall consist of the following components in the specified order:

1473

1474

- the *Object Class Term* of the *Aggregate Core Component* owning the corresponding *Association Core Component Property*,

1475

1476

- the *Property Term* of the corresponding *Association Core Component Property*,

1477

- the *Object Class Term* of the *Aggregate Core Component* on which the corresponding *Association Core Component Property* is based.

1478

1479 [Example]

1480 **Person. Residence. Address**

1481 [C25] The components of a *Dictionary Entry Name* shall be separated by dots. The  
 1482 space character shall separate words in multi-word *Object Class Terms* and/or  
 1483 multi-word *Property Terms*. Every word shall start with a capital letter. To allow  
 1484 spell checking of the *Directory Entry Names'* words, the dots after *Object Class*  
 1485 *Terms* and *Property Terms* shall be followed by a space character.

1486 [Note]

1487 The use of CamelCase for *Dictionary Entry Names* has been considered, but has been  
 1488 rejected for following reasons:

1489 ◆ Use of CamelCase will not allow the use of spell checkers

1490 ◆ Strict use of CamelCase makes it impossible to use separators (“.”) and  
 1491 therefore doesn't allow an unambiguous identification of the composing  
 1492 parts of the *Dictionary Entry Name*.

1493 [C26] The name of an *Object Class* shall always have the same semantic meaning  
 1494 throughout the dictionary and may consist of more than one word.

1495 [C27] The name of a *Property Term* shall occur naturally in the definition and may  
 1496 consist of more than one word. A name of a *Property Term* shall be unique within  
 1497 the *Context* of an *Object Class* but may be reused across different *Object Classes*.

1498 [Example]

1499 **Car. Colour. Code** and **Shirt. Colour. Code** may both exist.

1500 [C28] For *Basic* and *Association Core Components*, if the *Property Term* uses the same  
 1501 (or equivalent) word or words as the third component of the *Dictionary Entry*  
 1502 *Name*, the redundant word(s) in the *Property Term* shall be removed from the  
 1503 *Dictionary Entry Name*.

1504 [Note]

1505 This may lead to the case where the complete *Property Term* is removed from the  
 1506 *Dictionary Entry Name*.

1507

1508

[Example]

1509

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

1510

1511

1512

1513

[C30] The *Dictionary Entry Name* of a *Core Component Type* shall consist of a *Representation Term* followed by a dot, a space character, and the term *Type*.

1514

1515

[Example]

1516

**Amount. Type; Date Time. Type**

1517

[C31] In the *Dictionary Entry Name* of a *Core Component Type*, the name of the *Representation Term* shall be one of the primary terms specified in the list of permissible *Representation Terms* as included in this specification (See section 8.3).

1518

1519

1520

1521

[C32] The *Dictionary Entry Name* of an *Aggregate Core Component* shall consist of a meaningful *Object Class Term* followed by a dot, a space character, and the term *Details*. The *Object Class Term* may consist of more than one word.

1522

1523

1524

[Example]

1525

**Postal Address. Details; Party. Details**

1526

#### 6.1.4.1.5 Rules for Core Component Business Terms

1527

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

1528

1529

1530

1531

#### 6.1.4.2 Rules for Business Information Entities

1532

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

1533

##### 6.1.4.2.1 Business Information Entity Dictionary Information

1534

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

1535

- 1536 • **Dictionary Entry Name** (Mandatory). This is the unique official name of the  
1537 *Business Information Entity* in the dictionary.
- 1538 • **Definition** (Mandatory). This is the unique semantic business meaning of that  
1539 *Business Information Entity*.
- 1540 • **Business Term** (Optional). This is a synonym term under which the *Business*  
1541 *Information Entity* is commonly known and used in the business for a specific  
1542 *Context*. A *Business Information Entity* may have several business terms or  
1543 synonyms.

1544 The *Business Information Entity* naming rules are also based on the following concepts as  
1545 defined in ISO 11179:

- 1546 • **Object Class**. This represents the logical data grouping or aggregation (in a  
1547 logical data model) to which a data element belongs. The *Object Class* is  
1548 expressed as an *Object Class Term*. The *Object Class* thus is the part of a  
1549 *Business Information Entity's Dictionary Entry Name* that represents an  
1550 activity or object in a specific *Context*. *Object Classes* have explicit  
1551 boundaries and meaning and their *Properties* and behaviour follow the same  
1552 rules.
- 1553 • **Property Term**. This represents the distinguishing characteristic or *Property*  
1554 of the *Object Class* and shall occur naturally in the definition.
- 1555 • **Representation Term**. An element of the *Business Information Entity* name  
1556 which describes the form in which the *Business Information Entity* is  
1557 represented.
- 1558 • **Qualifier Term**. A word or words which help define and differentiate a  
1559 *Business Information Entity* from its associated *Core Component* and other  
1560 *Business Information Entities*.

#### 1561 6.1.4.2.2 Business Information Entity General Rules

1562 [B11] The dictionary content shall be in English Language following the primary  
1563 *Oxford English Dictionary* English spellings to assure unambiguous spelling.

#### 1564 6.1.4.2.3 Business Information Entity Rules for Definitions

1565 [B12] The definition shall be consistent with the requirements of ISO 11179-4 Section  
1566 4.4 and will provide an understandable meaning, which should also be  
1567 translatable to other languages.

1568 [B13] The definition shall take into account the fact that the users of the *Business*  
1569 *Information Entity* dictionary are not necessarily native English speakers. It shall  
1570 therefore contain short sentences, using normal words. Wherever synonym terms  
1571 are possible, the definition shall use the preferred term as identified in the  
1572 *Controlled Vocabulary*.



- 1573 [B14] The definition of a *Basic Business Information Entity* shall use a structure that is  
1574 based on the existence of the *Object Class Term*, the *Property Term*, and the  
1575 *Representation Term*, and enhanced by business related *Qualifier Terms*.
- 1576 [B15] The definition of an *Association Business Information Entity* shall use a structure  
1577 that is based on the existence of the *Object Class Term*, the *Property Term* and the  
1578 *Object Class Term* of the *Aggregate Business Information Entity* on which the  
1579 corresponding *Association Business Information Entity Property* is based, and  
1580 enhanced by business related *Qualifier Terms*.
- 1581 [B16] Whenever both the definite (i.e. the) and indefinite article (i.e. a) are possible in a  
1582 definition, preference shall be given to the indefinite article (i.e. a).
- 1583 **6.1.4.2.4 Rules for Business Information Entity Dictionary Entry Names**
- 1584 [B17] The *Dictionary Entry Name* shall be unique.
- 1585 [B18] The *Dictionary Entry Name* shall be extracted from the *Business Information*  
1586 *Entity* definition.
- 1587 [B19] The *Dictionary Entry Name* shall be concise and shall not contain consecutive  
1588 redundant words.
- 1589 [B20] The *Dictionary Entry Name* and all its components shall be in singular form  
1590 unless the concept itself is plural.
- 1591 [B21] The *Dictionary Entry Name* shall not use non-letter characters unless required by  
1592 language rules.
- 1593 [B22] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e. no  
1594 words like *and*, *of*, *the*, etc.). This rule shall be applied to the English language,  
1595 and may be applied to other languages as appropriate.
- 1596 [B23] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall be  
1597 expanded or explained in the definition.
- 1598 [B24] The *Dictionary Entry Name* of a *Basic Business Information Entity* shall consist  
1599 of the following components in the specified order:
- 1600       ▪ the *Object Class Term* of the corresponding *Basic Core Component*, and  
1601       additional *Qualifier Term(s)*,
  - 1602       ▪ the *Property Term* of the corresponding *Basic Core Component*, and possibly  
1603       additional *Qualifier Term(s)*,

1604           ▪ the *Representation Term* of the *Data Type* on which the corresponding *Basic*  
1605            *Business Information Entity Property* is based.

1606       [B25] The *Dictionary Entry Name* of an *Association Business Information Entity* shall  
1607       consist of the following components in the specified order:

1608           ▪ the *Object Class Term* of the corresponding *Association Core Component*,  
1609           and possibly additional *Qualifier Term(s)*,

1610           ▪ the *Property Term* of the corresponding *Association Core Component*, and  
1611           possibly additional *Qualifier Term(s)*,

1612           ▪ the *Object Class Term* of the *Association Business Information Entity* on  
1613           which the corresponding *Association Business Information Entity Property* is  
1614           based.

1615       [B26] The *Object Class Term*, *Property Term*, and *Representation Term* components of  
1616       a *Dictionary Entry Name* shall be separated by dots. The space character shall  
1617       separate words in multi-word *Object Class Terms* and/or multiword *Property*  
1618       *Terms*, including their *Qualifier Terms*. Every word shall start with a capital  
1619       letter. *Qualifier Terms* shall be separated from their associated *Object Class* or  
1620       *Property Term* by an underscore ( \_ ) followed by a space to separate each  
1621       qualifier. To allow spell checking of the words in the *Dictionary Entry Name*, a  
1622       space character shall follow the dots after *Object Class Term(s)* and *Property*  
1623       *Term(s)*.

1624       [B27] *Qualifier Terms* shall precede the associated *Object Class Term* or *Property Term*.  
1625       The order of qualifiers shall not be used to differentiate *Dictionary Entry Names*.

1626       [Example]

1627       In the *Business Information Entity* entitled **Cost. Budget Period\_ Total.**  
1628       **Amount**, the component *Budget Period* is a *Qualifier Term* for the *Property Term* of  
1629       *Total*. This is derived from the *Core Component* of **Cost. Total. Amount.**

1630       [B28] The name of a qualified *Object Class* refers to an activity or object within a  
1631       *Business Context*. It shall be unique throughout the dictionary and may consist of  
1632       more than one word.

1633       [B29] For *Basic* and *Association Business Information Entities*, if the *Property Term*  
1634       uses the same (or equivalent) word or words as the third component of the  
1635       *Dictionary Entry Name*, and the *Property Term* is not qualified, the redundant  
1636       word(s) in the *Property Term* shall be removed from the *Dictionary Entry Name*.

1637       [B30] The *Dictionary Entry Name* of an *Aggregate Business Information Entity* shall  
1638       consist of the name of the *Object Class* of its associated *Aggregate Core*

1639 *Component* and additional *Qualifier Term(s)* to represent its specific *Business*  
1640 *Context*, followed by a dot, a space character, and the term *Details*.

#### 1641 6.1.4.2.5 Rules for Business Information Entity Business Terms

1642 *Business Information Entity Business Terms* are those terms that are commonly used for  
1643 day-to-day information exchanges within a given domain. As such, no specific naming  
1644 rules apply to *Business Terms*. Interoperability of *Business Terms* will be given by  
1645 linking them to the formalised names of the corresponding *Business Information Entity*  
1646 dictionary entries.

#### 1647 6.1.4.3 Rules for Data Types

##### 1648 6.1.4.3.1 Data Type Dictionary Information

1649 Each *Data Type* contains the following dictionary information that is impacted by the  
1650 naming rules:

- 1651 • **Dictionary Entry Name** (Mandatory). This is the unique official name of the  
1652 *Data Type* in the dictionary.
- 1653 • **Definition** (Mandatory). This is the unique semantic meaning of that *Data*  
1654 *Type*.

1655 The *Data Type* naming rules are also based on the following concepts as defined in ISO  
1656 11179:

- 1657 • **Representation Term**. This defines the type of valid values for an information  
1658 entity.
- 1659 • **Qualifier Term**. A word or words which help define and differentiate a *Data*  
1660 *Type* from its associated *Core Component Type* and other *Data Types*.

##### 1661 6.1.4.3.2 Data Type General Rules

1662 [D3] The dictionary content shall be in English Language following the primary  
1663 *Oxford English Dictionary* English spellings to assure unambiguous spelling.

##### 1664 6.1.4.3.3 Data Type Rules for Definitions

1665 [D4] The definition shall be consistent with the requirements of ISO 11179-4 Section  
1666 4.4 and shall provide an understandable meaning, which should also be  
1667 translatable to other languages.

1668 [D5] The definition shall take into account the fact that the users of the *Data Type*  
1669 *Dictionary* are not necessarily native English speakers. It shall therefore contain  
1670 short sentences, using normal words. Wherever synonym terms are possible, the  
1671 definition shall use the preferred term as identified in the *Controlled Vocabulary*.

- 1672 [D6] The definition of a *Data Type* shall use a structure that is based on the existence  
 1673 of primary and secondary *Representation Terms* of the associated *Core*  
 1674 *Component Type*, and is enhanced by *Qualifier Terms*.
- 1675 [D7] Whenever both the definite (i.e. the) and indefinite article (i.e. a) are possible in a  
 1676 definition, preference shall be given to the indefinite article (i.e. a).
- 1677 6.1.4.3.4 Rules for Data Type Dictionary Entry Names
- 1678 [D8] The *Dictionary Entry Name* shall be unique.
- 1679 [D9] The *Dictionary Entry Name* shall be extracted from the *Data Type* definition.
- 1680 [D10] The *Dictionary Entry Name* shall be concise and shall not contain consecutive  
 1681 redundant words.
- 1682 [D11] The *Dictionary Entry Name* shall not use non-letter characters unless required by  
 1683 language rules.
- 1684 [D12] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e. no  
 1685 words like *and*, *of*, *the*, etc.). This rule shall be applied to the English language,  
 1686 and may be applied to other languages as appropriate.
- 1687 [D13] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall be  
 1688 expanded or explained in the definition.
- 1689 [D14] The *Dictionary Entry Name* of a *Data Type* shall consist of a *Representation*  
 1690 *Term* preceded by *Qualifier Term(s)*, followed by a dot, a space character, and the  
 1691 term *Type*. The space character shall separate words in multi-word *Qualifier*  
 1692 *Terms* and *Representation Terms*. Each *Qualifier Term* shall be followed by an  
 1693 underscore. To allow spell checking of the words in the *Dictionary Entry Name*, a  
 1694 space character shall follow the underscores after *Qualifier Terms*.

1695 [Example]

1696 **Country\_ Identifier. Type**

- 1697 [D15] In the *Dictionary Entry Name* of a *Data Type*, the name of the *Representation*  
 1698 *Term* shall be one of the primary or secondary terms specified in the *List of*  
 1699 *Permissible Representation Terms* as included in this specification (See section  
 1700 8.3).

1701 [Note]

1702 Whereas the name of the *Core Component Type* shall only be based on a primary  
1703 *Representation Term*, the *Representation Term* that is used in the *Dictionary Entry Name*  
1704 of a *Data Type* can also be a secondary *Representation Term*. This will be the case when  
1705 the *Data Type* restricts the *Core Component Type* in such a way that it only covers a part  
1706 of the full semantic meaning of the primary *Representation Term*.

#### 1707 6.1.4.3.5 List of Permissible Representation Terms

1708 The *Representation Term* is the part of a *Core Component* name that describes the form  
1709 of valid values in which the business information is expressed in a data item. For instance  
1710 all *Basic Core Components* representing a monetary amount shall be named *[Name]*.  
1711 *[Qualifier]\_Amount* where *[Name]* represents a specialisation of the generic amount,  
1712 *[Qualifier]* specifies a restriction of the possible values and *Amount* is the *Representation*  
1713 *Term*. Table 8-3 lists the permissible *Representation Terms*.

1714 [Note]

1715 Table 8-3 may subsequently be published separately to facilitate maintenance outside the  
1716 body of this specification.

1717 [C33] When a *Representation Term* contains more than one word, and the specific use  
1718 of the *Representation Term* requires only one word, the other word(s) in the  
1719 *Representation Term* may be dropped.

1720 [Example]

1721 For the *Core Component* entitled **Product Service Start. DateTime**, the  
1722 *Representation Term* is DateTime and the *Core Component* is defined as a date and/or  
1723 time on which a product/service starts. The *Representation Term* remains DateTime. For  
1724 the *Core Component* **Payment Card. Expiration. Date**, the *Representation Term*  
1725 is still DateTime, however since the specific use of the *Representation Term* requires  
1726 only date, the word time is dropped.

#### 1727 6.1.5 Catalogue of Core Components

1728 As originally articulated in the ebXML architecture concept and perpetuated in the  
1729 developing UN/CEFACT architecture concept, all *Core Components* will be recorded in  
1730 an ebXML compliant registry and stored in a related repository. However, small and  
1731 medium enterprise (SME) organisations may not be able to readily access such  
1732 architecture. As such, it is important that the full range of UN/CEFACT *Core*  
1733 *Components* be published in a freely available catalogue. This catalogue must convey the  
1734 full details of each *Core Component* consistent with how those components are stored as

1735 UML objects in the registry/repository. Table 6-1 identifies a proper format for the  
 1736 catalogue and contains representative entries from the existing UN/CEFACT *Core*  
 1737 *Components Catalogue*.

1738 **Table 6-1. Core Component Catalogue Format Example**

Temporary Identifier	Dictionary Entry Name	Type of Core Component - Basic, Association, Aggregate	Definition	Comments	Object Class Term	Property Term	Type (Data Type or Object Class Term)	Business Terms	Core Component Properties
000024	Address. Type. Code	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	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. Identifier	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	Identifier		
000012	Birth. Date	Basic	The date on which a person was born.	Applies only to parties being natural persons.	Birth	Date	DateTime		

1739 [Note]

1740 In Table 6-1, the \* in the *Property Term* column indicates cases where the *Property Term*  
 1741 is the same as either the *Representation Term* or *Object Class Term*, and is consequently  
 1742 dropped from the *Dictionary Entry Name*.

1743 The catalogue is intended to be part of a larger *Core Component Library*. The *Core*  
 1744 *Component Library* will consist of the following parts:

- 1745 • *Core Component Types and Data Types*
- 1746 • *Core Component Catalogue, including Basic Core Components, Association*  
 1747 *Core Components, and Aggregate Core Components*
- 1748 • *Catalogue of Business Information Entities*

### 1749 **6.1.6 Catalogue of Business Information Entities**

1750 For the same reasons that a *Core Components Catalogue* is necessary, a *Catalogue of*  
1751 *Business Information Entities* is also required. Predefined *Business Information Entities*  
1752 are not provided in this specification. Rather, the working registries and the groups  
1753 defining business messages will be responsible for developing a *Catalogue of Business*  
1754 *Information Entities* that will include *Basic*, *Association*, and *Aggregate Business*  
1755 *Information Entities*.

## 1756 **6.2 Context**

1757 This section fully describes applicable rules and applications for the use of *Context* in  
1758 *Core Component* discovery, analysis, and use to include *Context Categories* and their  
1759 values, and the *Constraint Language*.

### 1760 **6.2.1 Overview of Context Specification**

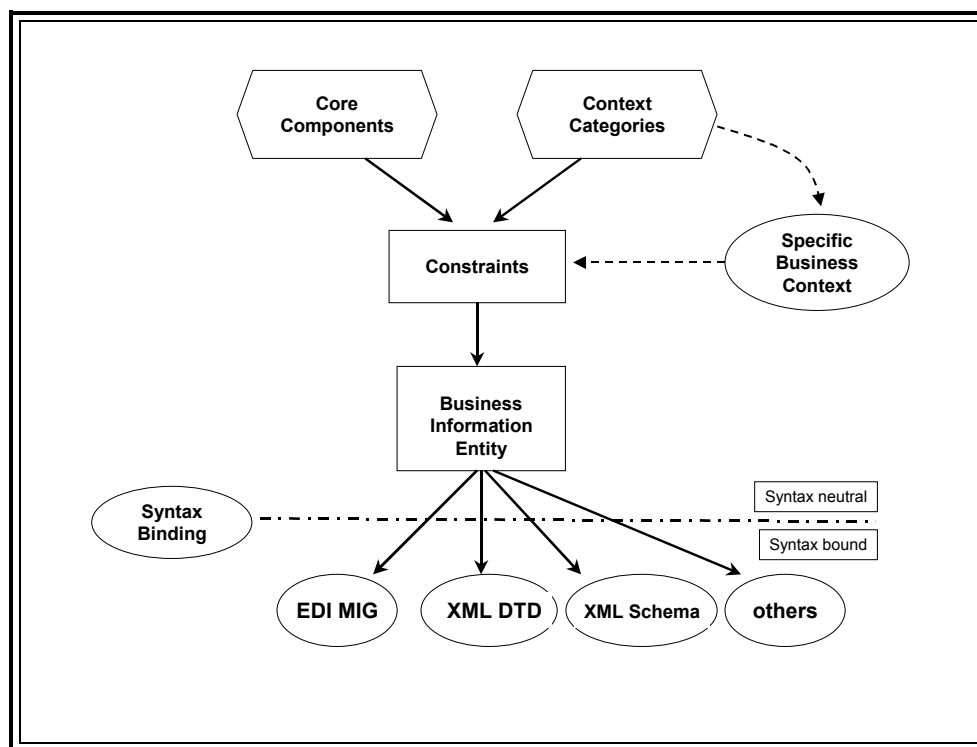
1761 Whenever business collaboration takes place between specific trading partners, data is  
1762 exchanged in the form of business messages. When used as such, that data exists in a  
1763 particular *Business Context*. In its simplest form, this is the idea of *Context* as used in  
1764 ebXML. The *Context* in which the business collaboration takes place can be specified by  
1765 a set of categories and their associated values.

1766 The *Core Components* have no *Context* independent of their use. The *Context* mechanism  
1767 provides a full semantic qualification for the *Core Component* used in a *Business*  
1768 *Process*. Figure 6-3 shows how the *Constraint Language* applies *Business Context*  
1769 *Categories* and specific *Business Context(s)* to *Core Components* to develop *Business*  
1770 *Information Entities*. Qualification is to be interpreted as Specialisation as defined in  
1771 UML. Qualification narrows the semantic concept to a more specific one. The structure  
1772 of qualified *Business Information Entities* may be a subset (but never a superset) of the  
1773 structure of the (unqualified) *Business Information Entities* or *Core Components* they are  
1774 based on. That means that value ranges may be restricted, components may be removed  
1775 or their repetition factor may be lowered and *Cardinality* may change from optional to  
1776 mandatory. The *Business Information Entity* resulting from this process can be  
1777 manifested as a model, which in turn can be used as the basis of a syntax-bound business  
1778 message description (an EDI message implementation guide, an XML schema<sup>9</sup>, etc.)

1779 The following sections address the *Context Categories*, and the *Constraint Language*  
1780 more closely.

---

<sup>9</sup> The term XML Schema includes XML Schema as defined in World Wide Web Consortium Extensible Markup language version 1.0, XML Document Type Definitions, Schematron, SOX, Relax NG, ASN.1, XDR or any other notation that specifies the form and information content of an XML document.

1781 **Figure 6-3. Operation of The Context Mechanism**

1782

1783 **6.2.1.1 Context Categories**

1784 *Context Categories* exist to allow users to uniquely identify and distinguish between  
 1785 different *Business Contexts*. Eight *Context Categories* have been identified (Table 6-2).  
 1786 Each of the identified categories, unless otherwise stated, uses a standard classification to  
 1787 provide values for the category. Constraint rules, and therefore *Business Information*  
 1788 *Entities*, are tied to a particular set of standard classifications for identifying and  
 1789 distinguishing *Contexts*.

1790 **6.2.1.2 Constraint Language**

1791 A *Constraint Language* is used to express the relationship between specific *Business*  
 1792 *Contexts* and how semantics are applied to the *Core Components* to produce *Business*  
 1793 *Information Entities*. The scope of this language covers two functional parts:

- 1794 • *Assembly* of a large aggregate (the *Document*). The *Constraint Language*  
 1795 addresses how *Assembly* is done. It does not address the design or design  
 1796 principles of business document assembly. That subject will be covered by the  
 1797 *Message Assembly* supplemental document.
- 1798 • Refinement of the assembly as appropriate. Refinement is both the addition of  
 1799 semantics specific to the *Business Process*, and the restriction of the semantic  
 1800 model.

1801 This separation is a convenience for implementation (it simplifies the development of  
 1802 processing tools) and development of standard assemblies that can then be refined by  
 1803 specific users (akin to how EDI standards and message implementation guides function  
 1804 today).



1805 The *Constraint Language* allows, for example, simple commands indicating how *Core*  
1806 *Components* will be used, how they will be named for these specific uses, and how to  
1807 refine the *cardinality* (if necessary). Further, conditional relationships can be expressed.  
1808 Specific *Context* values or sets of values can be tied to the actions performed on *Core*  
1809 *Components* to produce *Business Information Entities*.

1810 [Example]

1811 If the *Geopolitical Context* has a value of *Anywhere in the European Union*, and the  
1812 specific *Business Context Value* indicates that the *Business Process* occurs in France,  
1813 then the *Context-appropriate Business Information Entity* can be assembled by modifying  
1814 the correct *Core Component*.

1815 The *Constraint Language* would say—If the *Geopolitical Context* equals the *European*  
1816 *Union*, then take the core *NameAddress* component and rules to provide the correct  
1817 names, *cardinality*, and arrangement to the fields. To do business in France, the specific  
1818 *Context* value for that process will trigger this rule, giving a set of appropriate business  
1819 semantics (*Business Information Entities*).

### 1820 6.2.1.3 Syntax Binding

1821 The *Business Information Entity* in its standard form is a model that has no specific  
1822 relationship to any given syntax. A given *Business Information Entity* can subsequently  
1823 be expressed in any of a number of syntaxes through a binding process. This process is  
1824 called *syntax binding*, and is independent of (has no relationship to) a specific syntax.  
1825 The *Syntax Binding* process does not alter the semantics of the *Business Information*  
1826 *Entity*, but simply instantiates the *Business Information Entity* for use in syntax specific  
1827 documents. It may be possible to express *syntax binding* in an algorithm.

1828 [B31] *Syntax Binding* shall not change the semantics of a *Business Information Entity*.

### 1829 6.2.2 Approved Context Categories

1830 Table 6-3 contains the eight approved *Context Categories*.

1831 [C34] When describing a specific *Business Context*, a value or set of values shall be  
1832 assigned to each of the approved *Context Categories* in order to describe the  
1833 business situation in an unambiguous and formal way.

1834 **Table 6-3. Approved Context Categories**

<b>Context Category</b>	<b>Description</b>
<i>Business Process</i>	The <i>Business Process</i> name(s) as described using the <i>UN/CEFACT Catalogue of Common Business Processes</i> as extended by the user.
<i>Product Classification</i>	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)
<i>Industry Classification</i>	Semantic influences related to the industry or industries of the trading partners (e.g., product identification schemes used in different industries).
<i>Geopolitical</i>	Geographical factors that influence business semantics (e.g., the structure of an address).
<i>Official Constraints</i>	Legal and governmental influences on semantics (e.g. hazardous materials information required by law when shipping goods).
<i>Business Process Role</i>	The actors conducting a particular <i>Business Process</i> , as identified in the <i>UN/CEFACT Catalogue of Common Business Processes</i> .
<i>Supporting Role</i>	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.)
<i>System Capabilities</i>	This <i>Context Category</i> exists to capture the limitations of systems (e.g. an existing back office can only support an address in a certain form).

1835 **6.2.2.1 Business Process Context**

1836 In describing a business situation, generally the most important aspect of that situation is  
1837 the business activity being conducted. *Business Process Context* provides a way to  
1838 unambiguously identify the business activity. To ensure consistency with *Business*  
1839 *Process* activities, it is important to use a common point of reference. The definitive point  
1840 of reference for international standards is the *UN/CEFACT Catalogue of Common*  
1841 *Business Processes*.

1842 [C35] Assigned *Business Process Contexts* shall be from the standard hierarchical  
1843 classification: provided as part of the *UN/CEFACT Catalogue of Common*  
1844 *Business Processes*.

- 1845 [C36] *Business Process Context* values may be expressed as a single *Business Process*,  
1846 or as a hierarchical set of *Business Processes*.
- 1847 [C37] *Business Process Context* values may be taken from extensions to the *Business*  
1848 *Processes* described in the *UN/CEFACT Catalogue of Common Business*  
1849 *Processes* as provided for in that document.
- 1850 [C38] When *Business Process* extensions are used, they shall include full information  
1851 for each value sufficient to unambiguously identify which extension is providing  
1852 the value used.
- 1853 **6.2.2.2 Product Classification Context**
- 1854 The *Product Classification Context* describes those aspects of a business situation related  
1855 to the goods or services being exchanged by, or otherwise manipulated, or concerned, in  
1856 the *Business Process*. Recognized code lists exist that provide authoritative sources of  
1857 *Product Classification Contexts*.
- 1858 [C39] A single value or set of values may be used in a *Product Classification Context*.
- 1859 [C40] If a hierarchical system of values is used for *Product Classification Context*, then  
1860 these values may be at any level of the hierarchy.
- 1861 [C41] If more than one classification system is being employed, an additional value  
1862 specifying which classification scheme has supplied the values used shall be  
1863 conveyed.
- 1864 [C42] *Product Classification Context* code values shall be taken from recognized code  
1865 lists to include:
- 1866 • *Universal Standard Product and Service Specification (UNSPSC)*
  - 1867 - Custodian: Electronic Commerce Code Management Association  
1868 (ECCMA)
  - 1869 • *Standard International Trade Classification (SITC Rev .3)*
  - 1870 - Custodian: United Nations Statistics Division (UNSD)
  - 1871 • *Harmonized Commodity Description and Coding System (HS)*
  - 1872 - Custodian: World Customs Organization (WCO)
  - 1873 • *Classification Of the purposes of non Profit Institutions serving households*  
1874 (COPI)
  - 1875 - Custodian: UNSD (This provides a mapping between the first three.)

## 1876 6.2.2.3 Industry Classification Context

1877 The *Industry Classification Context* provides a description of the industry or sub-industry  
1878 in which the *Business Process* takes place.

1879 [C43] An *Industry Classification Context* may contain a single value or set of values at  
1880 any appropriate level of the value hierarchy.

1881 [C44] The *Industry Classification Context* value hierarchy must be identified.

1882 [C45] *Industry Classification Context* code values shall be taken from recognized code  
1883 lists to include:

- 1884 • *International Standard Industrial Classification (ISIC)*
- 1885 - Custodian: UNSD
- 1886 • *Universal Standard Product and Service Specification (UNSPSC) Top-level*  
1887 *Segment [digits 1 and 2] used to define industry.*
- 1888 - Custodian: ECCMA

1889 [Note]

1890 There are many other industry classification schemes that may be used for *Industry*  
1891 *Classification Context*.

## 1892 6.2.2.4 Geopolitical Context

1893 *Geopolitical Contexts* allow description of those aspects of the *Business Context* that are  
1894 related to region, nationality, or geographically based cultural factors.

1895 [C46] *Geopolitical Context* shall consist of appropriate continent, economic region,  
1896 country, and region identifiers.

1897 [C47] *Geopolitical Context* may associate one or more values with any business  
1898 message or component.

1899 [C48] *Geopolitical Context* shall employ the following hierarchical structure:

1900 Global

1901 [Continent]

1902 [Economic Region]

1903 [Country] - ISO 3166.1

1904 [Region] - ISO 3166.2

1905 [C49] At any level of the *Context* hierarchy, a value may be a single value, a named  
1906 aggregate, or cross-border value.

1907 [C50] *Geopolitical Context* hierarchy values shall be structured as follows:

- 1908           • **Single Value:** A single value indicating a single continent, economic region,  
1909           country, or region, depending on position within the hierarchy.
- 1910           • **Named Aggregate:** A related group of values (which may themselves be  
1911           single values, named aggregates, or cross-border pairs of values), which have  
1912           been related and assigned a name. A named aggregate contains at least two  
1913           values.
- 1914           • **Cross-Border:** One or more pairs of values, designated *To*, *From*, or *Bi-*  
1915           *directional*, indicating the direction of cross-border *Context*. Values may be  
1916           named aggregates or single values.

1917 [Example]

1918 The following example shows an extract of the basic, single-value hierarchy of  
1919 recommended values, based on the common ISO 3166.1 *Country Codes*. (The value at  
1920 the top of any hierarchy is always understood to be *Global*.)

1921 Europe

1922       Eastern Europe

1923               AL – ALBANIA

1924               AM – ARMENIA

1925 [C51] Points in the *Geopolitical Context* hierarchy shall be specified by the use of the  
1926       node value, or by the full or partial path.

1927 [C52] The full path of the *Geopolitical Context* hierarchy must be used to understand the  
1928       hierarchy when complex constructs are employed.

1929 [C53] A specific level in the *Geopolitical Context* hierarchy is understood to inherit all  
1930       of the properties within its specific hierarchical path except where otherwise  
1931       specified.

1932 [C54] *Geopolitical Context* values shall be taken from ISO 3166.1 and 3166.2

### 1933 6.2.2.5 Official Constraints Context

1934 The *Official Constraints Context Category* describes those aspects of the business  
1935 situation that result from legal or regulatory requirements and similar official categories.  
1936 This category contains two distinct parts:

- 1937           • Regulatory and Legislative. These are normally unilateral in nature and  
1938           include such things as Customs Authority regulations.

- 1939           • Conventions and Treaties. These are normally bi- or multilateral agreements  
1940           and as such are different from regulatory and legislative constraints.
- 1941   [C55] The *Official Constraints Context* shall consist of at least two values:
- 1942           • Identification of the legal or other classification used to identify the *Context*  
1943           values.
- 1944           • Identification of the official constraint itself. These values may represent a  
1945           hierarchical structure depending on the official constraints system being  
1946           referenced.
- 1947   Because there is no known global classification of all *Official Constraints Contexts* as  
1948   used here, any implementation must provide a set of recognized official constraints  
1949   classifications for use within the appropriate *Core Components Registry* implementation.
- 1950   [C56] Individual *Core Component* implementations shall register used official constraint  
1951   classification schemes with the appropriate supporting *Core Components Registry*  
1952   implementation.
- 1953   **6.2.2.6   Business Process Role Context**
- 1954   The *Business Process Role Context* describes those aspects of a business situation that are  
1955   specific to an actor or actors within the *Business Process*. Its values are taken from the set  
1956   of *Role* values provided by the *UN/CEFACT Catalogue of Common Business Processes*.  
1957   A *Business Process Role Context* is specified by using a value or set of values from this  
1958   source.
- 1959   [C57] *Business Process Role Context* values shall be taken from an approved list  
1960   provided by the *Business Process* model library being employed.
- 1961   [C58] The *UN/CEFACT Catalogue of Common Business Processes* shall be the  
1962   definitive source of *Business Process Role Context* values for all UN/CEFACT  
1963   *Business Information Entities*.
- 1964   **6.2.2.7   Supporting Role Context**
- 1965   The *Supporting Role Context* identifies those parties that are not active participants in the  
1966   *Business Process* being conducted but who are interested in it. A *Supporting Role*  
1967   *Context* is specified with a value or set of values from a standard classification.
- 1968   [C59] *Supporting Role Context* values shall be taken from the UN/EDIFACT *Code List*  
1969   *for DE 3035 Party Roles*.

1970

[Note]

1971

Users are cautioned that duplication exists in the current version of the required code list.

1972

UN/CEFACT will review this code list to clarify duplicates and identify non-*Supporting*

1973

*Role Context* values.

1974

**6.2.2.8 System Capabilities Context**

1975

This category identifies a system, a class of systems or standard in the business situation.

1976

The *System Capabilities Context* requires a least one pair of values: an identification of

1977

the classification scheme being used and a value from that scheme. A valid *System*

1978

*Capabilities Context* may include more than one such pair of values.

1979

[C60] *Systems Capabilities Context* values shall consist of pairs of values. Each pair

1980

shall be comprised of an identification of the referenced classification scheme and

1981

the value(s) being employed.

1982

[Note]

1983

There is no known classification of all types of information systems and standards. It is

1984

recommended that a mechanism for the registration of system and standard names be

1985

provided by the ebXML registry, as valid values for the *System Capabilities Context*.

1986

**6.2.3 Context Values**

1987

A specific *Business Context* is formally described using a set of *Context* values. Every

1988

*Context Category* must have a valid value, even if this value is *In All Contexts* or *None*.

1989

The value *None* is appropriate for *Official Constraints Context* because there will be

1990

instances where there are no official constraints.

1991

[C61] The *In All Contexts* value shall be a valid value for every *Context Category* except

1992

for *Official Constraints Context*.

1993

[C62] The value *None* shall be a valid value for *Official Constraints Context*.

1994

**6.2.4 Core Components Context Constraints Language**

1995

The *Core Components Context Constraints Language* consists of a set of constructs (See

1996

Table 6-3) that allow users to express the relationships between specific business

1997

situations and the specific structure and meaning of business data used in that situation.

1998

The *Constraints Language* refers to specific *Contexts* as described in the *Context*

1999

*Categories* specification and uses unique identifiers to refer to *Core Components*

2000

semantic models. The constraints applied to *Core Components* in specific *Business*

2001

*Contexts* to generate *Business Information Entities* are expressed using the *Constraints*

2002

*Language*.

2003

[Note]

2004

The ebXML *Unique Identifier* is fully described in the *ebXML Technical Architecture Specification Version 1.04*. Its construct is fully specified in the *ebXML Registry Specification 2.0*.

2005

2006

2007

[C63] The *Core Components Context Constraints Language* shall be used to describe the constraints being applied to *Core Components* to develop *Business Information Entities*.

2008

2009

2010

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

2011

2012

2013

2014

[C64] *Assembly* shall be the top-level construct in any set of *Assembly Rules*.

2015

The *ContextRules* construct is the overall expression of a single set of rules that are used to apply *Context* to *Core Components*. The *ContextRules* add the full semantic and structural refinement to the *Core Components* to produce *Business Information Entities*.

2016

2017

2018

This mechanism supports specifying *cardinality*, addition and subtraction of child *Core Components*, renaming of those children, assigning *Business Information Entity* names to the *Context*-specific instances of the *Core Components*, and adding structure to develop *Aggregate Business Information Entities*.

2019

2020

2021

2022

[C65] A single set of *Context* rules shall be described using the *ContextRules* expression.

2023

2024

**Table 6-3 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.



<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
<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

<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
<b>CreateBIE (Continued)</b>	Condition	Condition under which this rule should 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>		<i>ContextRules</i> contains one or more Rules <b>Note:</b> A <i>ContextRules</i> is the top level construct in a set of <i>Context Rules</i>
	Rule	List of refinement and qualification rules to be applied
	@id	Id of the <i>ContextRules</i> rule
	@idref	Reference to the ID of another <i>ContextRules</i> rule
	@version	Version of the <i>ContextRules</i> 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 <i>Context</i> category to which this Taxonomy applies
	@id	Id of the Taxonomy rule
	@idref	Reference to the ID of another Taxonomy rule

<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
<b>Condition</b>		A Condition contains at least one of Action or Condition or Occurs, one @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
	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 addition should occur.
	CreateGroup	Create a group of BIEs
	BIE	Adds a new BIE to the content model.

<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
<b>Add</b> (Continued)	Attribute	Adds a new attribute to the content model
	Annotation	Insert Annotation
	@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 attribute from the content model
	@id	Id of the Subtract rule
	@idref	Reference to the ID of another Subtract rule
		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
		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
<b>BIE</b>	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
	Annotation	Insert Annotation
	Name	Name of attribute to be modified

<b>Construct</b>	<b>Component Constructs</b>	<b>Description</b>
<b>Attribute (Continued)</b>	Type	Type of the attribute (representation class)
	Use	Indicates whether required or optional, and if the latter whether required or optional. If optional, indicates the presence of a default. May supply a fixed value instead.
	Value	Indicates whether required or optional, and if the latter whether required or optional. If optional, indicates the presence of a default. May supply a value to be modified
	@applyTo	Node to apply action to
	@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. An @name is required in any UseBIE that does not use a CreateGroup.
	@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. This design is intended to mirror the annotation functionality found in the W3C Schema Specification.
	@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

Construct	Component Constructs	Description
<b>Annotation (Continued)</b>	@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

2025

2026

[Note]

2027

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

2028

#### 2029 6.2.4.1 Assembly Construct

2030 The *MinOccurs* and *MaxOccurs* constructs in the *CreateBIE* construct specify the  
 2031 occurrence that the created *Business Information Entity* will have in the resulting  
 2032 semantic model.

2033 [C66] A *Business Information Entity* created with *MinOccurs* = 1 and *MaxOccurs* = 1  
 2034 shall be specified in the resulting semantic model as occurring only once.

2035 [C67] An *Assembly* may contain more than one assembled top-level semantic model.

#### 2036 6.2.4.2 ContextRules Construct

2037 Several built-in variables are used to access *Context* information. These variables  
 2038 correspond to the identified *Context Categories*. All of these variables have string values.

2039 [C68] The *Apply* attribute of the *ContextRules* construct type shall be used for  
 2040 determining the behaviour of rules that use hierarchical values.

2041 [C69] Allowed *Apply* attribute values are:

- 2042 • *exact* - match only if the value in the provided *Context* is precisely the same as  
 2043 that specified in the rule
- 2044 • *hierarchical* - match if the value provided is the same or a child of that  
 2045 specified in the rule.

2046

[Example]

2047

2048

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

2049

2050

[C70] The *Attribute* construct has four optional children in its content model, of which at least one must be present.

2051

2052

[C71] When the *Attribute* construct is used to refine an existing *Attribute*, then a value must be specified for *@applyTo* on that *Attribute* construct.

2053

2054

[C72] *ContextRules* must refer to the names of the *Core Components*, and not the names given to the resulting *Business Information Entities* elsewhere in the Rules.

2055

[Example]

2056

2057

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

2058

#### 6.2.4.3 Output Constraints

2059

2060

2061

[C73] Semantic models and document definitions produced through the application of *Assembly* and *Context Rules* must contain the metadata about the rules and *Context* that produced them.

2062

#### 6.2.4.4 Ordering and Application

2063

2064

2065

2066

2067

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

2068

2069

[C74] The *Order* property on the *Rule* construct shall determine the sequence for the application of the applicable set of rules.

2070

[C75] Two *Rule* constructs must not have the same value for the property *Order*.

2071 **7 Technical Details - Core Component**  
2072 **Registry/Repository Storage**

2073 Section 6 specified the basic definitions for *Core Components*, *Data Types*, *Business*  
2074 *Information Entities* and *Context*.

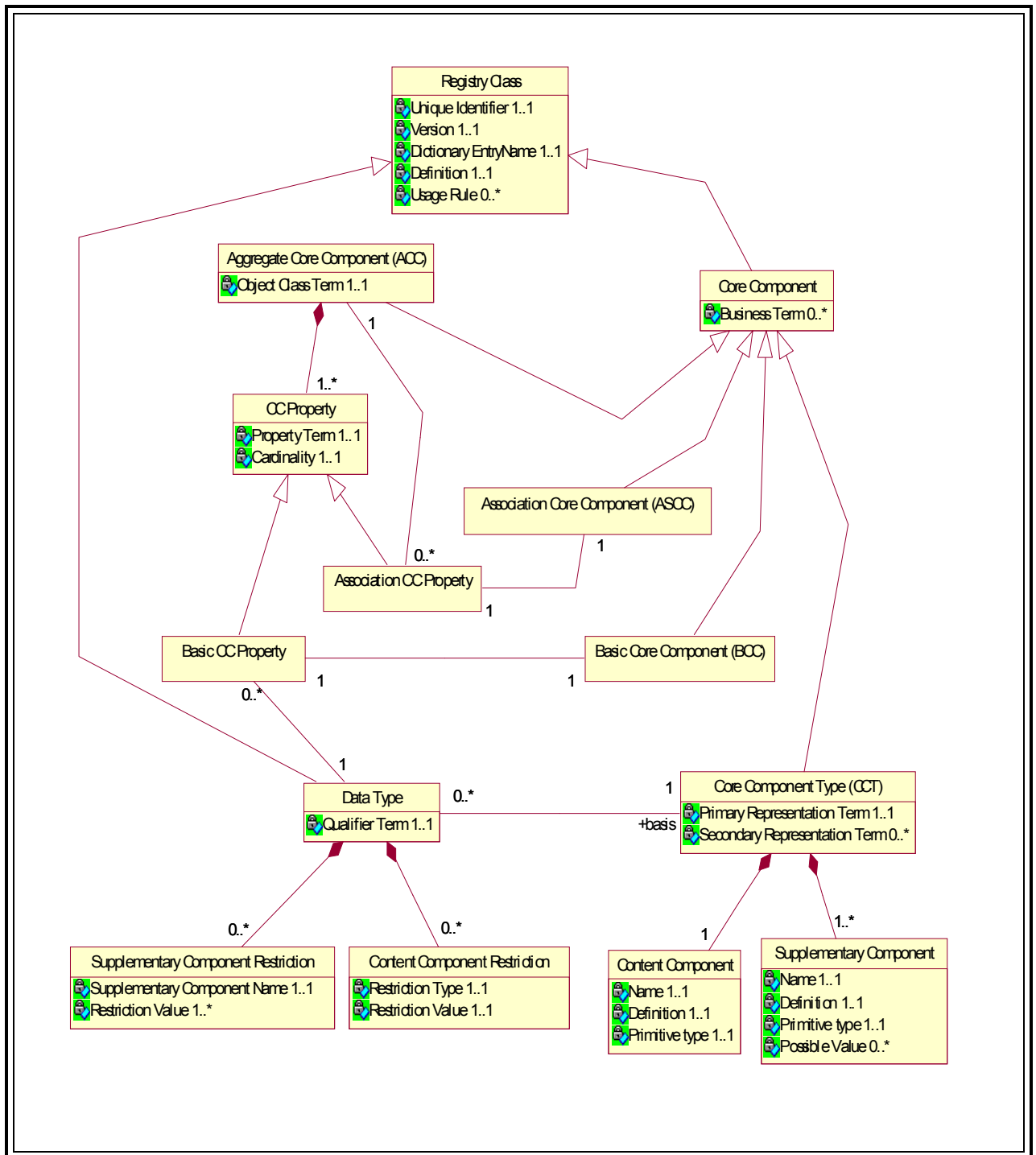
2075 This section details exact information required for design of *Unified Modeling*  
2076 *Language* objects to store *Core Components*, *Data Types*, *Business Information*  
2077 *Entities*, *Context* and relevant associated metadata in the registry/repository. Both  
2078 parts contain requirements that must be addressed by developers and users of *Core*  
2079 *Components*. Further, both parts contain requirements that must be satisfied in the  
2080 supported registry and repository suite of technical specifications and any  
2081 corresponding overarching information technology framework that uses *Core*  
2082 *Components* as the linchpin between process modelling and trade.

2083 **7.1 Storing Core Components**

2084 This section fully describes *Core Component* storage details. Figure 7-1 is the *Unified*  
2085 *Modeling Language* model of all aspects of *Core Components* and fully describes the  
2086 types of *Core Components* and their relationships as a requirement of storage.



2087 **Figure 7-1. Core Components and Data Types - Full Definition**



2088  
2089

2090 **7.1.1 Stored Core Components**

2091 [S1] *Core Components* are a particular category of *Registry Classes*. As such, all  
2092 stored *Core Components* shall include the following attributes:

- 2093           • **Unique Identifier (mandatory):** The identifier that references a *Core*  
2094            *Component* instance in a unique and unambiguous way.
- 2095           • **Version (mandatory):** An indication of the evolution over time of a *Core*  
2096            *Component* instance.
- 2097           • **Dictionary Entry Name (mandatory):** The official name of a *Core*  
2098            *Component*.
- 2099           • **Definition (mandatory):** The semantic meaning of a *Core Component*.
- 2100           • **Usage Rule (optional, repetitive):** A constraint that describes specific  
2101            conditions that are applicable to the *Core Component*.
- 2102   [S2]   Stored *Core Components* shall always be defined as one of the four recognized  
2103            types—*Basic Core Component*, *Association Core Component*, *Aggregate Core*  
2104            *Component* or *Core Component Type*.
- 2105   [S3]   Stored *Core Components* shall include the following attributes:
- 2106           • **Business Term (optional, repetitive):** A synonym term under which the  
2107            *Core Component* is commonly known and used in a business. A *Core*  
2108            *Component* may have several business terms or synonyms.
- 2109   **7.1.2 Stored Aggregate Core Components**
- 2110   [S4]   *Aggregate Core Components* are a particular category of *Core Components*.  
2111            As such, stored *Aggregate Core Components* shall include all attributes of  
2112            stored *Core Components*.
- 2113   [S5]   Stored *Aggregate Core Components* shall contain one or more *Core*  
2114            *Component Properties*.
- 2115   [S6]   Stored *Aggregate Core Components* can be referenced by one or more  
2116            *Association Core Component Properties* of other *Aggregate Core*  
2117            *Components*.
- 2118   [S7]   Stored *Aggregate Core Components* shall include the following attribute:
- 2119           • **Object Class Term (mandatory):** A semantically meaningful name for  
2120            the *Object Class* that is represented by the *Aggregate Core Component*. It  
2121            shall serve as basis for the *Dictionary Entry Name* of the *Aggregate Core*  
2122            *Component* and for the *Dictionary Entry Name* of all *Basic* and  
2123            *Association Core Components* that represent *Core Component Properties*  
2124            of this *Aggregate Core Component*.

### 2125 **7.1.3 Stored Core Component Properties**

2126 [S8] Stored *Core Component Properties* shall be stored as part of the stored  
2127 *Aggregate Core Component* to which they belong, i.e. they shall never exist  
2128 independently of their owning *Aggregate Core Component*.

2129 [S9] Stored *Core Component Properties* shall be defined as one of the two  
2130 recognized types—*Basic Core Component Property* or *Association Core*  
2131 *Component Property*.

2132 [S10] Stored *Core Component Properties* shall include the following attributes:

2133 • **Property Term (mandatory):** A semantically meaningful name for the  
2134 characteristic of the *Object Class* that is represented by the *Core*  
2135 *Component Property*. It shall serve as basis for the *Dictionary Entry Name*  
2136 of the *Basic* or *Association Core Component* that represents this *Core*  
2137 *Component Property*.

2138 • **Cardinality (mandatory):** Indication whether the *Core Component*  
2139 *Property* represents an optional, mandatory and/or repetitive characteristic  
2140 of the *Aggregate Core Component*.

### 2141 **7.1.4 Stored Basic Core Component Properties**

2142 [S11] *Basic Core Component Properties* are a particular category of *Core*  
2143 *Component Properties*. As such, stored *Basic Core Component Properties*  
2144 shall include all attributes of stored *Core Component Properties*.

2145 [S12] Stored *Basic Core Component Properties* shall be linked to the *Data Type* that  
2146 describes the possible values of the *Basic Core Component Property*.

### 2147 **7.1.5 Stored Association Core Component Properties**

2148 [S13] *Association Core Component Properties* are a particular category of *Core*  
2149 *Component Properties*. As such, stored *Association Core Component*  
2150 *Properties* shall include all attributes of stored *Core Component Properties*.

2151 [S14] Stored *Association Core Component Properties* shall be linked to the  
2152 *Aggregate Core Component* that describes the structure of the *Association*  
2153 *Core Component Property*.

### 2154 **7.1.6 Stored Basic Core Components**

2155 [S15] *Basic Core Components* are a particular category of *Core Components*. As  
2156 such, stored *Basic Core Components* shall include all attributes of stored *Core*  
2157 *Components*.

2158 [S16] Stored *Basic Core Components* shall represent a *Basic Core Component*  
2159 *Property* of a particular *Aggregate Core Component*.

### 2160 **7.1.7 Stored Association Core Components**

2161 [S17] *Association Core Components* are a particular category of *Core Components*.  
2162 As such, stored *Association Core Components* shall include all attributes of  
2163 stored *Core Components*.

2164 [S18] Stored *Association Core Components* shall represent an *Association Core*  
2165 *Component Property* of a particular *Aggregate Core Component*.

### 2166 **7.1.8 Stored Core Component Types**

2167 [S19] *Core Component Types* are a particular category of *Core Components*. As  
2168 such, stored *Core Component Types* shall include all attributes of stored *Core*  
2169 *Components*.

2170 [S20] Stored *Core Component Types* shall include one *Content Component* that  
2171 defines the *Primitive Type* and one or more *Supplementary Components* that  
2172 give meaning to the *Content Component*.

2173 [S21] Stored *Core Component Types* shall not reflect business meaning.

2174 [S22] Stored *Core Component Types* shall include the following attributes:

2175 • **Primary Representation Term (mandatory):** A semantically meaningful  
2176 name that forms the basis for the *Dictionary Entry Name* of the *Core*  
2177 *Component Type*. It can also form the basis for the *Dictionary Entry Name*  
2178 of *Data Types* that are based on the *Core Component Type*.

2179 • **Secondary Representation Term (optional, repetitive):** A semantically  
2180 meaningful name that represents a meaningful subset of the *Core*  
2181 *Component Type*. It can form the basis for the *Dictionary Entry Name* of  
2182 *Data Types* that are based on the *Core Component Type*.

### 2183 **7.1.9 Stored Supplementary Components**

2184 [S23] Stored *Supplementary Components* shall be stored as part of the stored *Core*  
2185 *Component Type* to which they belong, i.e. they shall never exist  
2186 independently of their owning *Core Component Type*.

2187 [S24] Stored *Supplementary Components* shall include the following attributes:

2188 • **Name (mandatory):** Name in the Registry of a *Supplementary Component*  
2189 of a *Core Component Type*.

2190           • **Definition (mandatory):** A clear, unambiguous and complete explanation  
 2191           of the meaning of a *Supplementary Component* and its relevance for the  
 2192           related *Core Component Type*.

2193           • **Primitive type (mandatory):** *Primitive type* to be used for the  
 2194           representation of the value of a *Supplementary Component*.

2195           [Note]

2196           Possible values for *primitive type* are String, Decimal, Integer, Boolean, Date and  
 2197           Binary.

2198           • **Possible Value (optional, repetitive):** one possible value of a  
 2199           *Supplementary Component*.

2200           [Note]

2201           Possible values shall only be stored if all possible values can be defined by an  
 2202           enumeration (e.g. list of quantity units).

## 2203    **7.1.10 Stored Content Components**

2204    [S25] Stored *Content Components* shall be stored as part of the stored *Core*  
 2205    *Component Type* to which they belong, i.e. they shall never exist  
 2206    independently of their owning *Core Component Type*.

2207    [S26] Stored *Content Components* shall include the following attributes:

2208           • **Name (mandatory):** Name in the Registry of a *Content Component* of a  
 2209           *Core Component Type*.

2210           • **Definition (mandatory):** A clear, unambiguous and complete explanation  
 2211           of the meaning of a *Content Component*.

2212           • **Primitive type (mandatory):** *Primitive type* to be used for the expression  
 2213           of the value of an instance of a *Basic Core Component* based on the  
 2214           associated *Core Component Type*.

## 2215    **7.2 Storing Data Types**

2216    This section fully describes *Data Type* storage details.

2217 **7.2.1 Stored Data Types**

2218 [S27] *Data Types* are a particular category of *Registry Classes*. As such, all stored  
2219 *Core Components* shall include the following attributes:

- 2220 • **Unique Identifier (mandatory):** The identifier that references a *Data Type*  
2221 instance in a unique and unambiguous way.
- 2222 • **Version (mandatory):** An indication of the evolution over time of a *Data*  
2223 *Type* instance.
- 2224 • **Dictionary Entry Name (mandatory):** The official name of a *Data Type*.
- 2225 • **Definition (mandatory):** The semantic meaning of a *Data Type*.
- 2226 • **Usage Rule (optional, repetitive):** A constraint that describes specific  
2227 conditions that are applicable to the *Data Type*.

2228 [S28] Stored *Data Types* shall include the following attribute:

- 2229 • **Qualifier Term (mandatory):** A semantically meaningful name that  
2230 differentiates the *Data Type* from its underlying *Core Component Type*. It  
2231 shall serve as basis for the *Dictionary entry Name* of the *Data Type*.

2232 [S29] Stored *Data Types* shall have a *Core Component Type* as their basis.

2233 [S30] Stored *Data Types* may include one or more *Content Component Restrictions*  
2234 and one or more *Supplementary Component Restrictions* to provide additional  
2235 information on the relationship between the *Data Type* and its corresponding  
2236 *Core Component Type*. They identify restrictions on the format of the *Content*  
2237 *Component* and/or restrictions on the possible values of the *Supplementary*  
2238 *Components* of the corresponding *Core Component Type*.

2239 [Example]

2240 The *Core Component Type Quantity* has a *Supplementary Component Quantity Unit*  
2241 with possible values like 'gram' and 'second'. A *Data Type* that is used for a *Basic*  
2242 *Core Component* such as **Person. Weight. Quantity** will not accept 'second' as  
2243 quantity unit.

2244 **7.2.2 Stored Content Component Restrictions**

2245 [S31] Stored *Content Component Restrictions* shall only be used to define format  
2246 restrictions on the *primitive type* of the *Content Component* of the *Core*  
2247 *Component Type* on which the *Data Type* is based. The list of allowed format  
2248 restrictions per *Primitive Type* is defined in Table 7-1.

2249 **Table 7-1. Primitive Types and their related facets**

Primitive Type	Format Restriction	Definition
String	Pattern	Defines the set of characters that can be used at a particular position in a string.
String	Length	Defines the required length of the string.
String	Minimum Length	Defines the minimum length of the string.  [Note] This format restriction shall not be used in combination with the “Length” format restriction.
String	Maximum Length	Defines the maximum length of the string.  [Note] This format restriction shall not be used in combination with the “Length” format restriction.
String	Enumeration	Defines the exhaustive list of allowed values.
Decimal, Integer	Total Digits	Defines the maximum number of digits to be used.
Decimal	Fractional Digits	Defines the maximum number of fractional digits to be used.
Decimal, Integer	Minimum Inclusive	Defines the lower limit of the range of allowed values. The lower limit is also an allowed value.
Decimal, Integer	Maximum Inclusive	Defines the upper limit of the range of allowed values. The upper limit is also an allowed value.
Decimal, Integer	Minimum Exclusive	Defines the lower limit of the range of allowed values. The lower limit is no allowed value.  [Note] This format restriction shall not be used in combination with the “Minimum Inclusive” format restriction.

Primitive Type	Format Restriction	Definition
Decimal, Integer	Maximum Exclusive	Defines the upper limit of the range of allowed values. The upper limit is no allowed value.  [Note] This format restriction shall not be used in combination with the “Maximum Inclusive” format restriction.
Date	Minimum Inclusive	Defines the lower limit of the range of allowed dates. The lower limit is also an allowed date.
Date	Maximum Inclusive	Defines the upper limit of the range of allowed dates. The upper limit is also an allowed date.
Date	Minimum Exclusive	Defines the lower limit of the range of allowed dates. The lower limit is no allowed date.  [Note] This format restriction shall not be used in combination with the “Minimum Inclusive” format restriction.
Date	Maximum Exclusive	Defines the upper limit of the range of allowed dates. The upper limit is no allowed date.  [Note] This format restriction shall not be used in combination with the “Maximum Inclusive” format restriction.

2250 [S32] Stored *Content Component Restrictions* shall contain the following attributes:

- 2251           • **Restriction Type (mandatory):** Defines the type of format restriction that  
2252           applies to the *Content Component*.
- 2253           • **Restriction Value (mandatory):** The actual value of the format restriction  
2254           that applies to the *Content Component*.

2255 [Note]

2256 The restriction values depend on the restriction type (e.g. integer for a **length**  
2257 restriction type, list of possible values for an **enumeration** restriction type.).



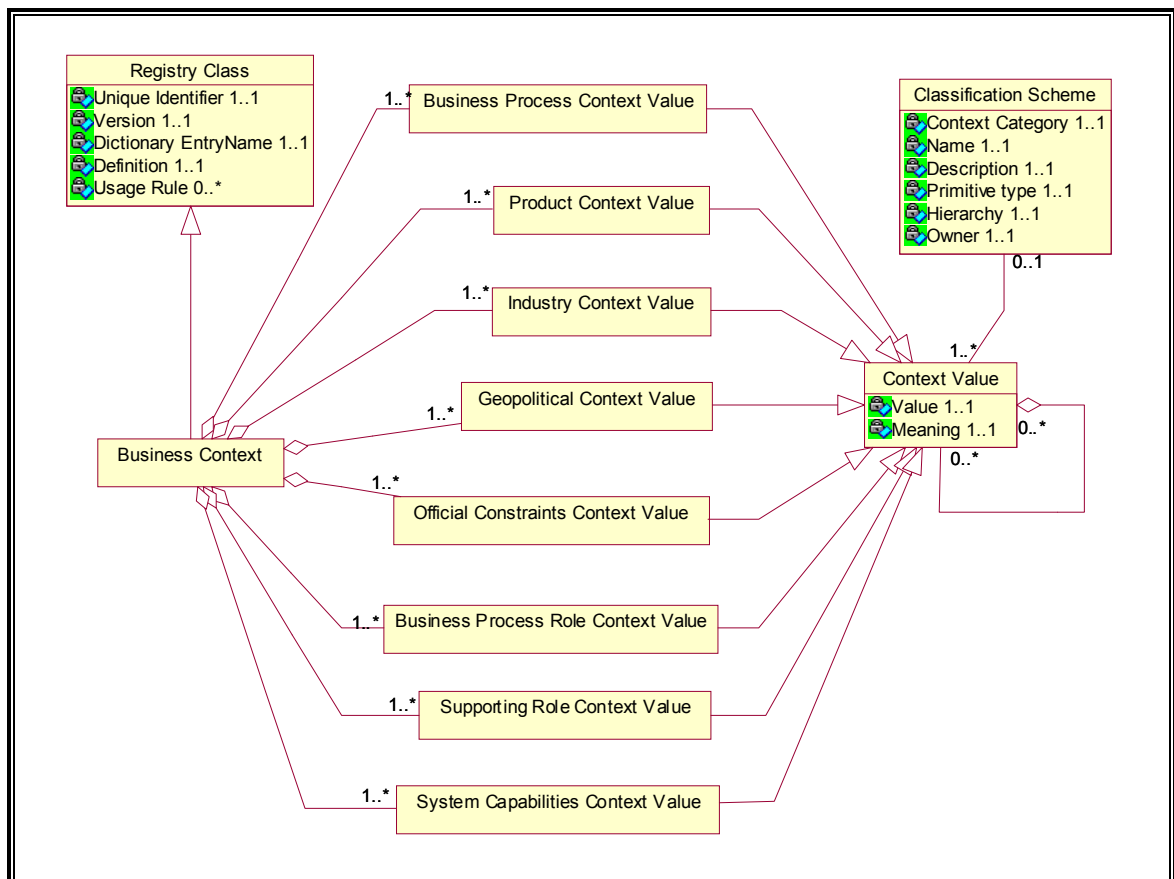
### 2258 7.2.3 Stored Supplementary Component Restrictions

2259 [S33] Stored *Supplementary Component Restrictions* shall only be used to restrict  
 2260 the possible values of the *Supplementary Component* of the *Core Component*  
 2261 *Type* on which the *Data Type* is based.

2262 [S34] Stored *Supplementary Component Restrictions* shall contain the following  
 2263 attributes:

- 2264 • **Supplementary Component Name (mandatory):** Identifies the  
 2265 *Supplementary Component* on which the restriction applies.
- 2266 • **Restriction Value (mandatory, repetitive):** The actual value(s) that is  
 2267 (are) valid for the *Supplementary Component*.

2268 **Figure 7-2 Core Components Context Definition Model**



2269

### 2270 7.3 Stored Context

2271 This section fully describes *Context* storage details. Figure 7-2 is the *Unified*  
 2272 *Modeling Language* model of all aspects of *Context*. It shows that there are a number  
 2273 of *Context Categories* (e.g. Region, Product), which can each be described by one or  
 2274 more *Classification Schemes* (e.g. United Nations scheme for products, World Trade  
 2275 Organization scheme for products). For each *Classification Scheme* the list of possible

2276 values (and their meaning) is defined. A *Business Context* is then defined as a unique  
2277 and meaningful combination of *Context* values.

### 2278 **7.3.1 Stored Business Contexts**

2279 [S35] *Business Contexts* are a particular category of *Registry Classes*. As such, all  
2280 stored *Business Contexts* shall include the following attributes:

- 2281 • **Unique Identifier (mandatory):** The identifier that references a *Business*  
2282 *Context* instance in a unique and unambiguous way.
- 2283 • **Version (mandatory):** An indication of the evolution over time of  
2284 *Business Context* instance.
- 2285 • **Dictionary Entry Name (mandatory):** The official name of a *Business*  
2286 *Context*.
- 2287 • **Definition (mandatory):** The semantic meaning of a *Business Context*.
- 2288 • **Usage Rule (optional, repetitive):** A constraint that describes specific  
2289 conditions that are applicable to the *Business Context*.

2290 [S36] Stored *Business Contexts* shall contain the combination of values for all  
2291 approved *Context Categories* so as to define a unique and meaningful *Business*  
2292 *Context*.

2293 [S37] Stored *Business Context* shall contain the combination of values for all  
2294 approved *Context Categories* so as to define a unique and meaningful *Business*  
2295 *Context*.

### 2296 **7.3.2 Stored Classification Schemes**

2297 [S38] Stored *Classification Schemes* shall include the following attributes:

- 2298 • **Context Category (mandatory):** Name used to identify the approved  
2299 *Context Category* for which the *Classification Scheme* can be used.
- 2300 • **Name (mandatory):** Name under which the *Classification Scheme* is  
2301 known.
- 2302 • **Definition (mandatory):** Definition of the *Classification Scheme*.
- 2303 • **Primitive type (mandatory):** *Primitive type* that is used for the  
2304 representation of a *Context Value* in the *Classification Scheme*.
- 2305 • **Hierarchy (mandatory):** Indicator describing whether the *Classification*  
2306 *Scheme* supports a hierarchical description of the *Context*.

- 2307           • **Owner (mandatory):** Organisation that is responsible for the  
2308           *Classification Scheme*.

### 2309   **7.3.3 Stored Context Values**

2310   [S39] *Stored Context Values* shall describe a possible value of a particular *Context*  
2311       *Category*.

2312   [S40] *Stored Context Values* shall be defined as one of the eight recognized  
2313       types—*Business Process Context Value*, *Product Context Value*, *Industry*  
2314       *Context Value*, *Geopolitical Context Value*, *Official Constraints Context*  
2315       *Value*, *Business Process role Context Value*, *Supporting Role Context Value*  
2316       or *System Capabilities Context Value*.

2317   [S41] *Stored Context Values* may belong to a particular *Classification Scheme*.

2318   [S42] *Stored Context Values* that belong to a particular *Classification Scheme* that  
2319       allows a hierarchy, may have a hierarchical **contains** relation with another  
2320       *Context Value* belonging to the same *Classification Scheme*.

2321   [S43] *Stored Context Value(s)* shall include the following attributes:

- 2322           • **Value (mandatory):** Value describing a particular *Context*.
- 2323           • **Meaning (mandatory):** Description of the meaning of the corresponding  
2324           value.

2325

2326   [Note]

2327   The *Context* value is derived from a *Business Process* model which presumably uses  
2328   values that have their meaning defined somewhere. For example, if the value is taken  
2329   from a code list (specified in the *Classification Scheme*), then the meaning of the code  
2330   should be provided by the code list specification. As an alternative solution, Meaning  
2331   could optionally be a Uniform Resource Identifier that points to the definition.

## 2332   **7.4 Stored Business Information Entities**

2333   This section fully describes *Business Information Entity* storage details. Figure 7-3 is  
2334   the *Unified Modeling Language* model of all aspects of *Business Information Entity*  
2335   and fully describes the types of *Business Information Entities* and their relationships  
2336   as a requirement of storage.



- 2348           • **Version (mandatory):** An indication of the evolution over time of a  
2349           *Business Information Entity* instance.
- 2350           • **Dictionary Entry Name (mandatory):** The official name of a *Business*  
2351           *Information Entity*.
- 2352           • **Definition (mandatory):** The semantic meaning of a *Business Information*  
2353           *Entity*.
- 2354           • **Usage Rule (optional, repetitive):** A constraint that describes specific  
2355           conditions that are applicable to the *Business Information Entity*.
- 2356       [S45] Stored *Business Information Entities* shall be based on a stored *Business*  
2357       *Context*.
- 2358       [S46] Stored *Business Information Entities* shall be based on a stored *Aggregate*  
2359       *Core Component*, *Basic Core Component* or *Association Core Component*.  
2360       They shall never be based on a *Core Component Type*.
- 2361       [S47] Stored *Business Information Entities* shall be defined as one of the three  
2362       recognized types—*Basic Business Information Entity*, *Association Business*  
2363       *Information Entity* or *Aggregate Business Information Entity*. The type of  
2364       *Business Information Entity* shall be the same as the type of its related *Core*  
2365       *Component*:
- 2366           • An *Aggregate Business Information Entity* is based on an *Aggregate Core*  
2367           *Component*.
- 2368           • A *Basic Business Information Entity* is based on a *Basic Core Component*.
- 2369           • An *Association Business Information Entity* is based on an *Association*  
2370           *Core Component*.
- 2371       [S48] Stored *Business Information Entities* shall include the following attributes:
- 2372           • **Constraint Language (optional, repetitive):** a formal description of a way  
2373           the *Business Information Entity* is derived from the corresponding stored  
2374           *Core Component* and stored *Business Context*.
- 2375           • **Business Term (optional, repetitive):** A synonym term under which the  
2376           *Business Information Entity* is commonly known and used in the business.  
2377           A *Business Information Entity* may have several business terms or  
2378           synonyms.
- 2379           • **Example (optional, repetitive):** Example of a possible value of a *Business*  
2380           *Information Entity*

## 2381 7.4.2 Stored Aggregate Business Information Entities

2382 [S49] *Aggregate Business Information Entities* are a particular category of *Business*  
2383 *Information Entities*. As such, stored *Aggregate Business Information Entities*  
2384 shall include all attributes of stored *Business Information Entities*.

2385 [S50] Stored *Aggregate Business Information Entities* shall contain one or more  
2386 *Business Information Entity Properties*.

2387 [S51] Stored *Aggregate Business Information Entities* can be referenced by one or  
2388 more *Association Business Information Entity Properties* of other *Aggregate*  
2389 *Business Information Entities*.

2390 [S52] Stored *Aggregate Business Information Entities* shall include the following  
2391 attribute:

- 2392 • **Qualifier Term (mandatory)**: Qualifies the *Object Class Term* of the  
2393 associated *Aggregate Core Component*.

## 2394 7.4.3 Stored Business Information Entity Properties

2395 [S53] Stored *Business Information Entity Properties* shall be stored as part of the  
2396 stored *Aggregate Business Information Entity* to which they belong, i.e. they  
2397 shall never exist independently of their owning *Aggregate Business*  
2398 *Information Entity*.

2399 [S54] Stored *Business Information Entity Properties* shall be based on a *Core*  
2400 *Component Property* that is stored as part of the *Aggregate Core Component*  
2401 on which the owning *Aggregate Business Information Entity* is based.

2402 [S55] Stored *Business Information Entity Properties* shall be defined as one of the  
2403 two recognized types—*Basic Business Information Entity Property* or  
2404 *Association Business Information Entity Property*. The type of *Business*  
2405 *Information Entity Property* shall be the same as the type of its related *Core*  
2406 *Component Property*:

- 2407 • A *Basic Business Information Entity Property* is based on a *Basic Core*  
2408 *Component Property*.

- 2409 • An *Association Business Information Entity Property* is based on an  
2410 *Association Core Component Property*.

2411 [S56] Stored *Business Information Entity Properties* shall include the following  
2412 attributes:

- 2413 • **Qualifier Term (optional)**: Qualifies the *Property Term* of the associated  
2414 *Core Component Property* in the associated *Aggregate Core Component*.

- 2415           • **Cardinality (mandatory):** Indication whether the *Business Information*  
2416            *Entity Property* represents an optional, mandatory and/or repetitive  
2417            characteristic of the *Aggregate Business Information Entity*.

#### 2418   **7.4.4 Stored Basic Business Information Entity Properties**

2419   [S57] *Basic Business Information Entity Properties* are a particular category of  
2420    *Business Information Entity Properties*. As such, stored *Basic Business*  
2421    *Information Entity Properties* shall include all attributes of stored *Business*  
2422    *Information Entity Properties*.

2423   [S58] Stored *Basic Business Information Entity Properties* shall be linked to the  
2424    *Data Type* that describes the possible values of the *Basic Business Information*  
2425    *Entity Property*. This *Data Type* shall either be the same as the *Data Type* that  
2426    is linked to the corresponding *Basic Core Component Property* or it shall be a  
2427    more restricted *Data Type* (i.e. additional and/or more restrictive *Content*  
2428    *Component Restrictions* and/or additional and/or more restrictive  
2429    *Supplementary Component Restrictions*).

#### 2430   **7.4.5 Stored Association Core Component Properties**

2431   [S59] *Association Business Information Entity Properties* are a particular category of  
2432    *Business Information Entity Properties*. As such, stored *Association Business*  
2433    *Information Entity Properties* shall include all attributes of stored *Business*  
2434    *Information Entity Properties*.

2435   [S60] Stored *Association Business Information Entity Properties* shall be linked to  
2436    the *Aggregate Business Information Entity* that describes the structure. This  
2437    *Aggregate Business Information Entity* shall be based on the *Aggregate Core*  
2438    *Component* that describes the structure of the corresponding *Association Core*  
2439    *Component Property*.

#### 2440   **7.4.6 Stored Basic Business Information Entities**

2441   [S61] *Basic Business Information Entities* are a particular category of *Business*  
2442    *Information Entities*. As such, stored *Basic Business Information Entities* shall  
2443    include all attributes of stored *Business Information Entities*.

2444   [S62] Stored *Basic Business Information Entities* shall represent a *Basic Business*  
2445    *Information Entity Property* of a particular *Aggregate Business Information*  
2446    *Entity*.

#### 2447   **7.4.7 Stored Association Business Information Entities**

2448   [S63] *Association Business Information Entities* are a particular category of *Business*  
2449    *Information Entities*. As such, stored *Association Business Information*  
2450    *Entities* shall include all attributes of stored *Business Information Entities*.

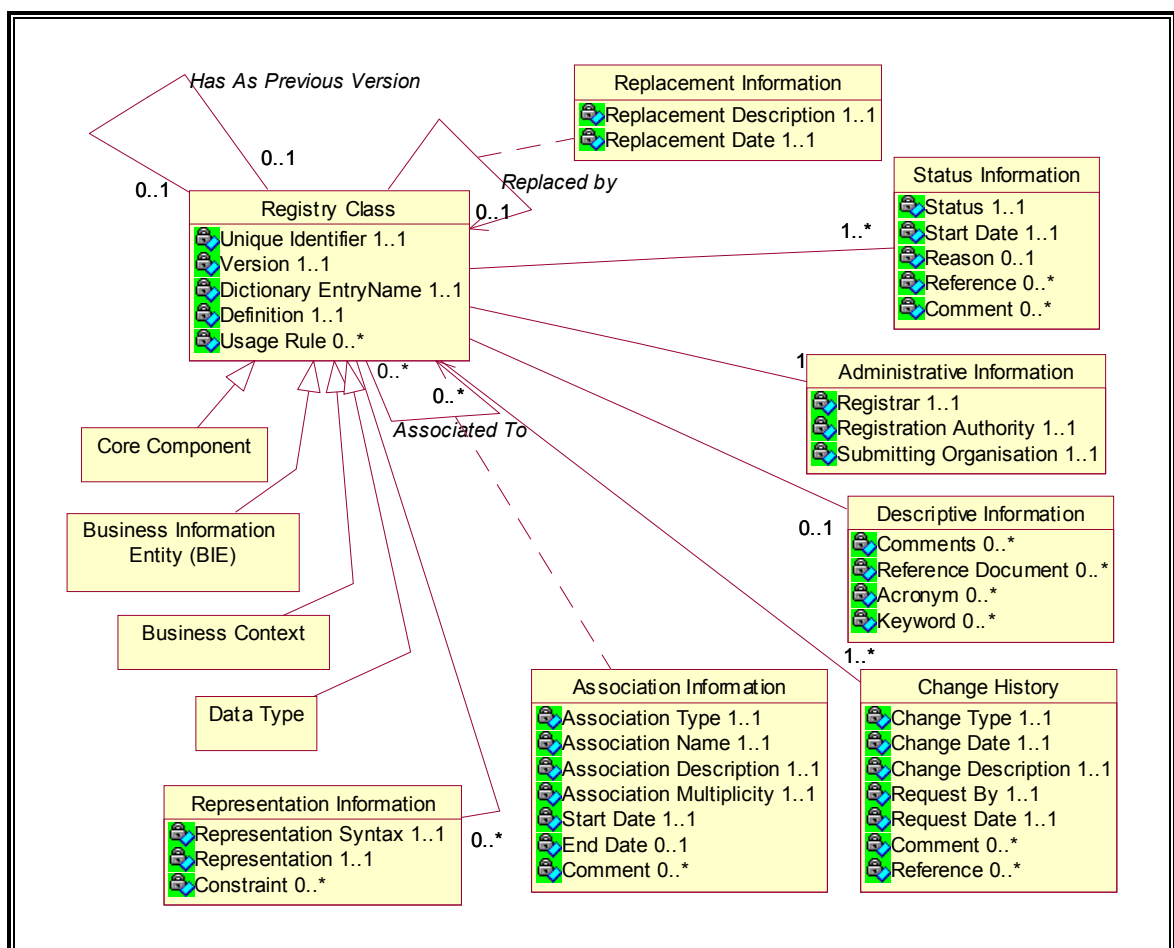
2451 [S64] Stored *Association Business Information Entities* shall represent an  
 2452 *Association Business Information Entity Property* of a particular *Aggregate*  
 2453 *Business Information Entity*.

## 2454 7.5 Core Component Storage Metadata

2455 *Core Components, Data Types, Business Contexts* and *Business Information Entities*  
 2456 are used to design business documents and document components. To facilitate re-  
 2457 usability, it is important that these artefacts be searchable and retrievable.

2458 Figure 7-4 focuses on the meta-information that needs to be defined for *Registry*  
 2459 *Metadata and Registry Classes* (i.e. all information needed to store for *Core*  
 2460 *Components, Data Types, Business Contexts* and *Business Information Entities*). To  
 2461 simplify the diagram all information regarding the structure of a *Core Component* and  
 2462 a *Business Information Entity* has been hidden.

2463 **Figure 7-4. Registry Metadata**



2464

2465 As shown in Figure 7-4, the following metadata categories will be required:

- 2466 • **Version Information:** even though at any given point in time only one  
 2467 version of a *Registry Class* can be valid, multiple previous versions may



- 2468 have existed and a future version may be in preparation. The **Version**  
2469 association makes it possible to link the consecutive versions of a *Registry*  
2470 *Class*. There will not be branches in the versioning; only a linear  
2471 versioning will be supported.
- 2472 • **Replacement Information:** a *Registry Class* may be replaced by another  
2473 *Registry Class* at some point in time (e.g. because a duplicate is  
2474 discovered). The **Replaced by** association makes it possible to do this and  
2475 *Replacement Information* makes it possible to document the date of and  
2476 reason for replacement.
  - 2477 • **Status Information:** information about the live status of a *Registry Class*.
  - 2478 • **Administrative Information:** information about the registration of the  
2479 *Registry Class*.
  - 2480 • **Descriptive Information:** additional descriptive information about a  
2481 *Registry Class*, giving further clarification about its meaning.
  - 2482 • **Change History:** information about all changes that are made to a *Registry*  
2483 *Class*.
  - 2484 • **Association Information:** a *Registry Class* may be associated to multiple  
2485 other *Registry Classes*.
  - 2486 • **Representation Information:** information about the physical  
2487 representation of a *Registry Class* in a particular syntax (e.g. to document  
2488 the XML-tag).

### 2489 **7.5.1 General Metadata Storage Rules**

- 2490 [S65] Stored *Registry Classes* shall include a unique identifier.
- 2491 [S66] Stored *Registry Classes* shall include a version number to keep track of the  
2492 evolution over time of a *Registry Class*.
- 2493 [S67] Stored *Registry Classes* shall include a *Dictionary Entry Name*.
- 2494 [S68] Stored *Registry Classes* shall include a *Definition*.
- 2495 [S69] Stored *Registry Classes* may include one or more *Usage Rules*, describing  
2496 how and/or when to use the *Registry Class*.
- 2497 [S70] Except for the first version of a *Registry Class*, each stored version shall be  
2498 linked to its previous version.
- 2499 [S71] Except for the last version of a *Registry Class*, each stored version shall be  
2500 linked to its next version.

2501 [S72] Stored *Registry Classes* shall include the history of the status lifecycle of each  
2502 version.

## 2503 7.5.2 Management Information

### 2504 7.5.2.1 Administrative Information

2505 [S73] Stored *Registry Classes* shall contain administrative information and shall  
2506 include the following attributes:

- 2507 • **Registrar (mandatory):** Name of the responsible person who has created  
2508 the *Registry Class* in the registry
- 2509 • **Registration Authority (mandatory):** Organisation authorised to register  
2510 the *Registry Class*.
- 2511 • **Submitting Organisation (mandatory):** The organisation that has  
2512 submitted / requested the *Registry Class*.

### 2513 7.5.2.2 Status Information

2514 [S74] Stored *Registry Classes* shall contain status information to include the  
2515 following attributes:

- 2516 • **Status (mandatory):** Status of the *Registry Class* (i.e. draft, provisionally  
2517 registered, registered, to be retired, retired, ...)
- 2518 • **Start Date (mandatory):** Date on which the status comes into effect
- 2519 • **Reason (optional):** Description of why the *Registry Class* status has been  
2520 changed.
- 2521 • **Reference (optional, repetitive):** External Document(s) containing  
2522 relevant information about the status change.
- 2523 • **Comment (optional, repetitive):** Remark about the *Registry Class* status.

### 2524 7.5.2.3 Change History

2525 [S75] Stored *Registry Classes* shall include the history of all modifications related to  
2526 each version to include the following attributes:

- 2527 • **Change Type (mandatory):** Purpose of the Change—such as *new*  
2528 *element, new version, element modification, status modification, element*  
2529 *replacement*.
- 2530 • **Change Date (mandatory):** Date on which the modification has been  
2531 made.

- 2532           • **Change Description (mandatory):** Description of why and how the  
2533           *Registry Class* has been modified.
- 2534           • **Request By (mandatory):** Name of the organisation that has requested the  
2535           modification of the *Registry Class*.
- 2536           • **Request Date (mandatory):** Date on which the modification was  
2537           requested.
- 2538           • **Comment (optional, repetitive):** Remark about the *Registry Class*  
2539           modification.
- 2540           • **Reference (optional, repetitive):** External Document(s) containing  
2541           relevant information about the modification.
- 2542    7.5.2.4   Replacement Information
- 2543    [S76] For each stored pair of *Registry Classes* where one *Registry Class* replaces the  
2544           other, the stored information shall specify *Replacement Information* to include  
2545           the following attributes:
- 2546           • **Replacement Description (mandatory):** Reason for the *Registry Class*  
2547           being replaced
- 2548           • **Replacement Date (mandatory):** Date from which the replacement is  
2549           effective.
- 2550    [S77] If another *Registry Class* has replaced a *Registry Class*, it shall be linked to the  
2551           *Registry Class* by which it has been replaced.
- 2552    [S78] If a *Registry Class* replaces one or more other *Registry Class(es)*, it shall be  
2553           linked to the *Registry Class(es)* it replaces
- 2554    **7.5.3 Content Information**
- 2555    7.5.3.1   Descriptive Information
- 2556    [S79] Stored *Registry Classes* may include additional descriptive information to  
2557           include the following attributes:
- 2558           • **Comments (optional, repetitive):** Comments is additional information  
2559           about a *Registry Class*, which is not part of the *Definition* but that is  
2560           considered relevant for clarification.
- 2561           • **Reference Document (optional, repetitive):** Reference Document is a  
2562           reference (e.g. a Uniform Resource Locator) to external documentation  
2563           that contains relevant additional information about a *Registry Class*.

- 2564           • **Acronym (optional, repetitive):** Acronym is an abbreviation or code  
2565           under which the *Registry Class* is commonly known.
- 2566           • **Keyword (optional, repetitive):** Keyword is one or more significant  
2567           words used for the search and retrieval of a *Registry Class*.
- 2568    7.5.3.2   Representation Information
- 2569    [S80]   Stored *Registry Classes* may optionally include information about the  
2570           representation of the *Registry Class* in one or more syntaxes to include the  
2571           following attributes.
- 2572           • **Representation Syntax (mandatory):** Identification of the representation  
2573           syntax
- 2574           • **Representation (mandatory):** Physical representation of the *Registry*  
2575           *Class* (e.g. Extensible Markup Language tag)
- 2576           • **Constraint (optional, repetitive):** Description of additional constraints  
2577           that apply to the representation of the *Registry Class* in the given syntax  
2578           (e.g. maximum length, ...)
- 2579    7.5.3.3   Association Information
- 2580    [S81]   Stored *Registry Classes* shall include all associations they have with other  
2581           stored *Registry Classes* and shall include the following attributes:
- 2582           • **Association Name (mandatory):** Name of the association
- 2583           • **Association Description (mandatory):** Descriptive text explaining the  
2584           meaning of the association
- 2585           • **Association Type (mandatory):** Type of association (e.g. aggregation,  
2586           specialisation, generalization, simple association ...)
- 2587           • **Association Multiplicity (mandatory):** *Cardinality* of the association (i.e.  
2588           optional/mandatory and repetition)
- 2589           • **Start Date (mandatory):** Date at which the association becomes valid
- 2590           • **End Date (optional):** Date from which the association is no longer valid
- 2591           • **Comment (optional, repetitive):** Relevant information about the  
2592           association (e.g. reason why it has been removed, ...)

2593 **8 Approved Core Component Type, Content, and**  
 2594 **Supplementary Components; and Permissible**  
 2595 **Representation Terms**

2596 The following subsections contain tables that convey the currently approved *Core*  
 2597 *Component Types* (Section 8.1), the approved *Core Component Type Content* and  
 2598 *Supplementary Components* (Section 8.2), and permissible *Representation Terms*  
 2599 (Section 8.3).

2600 **8.1 Approved Core Component Types**

2601 Table 8-1 presents the currently approved set of *Core Component Types*.

2602 *Table 8-1 Approved Core Component Types (CCT)*

CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
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</li> <li>Amount Currency. Identifier</li> <li>Amount Currency. Code List Version. Identifier</li> </ul>
Binary Object. Type	A set of finite-length sequences of binary octets.	Shall also be used for data types representing graphics (i.e., diagram, graph, mathematical curves or similar representations), pictures (i.e. visual representation of a person, object, or scene), sound, video, etc.	Binary Object	Type	<ul style="list-style-type: none"> <li>Binary Object. Content</li> <li>Binary Object. Format. Text</li> <li>Binary Object. Type. Code</li> <li>Binary Object. Encoding. Code</li> <li>Binary Object. Uniform Resource. Identifier</li> </ul>
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.	Should not be used if the character string identifies an instance of an object class or an object in the real world, in which case the Identifier. Type should be used.	Code	Type	<ul style="list-style-type: none"> <li>Code. Content</li> <li>Code List. Identifier</li> <li>Code List. Agency. Identifier</li> <li>Code List. Agency Name. Text</li> <li>Code List. Name. Text</li> <li>Code List. Version. Identifier</li> <li>Code. Name. Text</li> <li>Language. Identifier</li> <li>Code List. Uniform Resource. Identifier</li> <li>Code List Scheme. Uniform Resource. Identifier</li> </ul>
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</li> <li>Date Time. Format. Text</li> </ul>

<b>CCT Dictionary Entry Name</b>	<b>Definition</b>	<b>Remarks</b>	<b>Object Class</b>	<b>Property Term</b>	<b>CCT Components</b>
Identifier. Type	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects in the same scheme together with relevant supplementary information.		Identifier	Type	<ul style="list-style-type: none"> <li>Identifier. Content</li> <li>Identification Scheme. Identifier</li> <li>Identification Scheme. Name. Text</li> <li>Identification Scheme Agency. Identifier</li> <li>Identification Scheme. Agency Name. Text</li> <li>Identification Scheme. Version. Identifier</li> <li>Identification Scheme Data. Uniform Resource. Identifier</li> <li>Identification Scheme. Uniform Resource. Identifier</li> </ul>
Indicator. Type	A list of two mutually exclusive Boolean values that express the only possible states of a Property.		Indicator	Type	<ul style="list-style-type: none"> <li>Indicator. Content</li> <li>Indicator. Format. Text</li> </ul>
Measure. Type	A numeric value determined by measuring an object along with the specified unit of measure.		Measure	Type	<ul style="list-style-type: none"> <li>Measure. Content</li> <li>Measure Unit. Code</li> <li>Measure Unit. Code List Version. Identifier</li> </ul>
Numeric. Type	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.	May or may not be decimal	Numeric	Type	<ul style="list-style-type: none"> <li>Numeric. Content</li> <li>Numeric. Format. Text</li> </ul>
Quantity. Type	A counted number of non-monetary units possibly including fractions.		Quantity	Type	<ul style="list-style-type: none"> <li>Quantity. Content</li> <li>Quantity. Unit. Code</li> <li>Quantity Unit. Code List. Identifier</li> <li>Quantity Unit. Code List Agency. Identifier</li> <li>Quantity Unit. Code List Agency Name. Text</li> </ul>
Text. Type	A character string (i.e. a finite set of characters) generally in the form of words of a language.	Shall also be used for names (i.e. word or phrase that constitutes the distinctive designation of a person, place, thing or concept).	Text	Type	<ul style="list-style-type: none"> <li>Text. Content</li> <li>Language. Identifier</li> <li>Language. Locale. Identifier</li> </ul>

2603

2604 **8.2 Approved Core Component Type Content and**  
 2605 **Supplementary Components**

2606 Table 8-2 presents the currently approved set of *Core Component Type Content* and  
 2607 *Supplementary Components*.

2608 ***Table 8-2. Approved Core Component Type Content and Supplementary***  
 2609 ***Components***

2610

Name	Primitive data-type	Definition	Remarks
Amount. Content	decimal	A number of monetary units specified in a currency where the unit of currency is explicit or implied	
Amount Currency. Code List Version. Identifier	string	The version of the UN/ECE Rec. 9 code list.	
Amount Currency. Identifier	string	The currency of the amount	Reference UN/ECE Rec. 9, using 3-letter alphabetic codes. The UN/ECE Rec. 9 is also published as ISO 4217, but is available in electronic form and free of charge.
Binary Object. Content	binary	A set of finite-length sequences of binary octets.	
Binary Object. Format. Text	mime	The format of the binary content.	
Binary Object. Mime. Type	mime	The mime type of the binary	Reference IETF RFC 2046.
Binary Object. Encoding. Type	string	The binary encoding	Reference IETF RFC 2047
Binary Object. Uniform Resource. Identifier	string	The Uniform Resource Identifier that identifies where the Binary Object is located.	
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	
Code List. Agency. Identifier	string	An agency that maintains one or more code lists	Defaults to the UN/EDIFACT data element 3055 code list.
Code List. Agency Name. Text	string	The name of the agency that maintains the code list.	
Code List. Name. Text	string	The name of a list of codes.	
Code List. Identifier	string	The identification of a list of codes	Can be used to identify the URL of a source that defines the set of currently approved permitted values
Code List Scheme. Uniform Resource. Identifier	string	The Uniform Resource Identifier that identifies where the code list scheme is located.	
Code List. Uniform Resource. Identifier	string	The Uniform Resource Identifier that identifies where the code list is located.	
Code List. Version. Identifier	string	The version of the code list.	Identifies the version of the UN/EDIFACT data element 3055 code list.
Code. Name. Text	string	The textual equivalent of the code content	If no code content exists, the code name can be used on its own
Date Time. Content	string	The particular point in the progression of time	For times use an ISO 8601 compliant format that includes the UTC offset
Date Time. Format. Text	string	The format of the date/time content	Reference ISO 8601 and W3C note on date time

Name	Primitive data-type	Definition	Remarks
Identification Scheme Agency. Identifier	string	The identification of the agency that maintains the identification scheme.	Defaults to the UN/EDIFACT data element 3055 code list.
Identification Scheme Agency. Name. Text	string	The name of the agency that maintains the identification scheme	
Identification Scheme Data. Uniform Resource. Identifier	string	The Uniform Resource Identifier that identifies where the identification scheme data is located	
Identification Scheme. Identifier	string	The identification of the identification scheme.	
Identification Scheme. Name. Text	string	The name of the identification scheme.	
Identification Scheme. Uniform Resource. Identifier	string	The Uniform Resource Identifier that identifies where the identification scheme is located.	
Identification Scheme. Version. Identifier	string	The version of the identification scheme.	Identifies the version of the UN/EDIFACT data element 3055 code list.
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	
Indicator. Content	string	The value of the indicator	For example on, off, true, false
Indicator. Format. Text	String	Whether the indicator is numeric, textual or binary	
Language. Identifier	string	The identifier of the language used in the corresponding text string	Reference ISO 639: 1998
Language. Locale. Identifier	string	The identification of the locale of the language.	
Measure. Content	decimal	The numeric value determined by measuring an object.	For example, 24.387 kilograms (24.387 is the Measure. Content)
Measure Unit. Code	string	The type of unit of measure	Reference UN/ECE Rec. 20 and X12 355.
Measure Unit. Code List Version. Identifier	string	The version of the measure unit code list.	
Numeric. Content	As defined by Numeric. Format. Text	Numeric information that is assigned or is determined by calculation, counting or sequencing.	May be decimal
Numeric. Format. Text	string	Whether the number is an integer, decimal, real number or percentage	
Quantity. Content	decimal	A counted number of non-monetary units possibly including fractions.	For example 7 bales (7 is the Quantity. Content)
Quantity. Unit. Code	string	The unit of the quantity	May use UN/ECE Recommendation #20
Quantity Unit. Code List Agency. Identifier	string	The identification of the agency which maintains the quantity unit code list	
Quantity Unit. Code List. Identifier	string	The quantity unit code list.	Defaults to the UN/EDIFACT data element 3055 code list.
Quantity Unit. Code List Agency Name. Text	string	The name of the agency which maintains the quantity unit code list.	



Name	Primitive data-type	Definition	Remarks
Text. Content	string	A character string (i.e. a finite set of characters) generally in the form of words of a language.	

2611

### 2612 8.3 Permissible Representation Terms

2613 Table 8-3 presents the set of *Permissible Representation Terms*.

2614 **Table 8-3. Permissible Representation Terms**

2615

Primary Representation Term	Definition	Related Core Component Type	Secondary Representation Terms
<b>Amount</b>	A number of monetary units specified in a currency where the unit of currency is explicit or implied.	Amount. Type	
<b>Binary Object</b>	A set of finite-length sequences of binary octets. [Note: This Representation Term shall also be used for Data Types representing graphics (i.e. diagram, graph, mathematical curves, or similar representation), pictures (i.e. visual representation of a person, object, or scene), sound, video, etc.]	Binary Object. Type	Graphic, Picture, Sound, Video
<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 a Property. [Note: The term 'Code' should not be used if the character string identifies an instance of an object class or an object in the real world, in which case the representation term Identifier should be used.]	Code. Type	
<b>Date Time</b>	A particular point in the progression of time (ISO 8601). [Note: This Representation Term shall also be used for Data Types only representing a Date or a Time.]	Date Time. Type	Date, Time

<b>Primary Representation Term</b>	<b>Definition</b>	<b>Related Core Component Type</b>	<b>Secondary Representation Terms</b>
<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.	Identifier. Type	
<b>Indicator</b>	A list of exactly two mutually exclusive Boolean values that express the only possible states of a Property.  [Note: Values typically indicate a condition such as on/off; true/false etc.]	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.  [Note: This Representation Term shall also be used for measured coefficients (e.g. m/s).]	Measure. Type	
<b>Numeric</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.  [Note: This Representation Term shall also be used for Data Types representing Ratios (i.e. rates where the two units are not included or where they are the same), Percentages, etc.)	Numeric. Type	Value, Rate, Percent
<b>Quantity</b>	A counted number of non-monetary units. Quantities need to be specified with a unit of quantity.  [Note: This Representation Term shall also be used for counted coefficients (e.g. flowers/m <sup>2</sup> ).]	Quantity. Type	

<b>Primary Representation Term</b>	<b>Definition</b>	<b>Related Core Component Type</b>	<b>Secondary Representation Terms</b>
<b>Text</b>	A character string (i.e. a finite set of characters) generally in the form of words of a language.  [Note: This Representation Term shall also be used for names (i.e. word or phrase that constitutes the distinctive designation of a person, place, thing or concept).]	Text. Type	Name

2616

## 2617 9 Definition of Terms

2618 **Aggregate Business Information Entity (ABIE)**– A collection of related pieces of  
2619 business information that together convey a distinct business meaning in a specific  
2620 *Business Context*. Expressed in modelling terms, it is the representation of an *Object*  
2621 *Class*, in a specific *Business Context*.

2622 **Aggregate Core Component - (ACC)** – A collection of related pieces of business  
2623 information that together convey a distinct business meaning, independent of any  
2624 specific *Business Context*. Expressed in modelling terms, it is the representation of an  
2625 *Object Class*, independent of any specific *Business Context*..

2626 **Assembly Rules** - *Assembly Rules* group sets of unrefined *Business Information*  
2627 *Entities* into larger structures. *Assembly Rules* are more fully defined and explained in  
2628 the *Assembly Rules* Supplemental Document.

2629 **Association Business Information Entity (ASBIE)** - A *Business Information Entity*  
2630 that represents a complex business characteristic of a specific *Object Class* in a  
2631 specific *Business Context*. It has a unique business semantic definition. An  
2632 *Association Business Information Entity* represents an *Association Business*  
2633 *Information Entity Property* and is therefore associated to an *Aggregate Business*  
2634 *Information Entity*, which describes its structure. An *Association Business*  
2635 *Information Entity* is derived from an *Association Core Component*.

2636 **Association Business Information Entity Property** - A *Business Information Entity*  
2637 *Property* for which the permissible values are expressed as a complex structure,  
2638 represented by an *Aggregate Business Information Entity*.

2639 **Association Core Component (ASCC)** - A *Core Component* which constitutes a  
2640 complex business characteristic of a specific *Aggregate Core Component* that  
2641 represents an *Object Class*. It has a unique business semantic definition. An  
2642 *Association Core Component* represents an *Association Core Component Property*  
2643 and is associated to an *Aggregate Core Component*, which describes its structure.

2644

2645 **Association Core Component Property** – A *Core Component Property* for which the  
2646 permissible values are expressed as a complex structure, represented by an *Aggregate*  
2647 *Core Component*.

2648 **Attribute** – A named value or relationship that exists for some or all instances of some  
2649 entity and is directly associated with that instance.

2650 **Basic Business Information Entity (BBIE)** – A *Business Information Entity* that  
2651 represents a singular business characteristic of a specific *Object Class* in a specific  
2652 *Business Context*. It has a unique business semantic definition. A *Basic Business*

- 2653 *Information Entity* represents a *Basic Business Information Entity Property* and is  
2654 therefore linked to a *Data Type*, which describes its values. A *Basic Business*  
2655 *Information Entity* is derived from a *Basic Core Component*.
- 2656 ***Basic Business Information Entity Property*** – A *Business Information Entity*  
2657 *Property* for which the permissible values are expressed by simple values, represented  
2658 by a *Data Type*.
- 2659 ***Basic Core Component (BCC)*** – A *Core Component* which constitutes a singular  
2660 business characteristic of a specific *Aggregate Core Component* that represents a  
2661 *Object Class*. It has a unique business semantic definition. A *Basic Core Component*  
2662 represents a *Basic Core Component Property* and is therefore of a *Data Type*, which  
2663 defines its set of values. *Basic Core Components* function as the properties of  
2664 *Aggregate Core Components*.
- 2665 ***Basic Core Component (CC) Property*** – A *Core Component Property* for which the  
2666 permissible values are expressed by simple values, represented by a *Data Type*.
- 2667 ***Business Context*** – The formal description of a specific business circumstance as  
2668 identified by the values of a set of *Context Categories*, allowing different business  
2669 circumstances to be uniquely distinguished.
- 2670 ***Business Information Entity (BIE)*** – A piece of business data or a group of pieces of  
2671 business data with a unique business semantic definition. A *Business Information*  
2672 *Entity* can be a *Basic Business Information Entity (BBIE)*, an *Association Business*  
2673 *Information Entity (ASBIE)*, or an *Aggregate Business Information Entity (ABIE)*.
- 2674 ***Business Information Entity (BIE) Property*** – A business characteristic belonging to  
2675 the *Object Class* in its specific *Business Context* that is represented by an *Aggregate*  
2676 *Business Information Entity*.
- 2677 ***Business Libraries*** – A collection of approved process models specific to a line of  
2678 business (e.g., shipping, insurance).
- 2679 ***Business Process*** – The *Business Process* as described using the *UN/CEFACT*  
2680 *Catalogue of Common Business Processes*.
- 2681 ***Business Process Context*** – The *Business Process* name(s) as described using the  
2682 *UN/CEFACT Catalogue of Common Business Processes* as extended by the user.
- 2683 ***Business Process Role Context*** – The actors conducting a particular *Business*  
2684 *Process*, as identified in the *UN/CEFACT Catalogue of Common Business Processes*.
- 2685 ***Business Term*** – This is a synonym under which the *Core Component* or *Business*  
2686 *Information Entity* is commonly known and used in the business. A *Core Component*  
2687 or *Business Information Entity* may have several business terms or synonyms.

- 2688 **Cardinality** – An indication whether a characteristic is optional, mandatory and/or  
2689 repetitive.
- 2690 **Catalogue of Business Information Entities** – This represents the approved set of  
2691 *Business Information Entities* from which to choose when applying the *Core*  
2692 *Component* discovery process
- 2693 **Catalogue of Core Components** – see *Core Component Catalogue*.
- 2694 **CCL** – see *Core Component Library*.
- 2695 **Child Core Component** – A *Core Component* used as part of a larger aggregate  
2696 construct.
- 2697 **Classification Scheme** – This is an officially supported scheme to describe a given  
2698 *Context Category*.
- 2699 **Constraint Language** – A formal expression of actions occurring in specific *Contexts*  
2700 to assemble, structurally refine, and semantically qualify *Core Components*. The  
2701 result of applying the *Constraint Language* to a set of *Core Components* in a specific  
2702 *Context* is a set of *Business Information Entities*.
- 2703 **Content Component** – Defines the *primitive type* used to express the content of a  
2704 *Core Component Type*.
- 2705 **Content Component Restrictions** – The formal definition of a format restriction that  
2706 applies to the possible values of a *Content Component*.
- 2707 **Context** – Defines the circumstances in which a *Business Process* may be used. This  
2708 is specified by a set of *Context Categories* known as *Business Context*.
- 2709 **Context Category** – A group of one or more related values used to express a  
2710 characteristic of a business circumstance.
- 2711 **Context Rules Construct** – The overall expression of a single set of rules used to  
2712 apply *Context* to *Core Components*.
- 2713 **Controlled Vocabulary** – A supplemental vocabulary used to uniquely define  
2714 potentially ambiguous words or business terms. This ensures that every word within  
2715 any of the *Core Component* names and definitions is used consistently,  
2716 unambiguously and accurately.
- 2717 **Core Component (CC)** – A building block for the creation of a semantically correct  
2718 and meaningful information exchange package. It contains only the information  
2719 pieces necessary to describe a specific concept.

- 2720 **Core Component Catalogue** – The temporary collection of all metadata about each  
2721 *Core Component* discovered during the development and initial testing of this Core  
2722 Component Technical Specification, pending the establishment of a permanent  
2723 Registry/repository.
- 2724 **Core Component Dictionary** – An extract from the *Core Component Catalogue* that  
2725 provides a ready reference of the *Core Component* through its *Dictionary Entry*  
2726 *Name*, component parts, and definition.
- 2727 **Core Component Library** – The *Core Component Library* is the part of the  
2728 registry/repository in which *Core Components* shall be stored as *Registry Classes*. The  
2729 *Core Component Library* will contain all the *Core Component Types*, *Basic Core*  
2730 *Components*, *Aggregate Core Components*, *Basic Business Information Entities* and  
2731 *Aggregate Business Information Entities*.
- 2732 **Core Component Property** – A business characteristic belonging to the *Object Class*  
2733 represented by an *Aggregate Core Component*.
- 2734 **Core Component Type (CCT)** – A *Core Component*, which consists of one and only  
2735 one *Content Component*, that carries the actual content plus one or more  
2736 *Supplementary Components* giving an essential extra definition to the *Content*  
2737 *Component*. *Core Component Types* do not have business semantics.
- 2738 **Data Type** – Defines the set of valid values that can be used for a particular *Basic*  
2739 *Core Component Property* or *Basic Business Information Entity Property*. It is  
2740 defined by specifying restrictions on the *Core Component Type* that forms the basis of  
2741 the *Data Type*.
- 2742 **Definition** – This is the unique semantic meaning of a *Core Component*, *Business*  
2743 *Information Entity*, *Business Context* or *Data Type*.
- 2744 **Dictionary Entry Name** – This is the unique official name of a *Core Component*,  
2745 *Business Information Entity*, *Business Context* or *Data Type* in the dictionary.
- 2746 **Information Entity** – A reusable semantic building block for the exchange of  
2747 business-related information.
- 2748 **Geopolitical Context** – Geographic factors that influence business semantics (e.g.,  
2749 the structure of an address).
- 2750 **Industry Classification Context** – Semantic influences related to the industry or  
2751 industries of the trading partners (e.g., product identification schemes used in different  
2752 industries).
- 2753 **Naming Convention** – The set of rules that together comprise how the dictionary  
2754 entry name for *Core Components* (See Section 6.1.4.1.4) and *Business Information*  
2755 *Entities* (See Section 6.1.4.2.4) are constructed.

- 2756 **Object Class** – The logical data grouping (in a logical data model) to which a data  
2757 element belongs (ISO11179). The *Object Class* is the part of a *Core Component*'s  
2758 *Dictionary Entry Name* that represents an activity or object in a specific *Context*.
- 2759 **Object Class Term** – A component of the name of a *Core Component* or *Business*  
2760 *Information Entity* which represents the Object Class to which it belongs.
- 2761 **Official Constraints Context** – Legal and governmental influences on semantics (e.g.  
2762 hazardous materials information required by law when shipping goods).
- 2763 **Order** – In the *Constraint Language*, the *Property* on the *ContextRules Construct* that  
2764 applies a sequence to the application of a set of rules. Two Rule constructs cannot  
2765 have the same value for the *Property Order*.
- 2766 **Primitive Type** – Used for the representation of a value. Possible values are String,  
2767 Decimal, Integer, Boolean, Date and Binary.
- 2768 **Product Classification Context** – Factors influencing semantics that are the result of  
2769 the goods or services being exchanged, handled, or paid for, etc. (e.g. the buying of  
2770 consulting services as opposed to materials)
- 2771 **Property** – A peculiarity common to all members of an *Object Class*.
- 2772 **Property Term** – A semantically meaningful name for the characteristic of the *Object*  
2773 *Class* that is represented by the *Core Component Property*. It shall serve as basis for  
2774 the *Dictionary Entry Name* of the *Basic* and *Association Core Components* that  
2775 represents this *Core Component Property*.
- 2776 **Qualifier Term** – A word or group of words that help define and differentiate an item  
2777 (e.g. a *Business Information Entity* or a *Data Type*) from its associated items (e.g.  
2778 from a *Core Component*, a *Core Component Type*, another *Business Information Entity*  
2779 or another *Data Type*).
- 2780 **Registry Class** – The formal definition of all the information necessary to be recorded  
2781 in the Registry about a *Core Component*, a *Business Information Entity*, a *Data Type*  
2782 or a *Business Context*.
- 2783 **Representation Term** – The type of valid values for a *Basic Core Component* or  
2784 *Business Information Entity*.
- 2785 **Supplementary Component** – Gives additional meaning to the *Content Component* in  
2786 the *Core Component Type*.
- 2787 **Supporting Role Context** – Semantic influences related to non-partner roles (e.g., data  
2788 required by a third-party shipper in an order response going from seller to buyer.)



- 2789 **Supplementary Component Restrictions** – The formal definition of a format  
2790 restriction that applies to the possible values of a *Supplementary Component*.
- 2791 **Syntax Binding** – The process of expressing a *Business Information Entity* in a  
2792 specific syntax.
- 2793 **System Capabilities Context** – This *Context category* exists to capture the limitations  
2794 of systems (e.g. an existing back office can only support an address in a certain form).
- 2795 **Unique Identifier** – The identifier that references a *Registry Class* instance in a  
2796 universally unique and unambiguous way.
- 2797 **Usage Rules** – *Usage Rules* describe how and/or when to use the *Registry Class*.
- 2798 **User Community** – A user community is a group of practitioners, with a publicised  
2799 contact address, who may define *Context* profiles relevant to their area of business.  
2800 Users within the community do not create, define or manage their individual *Context*  
2801 needs but conform to the community’s standard. Such a community should liase  
2802 closely with other communities and with general standards-making bodies to avoid  
2803 overlapping work. A community may be as small as two consenting organisations.
- 2804 **Version** – An indication of the evolution over time of an instance of a *Core*  
2805 *Component, Data Type, Business Context, or Business Information Entity*.
- 2806 **XML schema** – A generic term used to identify the family of grammar based XML  
2807 document structure validation languages to include the more formal W3C XML  
2808 Schema Technical Specification, Document Type Definition, Schematron, Regular  
2809 Language Description for XML (RELAX), and the OASIS RELAX NG.

2810 **10 References**

- 2811 — *ebXML Technical Architecture Specification v1.04*  
2812 *ebXML Business Process Specification Schema v1.01*
- 2813 — *OASIS/ebXML Registry Information Model v2.0*
- 2814 — *OASIS/ebXML Registry Services Specification v2.0*
- 2815 — *ebXML Requirements Specification v1.06*
- 2816 — *OASIS/ebXML Collaboration-Protocol Profile and Agreement Specification*  
2817 *v2.0*
- 2818 — *OASIS/ebXML Message Service Specification v2.0*  
2819 *ebXML Technical Report, Business Process and Business Information Analysis Overview v1.0*  
2820 *ebXML Technical Report, Business Process Analysis Worksheets & Guidelines v1.0*
- 2821 — *ebXML Technical Report, E-Commerce Patterns v1.0*
- 2822 — *ebXML Technical Report, Catalog of Common Business Processes v1.0*  
2823 *ebXML Technical Report, Core Component Overview v1.05*
- 2824 — *ebXML Technical Report, Core Component Discovery and Analysis v1.04*
- 2825 — *ebXML Technical Report, Context and Re-Usability of Core Components v1.04*
- 2826 — *ebXML Technical Report, Guide to the Core Components Dictionary v1.04*
- 2827 — *ebXML Technical Report, Naming Convention for Core Components v1.04*
- 2828 — *ebXML Technical Report, Document Assembly and Context Rules v1.04*
- 2829 — *ebXML Technical Report, Catalogue of Context Categories v1.04*
- 2830 — *ebXML Technical Report, Core Component Dictionary v1.04*
- 2831 — *ebXML Technical Report, Core Component Structure v1.04*
- 2832 — *Information Technology — Metadata registries: Framework for the*  
2833 *Specification and Standardization of Data Elements, International*  
2834 *Standardization Organization, ISO 11179-1*
- 2835 — *Information Technology — Metadata registries: Classification of Concepts for*  
2836 *the Identification of Domains, International Standardization Organization, ISO*  
2837 *11179-2*
- 2838 — *Information Technology — Metadata registries: Registry Metamodel,*  
2839 *International Standardization Organization, ISO 11179-3*
- 2840 — *Information Technology — Metadata registries: Rules and Guidelines for the*  
2841 *Formulation of Data Definitions, International Standardization Organization,*  
2842 *ISO 11179-4*
- 2843 — *Information Technology — Metadata registries: Naming and Identification*  
2844 *Principles for Data Elements, International Standardization Organization, ISO*  
2845 *11179-5*
- 2846 — *Information Technology — Metadata registries: Framework for the*  
2847 *Specification and Standardization of Data Elements, International*  
2848 *Standardization Organization, ISO 11179-6*
- 2849 — *Information Technologies – Open-edi Reference Model, ISO/IEC 14662*

- 2850 — *Information Technologies – Business Agreement Semantic Descriptive*  
2851 *Techniques – Part 1: Operational Aspects of Open-edi for Implementation ,*  
2852 *ISO/IEC 15944-1*
- 2853 — *UN/CEFACT Modelling Methodology, UN/CEFACT TMWG N090*
- 2854 — *Information Technologies – IT Enablement for Widely Used Coded Domains,*  
2855 *ISO/IEC 18022*
- 2856 — *Information Technologies – Identification and Mapping of Various Categories*  
2857 *of Jurisdictional Domains, ISO/IEC 18038*

**2858 11 Disclaimer**

2859 The views and specification expressed in this document are those of the authors and  
2860 are not necessarily those of their employers. The authors and their employers  
2861 specifically disclaim responsibility for any problems arising from correct or incorrect  
2862 implementation or use of this design.

**2863 12 Contact Information**

2864 Team Leader

2865 Name Hartmut Hermes

2866 Company Siemens AG

2867 Street Suedallee 1

2868 City, state, zip/other 85326 Munich

2869 Nation Germany

2870

2871 Phone: +49 89 636 718 580

2872 Email: hartmut.hermes@siemens.com

2873

2874 Editor

2875 Name Mark Crawford

2876 Company Logistics Management Institute

2877 Street 2000 Corporate Ridge

2878 City, state, zip/other McLean, Virginia 22102

2879 Nation USA

2880

2881 Phone: +01 703 917 7177

2882 Email: mcrawford@lmi.org

2883

## 2884 **Copyright Statement**

2885

2886 Copyright © UN/CEFACT 2002. All Rights Reserved.

2887

2888 This document and translations of it may be copied and furnished to others, and  
2889 derivative works that comment on or otherwise explain it or assist in its  
2890 implementation may be prepared, copied, published and distributed, in whole or in  
2891 part, without restriction of any kind, provided that the above copyright notice and this  
2892 paragraph are included on all such copies and derivative works. However, this  
2893 document itself may not be modified in any way, such as by removing the copyright  
2894 notice or references to UN/CEFACT except as required to translate it into languages  
2895 other than English.

2896

2897 The limited permissions granted above are perpetual and will not be revoked by  
2898 UN/CEFACT or its successors or assigns.

2899

2900 This document and the information contained herein is provided on an "AS  
2901 IS" basis and UN/CEFACT DISCLAIMS ALL WARRANTIES, EXPRESS OR  
2902 IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE  
2903 USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS  
2904 OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR  
2905 A PARTICULAR PURPOSE.