



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

Core Components Technical Specification, Part 1

31 October 2001
Version 1.7



14 **1 Status of This Document**

15 This Technical Specification is being developed in accordance with the
16 UN/CEFACT/TRADE/22 Open Development Process. It has been approved by the
17 eBTWG Core Component Project Team for first draft public release for comment as
18 defined in Step 5 of the Open Development Process.

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

21 Distribution of this document is unlimited.

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

23 This version: Core Components Technical Specification, Version 1.7 of 26 October
24 2001

25 Previous version: Core Components Technical specification, Version 1.6 of 12
26 October 2001

27

28 **2 eBTWG CC Project Team Participants**

29 We would like to recognize the following for their significant participation to the
30 development of this document.

31

32 Project Team Leader:	Hartmut Hermes	Siemens
33 Lead Editor:	Mark Crawford	Logistics Management Institute
34 Editing Team	Mike Adcock	APACS
35	James Whittle	e Centre
36	Alan Stitzer	Marsh, Inc.

37

38 Contributors:	Mary Kay Blantz	Iona Technologies
39	Sue Probert	CommerceOne
40	Stig Korsgaard	Danish Bankers Association
41	Andreas Schultz	GDV
42	Arofan Gregory	CommerceOne
43	Marianne Cockle	APACS
44	Frank VanDamme	SWIFT
45	Paula Heilig	Worldspan
46	Lisa Seaburg	CommerceOne
47	Eduardo Gutentag	Sun Microsystems
48	Nigel Wooden	Accord
49	Gunther Stuhec	SAP AG
50	Hisanao Sugamata	ECOM-Japan
51	James Werner	Boeing
52	Alain Dechamps	CEN/ISSS
53	Kerstien Celis	Seagha c.v.
54	Bill Murray	General Motors
55	Tom Warner	Boeing
56	Scott Coulthurst	State Farm
57	Dale McKay	Logicon/Northrop Grumman
58	Richard May	Marsh, Inc
59	Herbert Thomas	AustriaPro
60	Bernd Boesler	DIN
61	Margaret Pemberton	Diskray
62	Joe Zurlo	Logistics Management Institute

63

64

65	3	Table of Contents	
66	1	Status of This Document.....	2
67	2	eBTWG CC Project Team Participants	3
68	3	Table of Contents	4
69	4	Introduction.....	7
70	4.1	Scope and Focus	7
71	4.2	Structure of this Specification.....	8
72	4.2.1	Notation.....	8
73	4.3	Related Documents	9
74	4.4	Executive Summary	10
75	5	Working Process and Methodology.....	15
76	5.1	Overview	15
77	5.1.1	Discovery	15
78	5.1.2	How to use UN/CEFACT Core Components	16
79	5.1.3	Core Components and Semantic Interoperability	16
80	5.2	Core Components Discovery	19
81	5.2.1	Core Component Discovery – Preparation Steps.....	19
82	5.2.2	Core Components Discovery – Search Registry/Repository	21
83	5.2.3	Core Component Discovery – Basic Business Information Entities	23
84	5.3	Submission.....	23
85	5.3.1	Applying the Naming Convention to a New Item	24
86	5.3.2	Submitting New Aggregate Core Components.....	26
87	5.3.3	Preparation Steps for Requesting a New Basic Core Component	27
88	5.3.4	Preparation for Requesting a New ABIE which re-uses an Existing Aggregate Core Component	27
89	5.4	Harmonisation.....	28
90	5.5	Technical Assessment and Approval.....	29
91	5.6	Context in the Discovery Process	30
92	5.6.1	Guidelines for Analyzing BIEs in Context	31
93	5.6.2	Context Categories.....	32
94	6	Technical Details	36
95	6.1	Core Components and Business Information Entities	36
96	6.1.1	Core Components.....	36
97	6.1.2	Business Information Entities	42
98	6.1.3	Naming Convention	43
99	6.1.3.1	Dictionary Information	43
100	6.1.3.2	Rules	44
101	6.1.3.2.1	General Rules.....	44
102	6.1.3.2.2	Rules for Definitions.....	44
103	6.1.3.2.3	Rules for Dictionary Entry Names.....	45
104	6.1.3.2.4	Rules for Business Terms	47
105	6.1.3.3	List of Representation Terms.....	47
106	6.1.4	Catalogue of Core Components	49
107	6.1.5	Catalogue of Business Information Entities	52
108	6.2	Context.....	52
109	6.2.1	Overview of Context Specification.....	52
110	6.2.1.1	Approved Context Categories	53
111	6.2.1.2	Constraint Language	54
112			

113	6.2.1.3	Syntax Binding.....	55
114	6.2.2	Context Categories Specification.....	55
115	6.2.2.1	Business Process Context	55
116	6.2.2.2	Product Classification Context	55
117	6.2.2.3	Industry Classification Context.....	56
118	6.2.2.4	Geopolitical Context.....	57
119	6.2.2.5	Official Constraints Context	58
120	6.2.2.6	Business Process Role Context.....	58
121	6.2.2.7	Supporting Role Context.....	58
122	6.2.2.8	System Capabilities Context.....	59
123	6.2.3	Context Values.....	59
124	6.2.4	Constraints Language.....	59
125	6.2.4.1	Notes about Assembly	65
126	6.2.4.2	Notes about Context Rules.....	65
127	6.2.4.3	Output Constraints	66
128	6.2.4.4	Ordering and Application	66
129	7	Technical Details - Core Component Repository Storage.....	67
130	7.1	Storing Core Components.....	67
131	7.1.1	Stored Core Components	68
132	7.1.2	Stored Basic Core Components	69
133	7.1.3	Stored Core Component Types.....	69
134	7.1.4	Stored Aggregate Core Components	69
135	7.1.5	Stored Content Components and Supplementary Components	70
136	7.1.6	Stored Data Types.....	70
137	7.1.7	Stored Representation Term	70
138	7.1.8	Stored Supplementary Components.....	70
139	7.1.9	Stored Supplementary Component Value.....	70
140	7.1.10	Stored Supplementary Component Values	71
141	7.1.11	Stored Content Components	71
142	7.1.12	Stored Content Component Restrictions.....	71
143	7.2	Stored Context	72
144	7.2.1	Context categories.....	73
145	7.2.2	Context categories Scheme	74
146	7.2.3	Context Value	74
147	7.3	Stored Business Information Entities.....	75
148	7.3.1	Stored Aggregate Business Information Entities.....	77
149	7.3.2	Stored Assembly Component	77
150	7.3.3	Stored Assembly Data Type	77
151	7.3.4	Basic Business Information Entity (BBIE).....	78
152	7.3.5	BBIE Data Type.....	78
153	7.3.6	Business Information Entity	78
154	7.3.7	Data Type.....	78
155	7.3.8	Supplementary Component Value	79
156	7.3.9	Stored Content Component Restriction	79
157	7.4	Core Component Storage Metadata	80
158	7.4.1	General Metadata Storage Rules.....	81
159	7.4.2	Administrative Information	82
160	7.4.3	Association Information.....	82
161	7.4.4	Change History	82
162	7.4.5	Descriptive Information	83

163	7.4.6	Replacement Information.....	83
164	7.4.7	Representation Information	84
165	7.4.8	Status Information.....	84
166	8	Definition of Terms.....	85
167	9	Disclaimer.....	90
168	10	Contact Information	91
169		Copyright Statement	92
170			

171 **4 Introduction**

172 This Core Components technical specification describes and specifies a new approach
173 to the well-understood problem of the lack of interoperability between applications in
174 the e-business arena. Traditionally, standards for the exchange of business data have
175 been focused on static message definitions that have not enabled a sufficient degree of
176 inter-operability or flexibility. A more flexible and inter-operable way of
177 standardising business semantics is required. The UN/CEFACT Core Component
178 solution described in this technical specification presents a methodology for
179 developing a common set of semantic building blocks that represent the general types
180 of business data in use today.

181 The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD,
182 SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in
183 this document, are to be interpreted as described in Internet Engineering Task Force
184 (IETF) Request For Comments (RFC) 2119.¹

185 **4.1 Scope and Focus**

186 This Core Components Technical Specification can be employed wherever business
187 information is being shared or exchanged amongst and between enterprises,
188 governmental agencies, and/or other organisations in an open and world wide
189 environment. The prime users are business people and applications developers of
190 different organisations that require interoperability of business information. This
191 interoperability covers both interactive and batch exchanges of business data between
192 applications through the use of web-services and web browsers as well as traditional
193 Electronic Data Exchange (EDI) systems.

194 This document will form the basis for standards development work of business
195 analysts, business users and information technology specialists supplying the content
196 of and implementing applications that will employ the UN/CEFACT Core Component
197 Library (CCL).

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

¹ 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>

203 **4.2 Structure of this Specification**

204 Due to the diversity of the intended audience, this document has been divided into
205 three main Sections and an Appendix.

206 • Section 5: Working Process and Methodology for Business
207 Users—Discovery, Harmonisation, Assessment and How to Use

208 • Section 6: Technical Details—Core Components and Context

209 • Section 7: Technical Details—Storage and Metadata

210 • Section 8: Full glossary of terms

211 Sections 5, 6 and 7 are complementary, but may also be used independently of each
212 other. Section 5 is informative. A business audience may choose to read through the
213 working process and methodology section (Section 5) and only reference the
214 Technical Details (Sections 6 and 7) as needed. Sections 6 and 7 are normative. A
215 technical audience may choose to focus on the technical details (Sections 6 and 7),
216 referring to the methodology (Section 5) and example (Part 2 a separate document)
217 sections as appropriate, using the glossary (Section 8).

218 In addition, the Core Components Team has prepared the Core Components Technical
219 Specification, Parts 2 and 3. *Part 2—Core Components Primer* details how the
220 contents of Sections 5, 6, and 7 would be used. *Part 3—Catalogue of Discovered*
221 *Core Components* represents work of various organisations working in a joint
222 endeavour to develop a beginning catalogue of core components.

223 **4.2.1 Notation**

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

225 [Ed. Note] - A note from the editing team indicating where additional work is required
226 before the document becomes final. Ed. Notes are informative.

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

228 [Issue] - A recorded issue. Issues are informative.

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

230 [Rn] - Identification of a rule that requires conformance to ensure discovered core
231 components are properly discovered, named and stored. The value R is a prefix to
232 categorize the type of rule where R=B for Business Information rule, R=C for Core
233 Component rule, or R=S for Storage rule; and n (1..n) indicates the sequential number
234 of the rule]. Rules are normative.

235 **4.3 Related Documents**

236 The following documents provided significant levels of influence in the development
237 of this document:

- 238 — ebXML Technical Architecture Specification v1.04ebXML Business Process
239 Specification Schema v1.01
- 240 — ebXML Registry Information Model v1.0
- 241 — ebXML Registry Services Specification v1.0
- 242 — ebXML Requirements Specification v1.06
- 243 — ebXML Collaboration-Protocol Profile and Agreement Specification v1.0
- 244 — ebXML Message Service Specification v1.0ebXML Technical Reports
- 245 — Business Process and Business Information Analysis Overview v1.0Business
246 Process Analysis Worksheets & Guidelines v1.0 -
- 247 — E-Commerce Patterns v1.0
- 248 — Catalog of Common Business Processes v1.0Core Component Overview v1.05
- 249 — Core Component Discovery and Analysis v1.04
- 250 — Context and Re-Usability of Core Components v1.04
- 251 — Guide to the Core Components Dictionary v1.04
- 252 — Naming Convention for Core Components v1.04
- 253 — Document Assembly and Context Rules v1.04
- 254 — Catalogue of Context categories v1.04
- 255 — Core Component Dictionary v1.04
- 256 — Core Component Structure v1.04
- 257 — Information Technology — Metadata registries: Framework for the
258 Specification and Standardization of Data Elements, International
259 Standardization Organization, ISO 11179-1
- 260 — Information Technology — Metadata registries: Classification of Concepts for
261 the Identification of Domains, International Standardization Organization, ISO
262 11179-2
- 263 — Information Technology — Metadata registries: Registry Metamodel,
264 International Standardization Organization, ISO 11179-3
- 265 — Information Technology — Metadata registries:Rules and Guidelines for the
266 Formulation of Data Definitions, International Standardization Organization,
267 ISO 11179-4
- 268 — Information Technology — Metadata registries: Naming and Identification
269 Principles for Data Elements, International Standardization Organization, ISO
270 11179-5
- 271 — Information Technology — Metadata registries: Framework for the
272 Specification and Standardization of Data Elements, International
273 Standardization Organization, ISO 11179-6

274 **4.4 Executive Summary**

275 This Core Component technical specification provides a way to identify, capture and
276 maximize the reuse of business information to support and enhance information
277 interoperability across multiple business situations. The specification focuses both on
278 human-readable and machine-processable representation of this information.

279 The system is more flexible than current standards in this area because the semantic
280 standardisation is done in a syntax-neutral fashion. UN/CEFACT can guarantee that
281 two trading partners using different syntaxes (e.g. XML and EDIFACT) are using
282 business semantics in the same way. This enables clean mapping between disparate
283 message definitions across syntaxes, industry and regional boundaries.

284 UN/CEFACT Business Process and Core Component solutions capture a wealth of
285 information about the business reasons for variation in message semantics and
286 structure. In the past, such variations have introduced incompatibilities. The core
287 components mechanism uses this rich information to allow identification of exact
288 similarities and differences between semantic models. Incompatibility becomes
289 incremental rather than wholesale, i.e. the detailed points of difference are noted,
290 rather than a whole model being dismissed as incompatible.

291 The key concepts in the Core Components Technical Specification are:

- 292 • Core Component — The Core Component is a semantic building block
293 that is used as a basis to construct all electronic business messages. A
294 technical specification for creating a core component library is provided.

[Definition] Core Component (CC)

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

- 295 • Context – Context is a mechanism for classifying business situations. Once
296 business contexts are identified, the appropriate core components can be
297 selected or created and differentiated to indicate any necessary
298 qualification and refinement needed to support the business process in a
299 given context.

- 300 • Business Information Entity –When a Core Component is used in a real
301 business situation it is used to define a Business Information Entity. The
302 BIE is the result of using a core construct within a specific business
303 context.

[Definition] Business Information Entity (BIE)

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

- 304 • Repository Metadata – Core Components, Context Categories and
305 Business Information Entities along with syntax bound business message
306 descriptions are available in the repository. The relationships between
307 these objects are stored to encourage standard use and re-use at all levels.

308 There are three different categories of Core Components: Basic Core Component,
309 Core Component Type and Aggregate Core Component. The following definitions
310 explain each of these:

[Definition] Basic Core Component (BCC)

A Core Component that represents a singular business concept with a unique business semantic definition. A BCC is constructed by using a Core Component Type. BCCs are used in developing Aggregate Core Components.

[Definition] Core Component Type (CCT)

This is a Core Component that has no business meaning on its own. For example, date on its own has no business meaning, whereas the date of birth, the contact date, the delivery date do express business meaning.

Each Core Component Type contains one Content Component that carries the actual content. It will also contain Supplementary Component(s) that provide essential definition to the content.

[Example] Core Component Types

If the content component carries “12” this has no meaning on its own. But “12 Kilometers” or “12 Euro”, where ‘Kilometers’ or ‘Euro’ are supplementary components that give essential extra definition, do have meaning.

311

[Definition] **Aggregate Core Component**

A collection of pieces of business information that together form a single business concept (e.g. postal address). Each Aggregate Core Component has its own unique business semantic definition and can contain either:

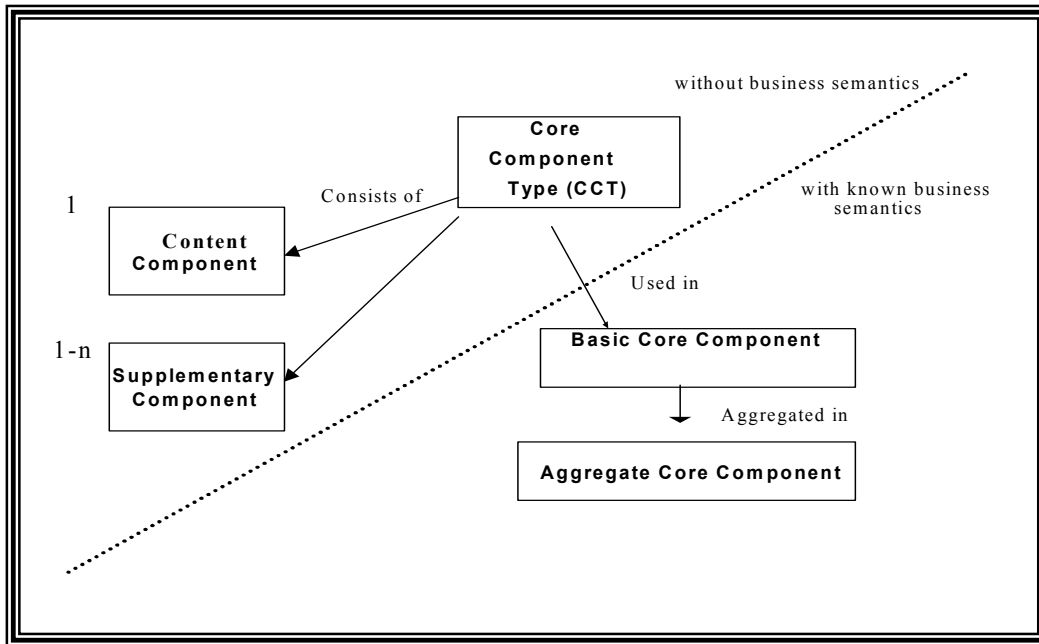
- two or more Basic Core Components, or
- at least one Basic Core Component plus one or more Aggregate Core Components

[Example] - Aggregate Core Components

account details, party details

312 Figure 4-1 shows the relationships between these three categories.

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



314 A specific relationship exists between Core Components and Business Information
 315 Entities. Core and Business elements are complementary in many respects. Core
 316 components are intended to be the linchpin for creating business process models and
 317 business documents using a fixed vocabulary.

[Definition] **Basic Business Information Entity**

A core component used in a specific business context. A Basic Business Information Entity is derived from a Basic Core Component.

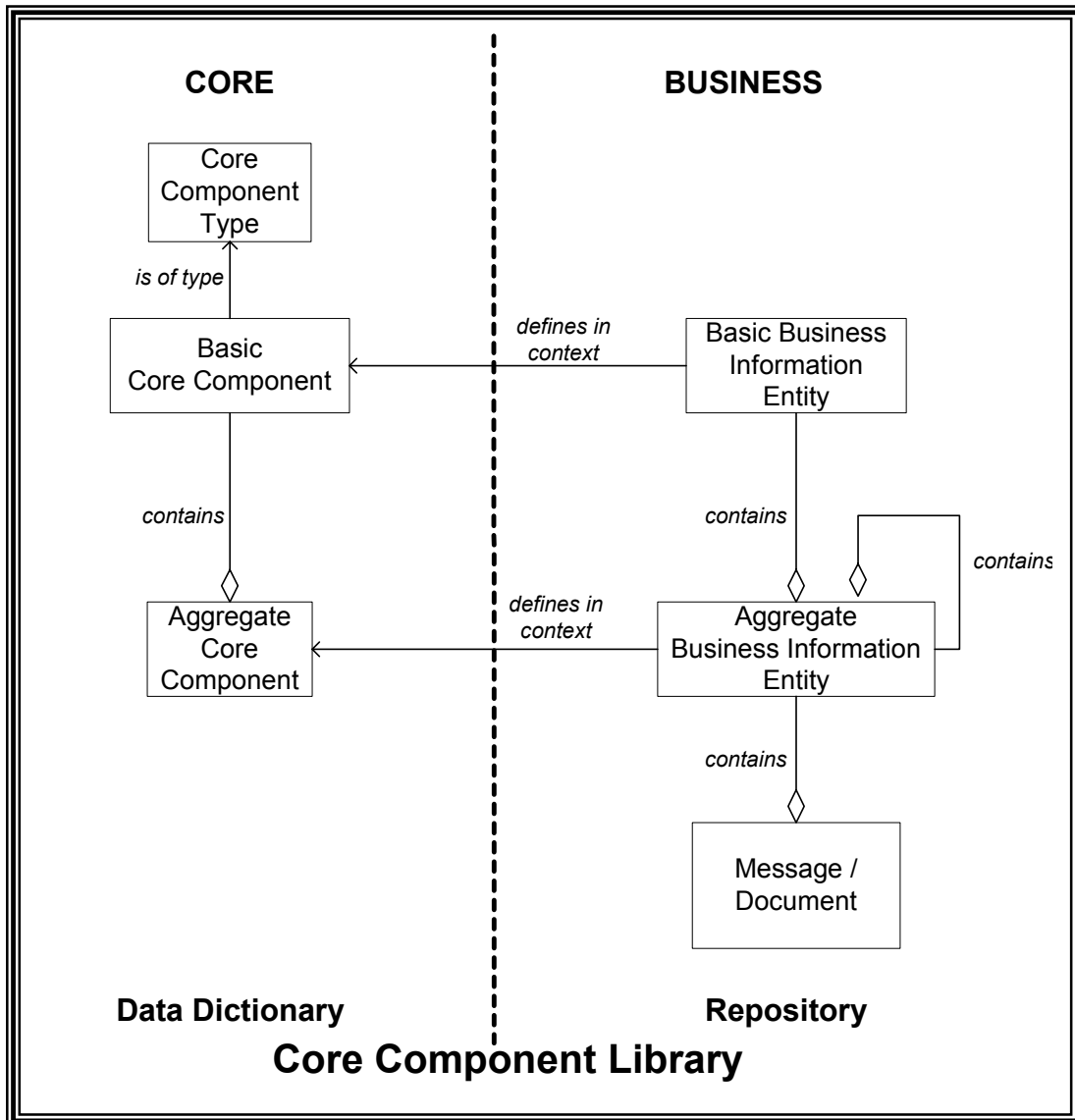
[Definition] Aggregate Business Information Entity

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

318
 319 The features of the relationship between core components and business information
 320 entities are described in Figure 4-2.
 321

322 **Figure 4-2. Relationships between Core Components and Business Information**
 323 **Entities**

324
 325



[Note]

From previous words it should be clear that the term Core Component (CC) is used as a generic term that encompasses Basic Core Components (BCCs) and Aggregate Core Components (ACCs). Equally the term Business Information Entity (BIE) is used as a generic term encompassing Basic Business Information Entities (BBIEs) and Aggregate Business Information Entities (ABIEs).

327 **5 Working Process and Methodology**

328 This chapter identifies aspects of core components working processes and
329 methodologies for use, to include an overview of the discovery and use characteristics
330 of core components as well as detailed recommendations for conducting discovery,
331 storage, approval, and application of context.

332 **5.1 Overview**

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

337 **5.1.1 Discovery**

338 A business process is modeled using the Unified Modeling Methodology (UMM).²
339 One of the results is a class diagram that shows the business information and its inter-
340 relationships. Business Information Entities (BIEs) can be identified from the class
341 diagram.

342 For example, if a domain team has modeled the publication of catalogue data to
343 trading partners, the result will be a BIE representing the distributed catalogue data
344 which is made up of a set of smaller BIEs that are its component parts. Thus, the
345 description of an item is identified as a BIE for this business process.

346 Ultimately, BIEs must be based on a basic library of clearly defined semantic
347 constructs to guarantee that they will inter-operate. This library must include a set of
348 globally agreed semantic definitions such as what will be contained in the
349 UN/CEFACT Core Components library.

350 A BIE is a CC used in a specific business context and given its own unique name. As
351 Basic Core Components (BCCs) are single pieces of business information, when they
352 are used directly in specific business contexts, they do not change.

353

² The UN/CEFACT Modeling Methodology (UMM) is a methodology for business process and information modeling that is based on the Unified Modeling Language.

[Example]

An invoicing business process may need to send/receive an explicit piece of information such as Pre-tax Sub-total Amount. This is a Basic Business Information Entity (BBIE) that uses the generic Amount, which is a Basic Core Component (BCC). It uses the generic Amount in a specific business context and adds a specialised definition, but in all other respects it is the same as Amount, i.e. it has the same structure and data type.

354

355 Just as each BIE must ultimately be based on BCC's, each Aggregate Business
356 Information Entity (ABIE) must ultimately be based on an existing Aggregate Core
357 Component (ACC). The underlying ACC identifies the generic, standard definition of
358 business information that is being used in the ABIE. The ABIE inherits the generic
359 description, which is then modified and enhanced to be specific to the business
360 process in which the ABIE is used. An ABIE is thus directly tied to a specific
361 business process, or "business context." (See Section 5.7 for a fuller understanding of
362 context.)

363 Interoperability of BIEs is therefore guaranteed by the fact that they each inherit a
364 core component structure and associated semantic definitions derived from the core
365 component library.

366 The following section describes the procedure by which the original ebXML Core
367 Component Library was identified and how the next generation UN/CEFACT ebXML
368 compliant library will be developed and maintained.

369 **5.1.2 How to use UN/CEFACT Core Components**

370 This section provides a procedure for the technical user who wants to understand how
371 to implement core components. It assumes the user is dealing with an established set
372 of core components, context categories and metadata/storage. The established set of
373 core components being used should be based on those discovered, harmonised, and
374 published by recognised standards groups. It is further assumed that the recognised
375 standards group(s), and other business association group(s), have also made available
376 sets of BIEs for use in a published set of business processes.

377 **5.1.3 Core Components and Semantic Interoperability**

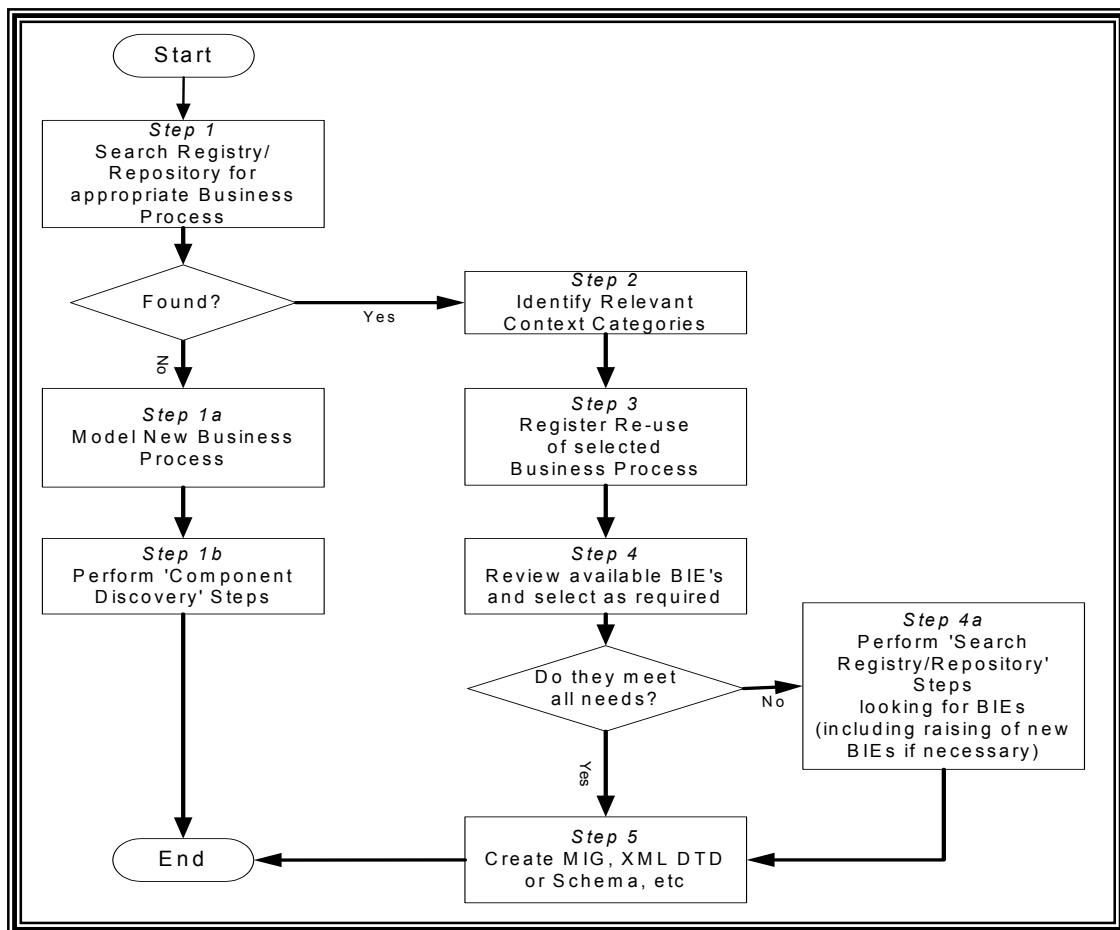
378 Today, the e-business community generally agrees on the definition of a standard
379 message structure expressed as an UN/EDIFACT Message Implementation Guide
380 (MIG), an XML Schema, or similar syntax specific representation. UN/CEFACT will
381 produce standards based representations of these artefacts for implementation.

382 Under the core components concept, defining and storing Core Components and
383 associated context mechanisms occur prior to the creation of a MIG or a Schema. In
384 this manner, the focus of the user changes from examining the MIG or DTD, and

385 moves to an examination of the semantic models. Accordingly, interoperability
 386 between syntaxes is no longer dependent on analysing the various syntactic
 387 instantiations, but naturally occurs during the business process model definition
 388 phase.

389 The overall discovery and document creation process can be thought of as a series of
 390 steps that starts with determining the availability of existing business process
 391 definitions and ultimately results in standard business documents. Figure 5-1
 392 illustrates this process. Specific steps to be followed are further described below.

393 **Figure 5-1. Steps from BP Discovery to CC Discovery**



394

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

398 • If no existing business process is found to be appropriate, then the new
 399 business process should be modelled and submitted to the registry. The
 400 process includes conducting a thorough analysis of the business
 401 information requirements by following the Core Component Discovery
 402 Steps (Section 5.2.2).

- 403 • If an existing business process is located that will be used, the new use
404 should be identified to the registry. If the searcher does not have access
405 to the registry, the catalogue of common Business Processes (CCBP) can
406 be substituted. The searcher continues with Step 2.
- 407 Step 2: Identify relevant context categories—Access the registry interface and
408 identify the relevant context categories of the selected business process by
409 determining the following:
- 410 • Product Classification Context – Determine the goods or services
411 concerned in the collaboration.
- 412 • Industry Classification Context – Determine the relevant trading partner
413 industries.
- 414 • Geopolitical Context– Determine where the business process is to be
415 conducted. Determine if the business process crosses international
416 boundaries.
- 417 • Official Constraints Context – Determine any legal restrictions or
418 requirements on this business process.
- 419 • Business Process Role Context – Identifies the role played by the user and
420 their trading partners. Can be derived from the business process.
- 421 • Supporting Role Context– Determine what other significant parties will be
422 using the data in the messages. Determine their role in the overall
423 process.
- 424 • System Capabilities Context – Determine any major restrictions derived
425 from legacy systems. Identify the type of system.
- 426 The registry will provide a list of pre-defined BIEs that are available to the
427 selected business process, and which meet the context criteria specified.
428 These will come with links to the Core Components that they are based on
429 and the constraint rules that fully qualify them. The Registry should also
430 return partial matches with an indication of how closely they match the
431 specified context.
- 432 Step 3: Register re-use of selected Business Process in the context(s) in which it is
433 being used.
- 434 Step 4: Review the available BIEs and select the appropriate subset for use that
435 meets the needs of the business process requirement that is being developed.
- 436 If the BIEs available for the specific business process do not address
437 all of the data requirements, the repository of all BIEs should be
438 searched to see if the appropriate BIE(s) already exist. The procedure
439 for this is described under Search Repository (Section 5.2) which
440 includes the steps to raise any new BIE(s) required because no
441 appropriate BIE(s) can be found.

442 Step 5: Create MIG, XML DTD or Schema, etc. – The resulting semantic model
443 (the set of BIEs) is manually or programmatically rendered into a syntax-
444 specific message description. The resulting MIG, DTD or Schema is
445 submitted to the repository where it is associated with the BIEs it represents.

[Note]

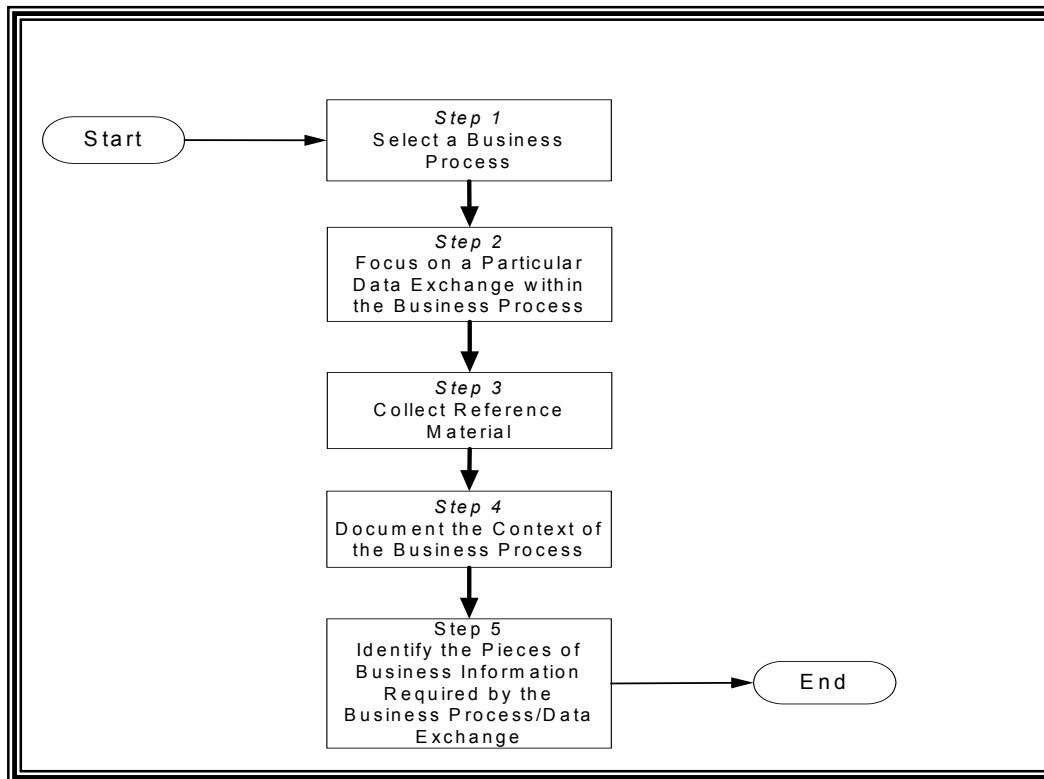
When selecting a business process and defining the required messages, searches may be made against potential trading partners' data requirements and processes. The context rules and BIEs represent useful metadata in determining the best possible match between the user and their partners. The fact that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

446 **5.2 Core Components Discovery**

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

452 **5.2.1 Core Component Discovery – Preparation Steps**

453 These steps identify pieces of business information such as BIEs and ABIEs. An
454 analysis of BIEs from a variety of similar business processes leads to the underlying
455 core structures and semantics of the Core Components. Figure 5-2 graphically
456 portrays the prescribed preparation steps that are described below.

457 **Figure 5-2 Preparation Steps**

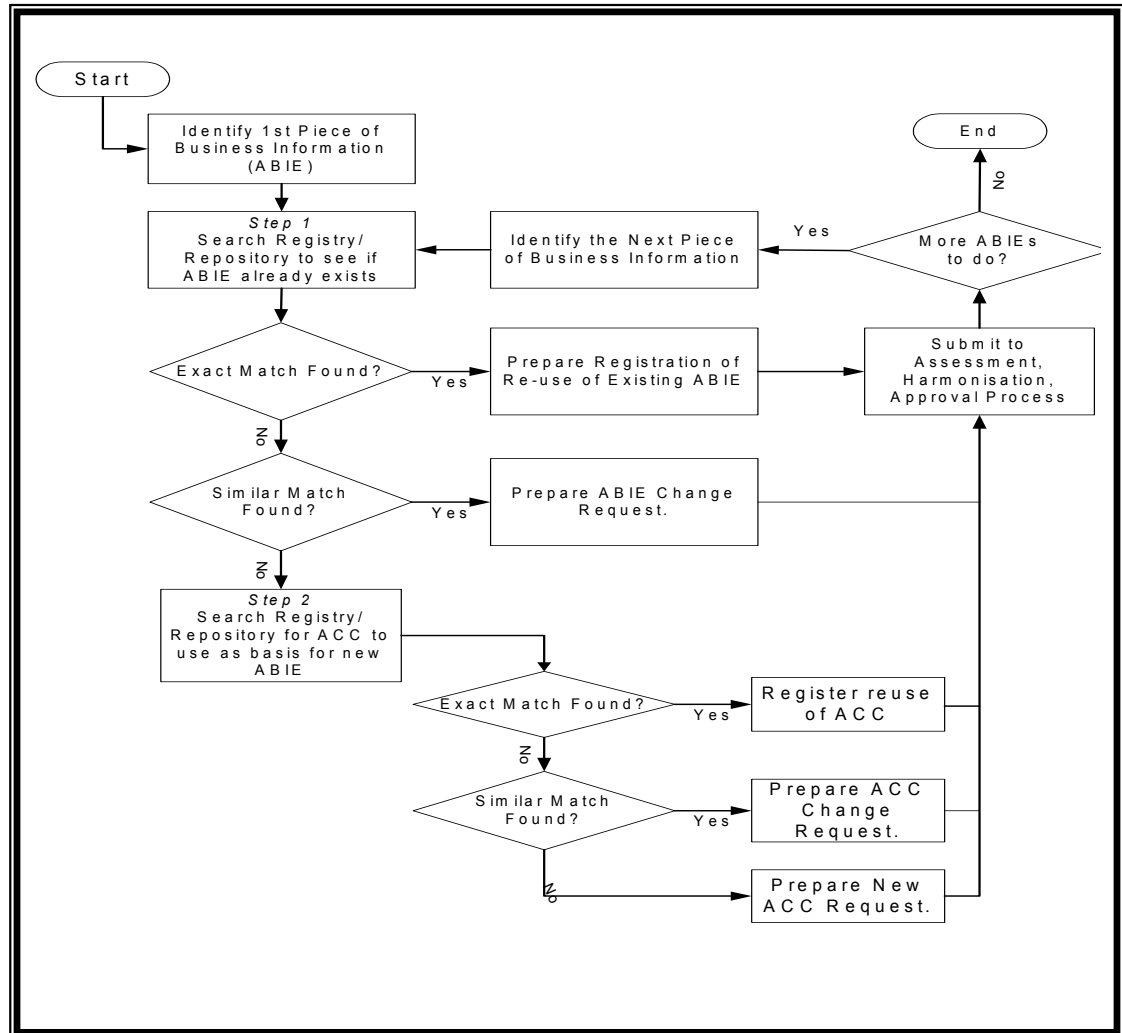
- 458 Step 1. Select the Business Process that provides the widest range of business
 459 information content within the domain being addressed. (e.g. Make a
 460 Payment, Place an Order, Issue an Invoice)
- 461 Step 2. Focus on a particular data exchange within the Business Process that
 462 contains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

- 477 • Industry Classification Context
- 478 • Geopolitical Context
- 479 • Official Constraints Context
- 480 • Business Process Role Context
- 481 • Supporting Role Context
- 482 • System Capabilities Context
- 483 Step 5. Compile a list of the pieces of information required for the business process.
- 484 • If starting from a model (UN/CEFACT recommends UMM models of
- 485 business processes), identify the objects (ABIEs) that are needed.
- 486 • If not starting from a model, collect the pieces of information into object-
- 487 like groups (ABIEs). It is important to recognise and avoid pieces of
- 488 information that are purely used for legacy system or syntax purposes.
- 489 • For each ABIE, capture its semantic definition and any Business Terms by
- 490 which it is commonly known.

491 **5.2.2 Core Components Discovery – Search Registry/Repository**

492 Having discovered a number of ABIEs in the preparation Step 5 identified in Section
493 5.2.1 above, repeat the following steps for each ABIE as shown in Figure 5-3.

494

495 *Figure 5-3 Search Steps*

496 Step 1 Starting with ABIEs at the highest level of aggregation, search the Catalogue
497 of ABIEs for an existing ABIE that has the same definition.

498 • If there is an ABIE with a definition that meets the business need, register
499 the re-use including business context and any business terms. (Go to next
500 ABIE)

501 • If there is an ABIE with a definition that potentially could be modified to
502 meet the business need, prepare an ABIE change request for submission to
503 the harmonisation and approval process. Include re-use, business context
504 and any business terms. (Go to next ABIE)

505 • If there is not an ABIE with a suitable definition, go to Step 2.

506 Step 2 Search the Catalogue of Core Components for an existing ACC that has the
507 appropriate generic definition and structure.

508 • If there is an existing ACC with a definition and structure that meets the
509 business needs, register the re-use of the ACC as an ABIE including the
510 business context and any business terms. (Go to next ABIE)

511 • If there is an ACC with a definition and structure that potentially could be
512 modified to meet the business need, prepare an ACC change request for
513 submission to the harmonisation and approval process. Include the re-use
514 of the ACC as an ABIE, the business context and any business terms. (Go
515 to next ABIE)

516 • If there is not an ACC with a suitable definition and structure, prepare a
517 new ACC request for submission to the harmonisation and approval
518 process. Include the re-use of the ACC as an ABIE, the business context
519 and any business terms. (Go to next ABIE)

520 **5.2.3 Core Component Discovery – Basic Business Information Entities**

521 This procedure is exactly the same as that described in Section 5.2.2, except that the
522 reader should read BBIE for ABIE and BCC for ACC.

523 **5.3 Submission**

524 Following the search of the Core Component Library, there may be a need to prepare
525 submissions for the harmonisation and approval process. The different types of
526 submissions that may be required are detailed below.

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

529 • To register a Re-use of an Existing ABIE

530 • To make a Change Request for an Existing ABIE

531 • To make a Change Request for an Existing Aggregate Core Component

532 The following submissions require more significant preparation, as part of the CC
533 working methodology, to be carried out by the Business Team making the discovery
534 and analysis.

535 • Preparation for Requesting a new Aggregate Core Component

536 • Preparation for Requesting a new Core Component

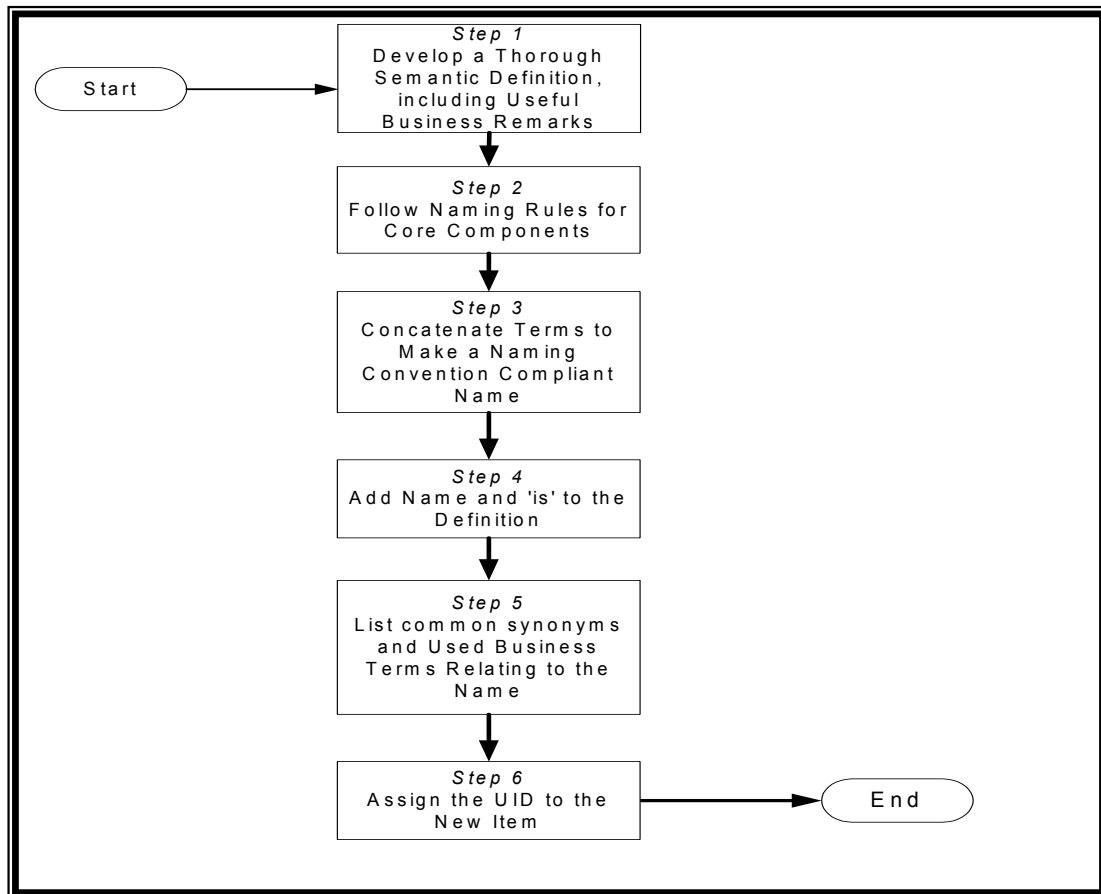
537 • Preparation for Requesting a new ABIE which re-uses an Existing
538 Aggregate Core Component

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

541 **5.3.1 Applying the Naming Convention to a New Item**

542 For all new items, the Naming Convention and associated rules that are defined in
543 Section 6.1.3 must be exercised. The following diagram shows the steps that must be
544 taken, each of which is described in the accompanying text.
545

546 *Figure 5-4 Applying the Naming Convention*



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

- 549
- globally applicable,
 - 550 • generic (i.e. able to cover the same business concept for different
551 products/services),
 - 552 • applicable across multiple industries or domains, and
 - 553 • simple and clear to enable unambiguous translation to other languages

554 Step 2. Follow the Naming Rules for Core Components (Section 6.1.3) to assign:

- 555 • Representation Type
- 556 • Property Term
- 557 • Object Class

558 Step 3. Concatenate the terms to create a Naming Convention compliant name.

[Note]

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

559 Step 4. Check the quality of the definition by adding the words “[*Dictionary Name*]
560 is” to the front of the definition, where [*Dictionary Name*] is the agreed
561 name.

562 Step 5. List common synonyms or business term(s) that are used within the domain
563 to identify the piece of business information (e.g. Account Number, Account
564 Identifier).

[Note]

Some business terms are used for several different pieces of business information. It is perfectly acceptable to have the same business term listed as a synonym for two or more pieces of business information. For example, as shown in Figure 5-5, Account Number is a synonym for Financial Account Identifier and for Sales Account Identifier.

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

Temp UID	Definition	Remarks	Business Terms	CCT	Dictionary Entry Name			Representation Term
					Name	Object Class	Property Term *to be suppressed according to rule 5	
T00010	A Financial Account is a service through a bank or other organisation through which funds are held on behalf of a client or goods or services are supplied on credit	Not a general ledger.	Account	n/a	Financial Account. Details	Financial Account	Details	
T00012	A Sales Account is a relationship between a vendor and a customer.	Usually includes a contract specifying the terms of business.	Account	n/a	Sales Account. Details	Sales Account	Details	

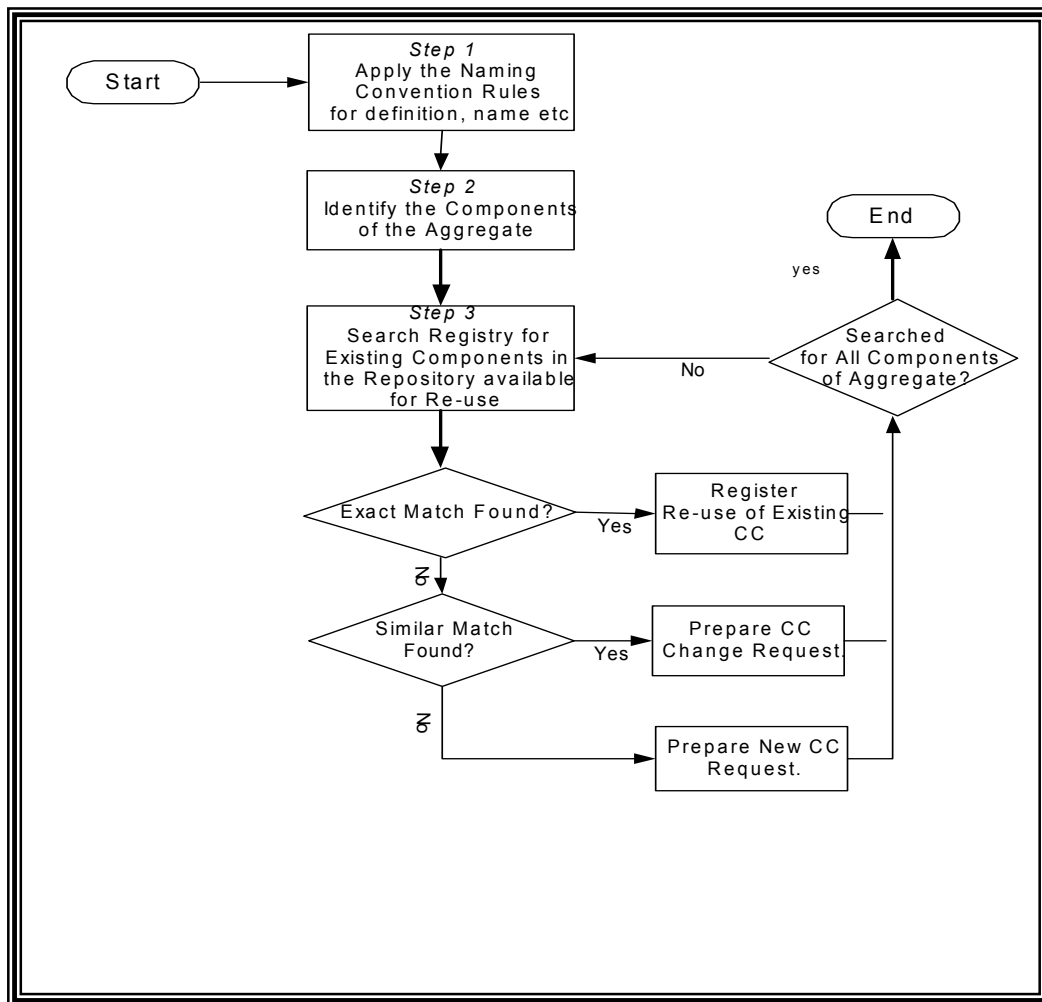
Same Business Term

566 Step 6. Assign a temporary UID to the new item in the form of a 6 digit
567 alphanumeric string.

568 5.3.2 Submitting New Aggregate Core Components

569 The development of a new aggregate requires adherence to the naming convention
570 rules for naming and definition. Once named, the new aggregate's constituent parts
571 need to be individually examined. The following diagram and text describes the
572 procedure that is to be followed.

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



574 Step 1. Apply the Naming Convention and Rules to arrive at the name of the new
575 Aggregate Core Component

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

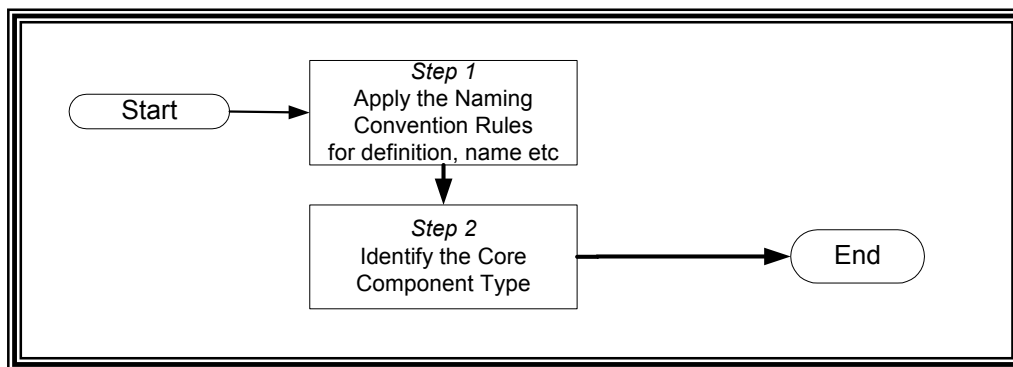
577 Repeat the following step for each constituent component identified in step 2:

- 578 Step 3. Search the Registry for an existing CC that has the appropriate generic
579 definition and structure.
- 580 • If there is an existing CC with a definition and structure that meets the
581 requirement, register this re-use of the CC including the context in which it
582 is used.
 - 583 • If there is an existing CC with a definition and structure that potentially
584 could be modified to meet the requirement, prepare an CC change request
585 for submission to the harmonisation and approval process, including the
586 re-use of the CC and the context in which it is used.
 - 587 • If there is not an existing CC with a suitable definition and structure,
588 prepare a new CC request for submission to the harmonisation and
589 approval process, including the re-use of the CC and the context.

590 5.3.3 Preparation Steps for Requesting a New Basic Core Component

591 As shown in Figure 5-7, there are two steps necessary to prepare for requesting a new
592 Basic Core Component. These two steps are:

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



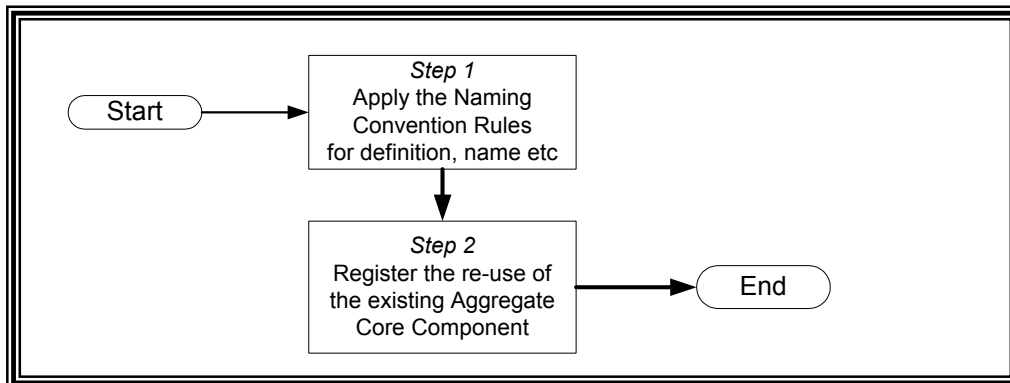
594 Step 1. Apply the Naming Convention and Rules to arrive at the name of the new
595 Aggregate Core Component

596 Step 2. Select the appropriate Core Component Type (CCT). (See Section 6.1.1 for
597 an explanation and listing of CCTs).

598 5.3.4 Preparation for Requesting a New ABIE which re-uses an Existing 599 Aggregate Core Component

600 As shown in Figure 5-8, there are two steps necessary to prepare for requesting a new
601 ABIE that re-uses an existing Aggregate Core Component. These two steps are:

602

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

603 Step 1. Apply the Naming Convention and Rules to arrive at the name of the new
604 Aggregate Business Information Entity.

605 Step 2. Register the re-use of the existing Aggregate Core Component by this new
606 Aggregate Business Information Entity

607 5.4 Harmonisation

608 The purpose of harmonisation is to take a set of proposed Core Components or
609 Business Information Entities from different domains, identify differences and
610 similarities between the various submissions, and produce a single, complete cross-
611 domain set of core components. Harmonisation is a critical step in the overall core
612 component procedures. The following describes a set of recommended harmonisation
613 procedures.

614 Step 1 Evaluate each submitted core component for consistent application of the
615 Discovery methodology. Resolve any questions or issues by discussion with
616 the submitting groups.

[Note]

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

617 Step 2 Compare the definition and structure of each submitted core component with
618 what already exists in the core component library.

- 619 • If the submitted core component is the same or similar, compare the
620 properties of each to identify any differences. If the submitted component
621 has properties missing in the existing one, enforce a harmonised form that

- 622 contains the properties of each. If the submitted component is a subset of
623 the existing core component definition, then recommend the use of the
624 existing one.
- 625 • If the definition of the core component does not match any existing ones,
626 then go to Step 3.
- 627 Step 3 Publish the results of harmonisation to the submitting groups for review and
628 finalisation.
- 629 Once the submitted material has passed the harmonisation procedure, it may now be
630 submitted for assessment and approval.

631 **5.5 Technical Assessment and Approval**

- 632 Technical assessment must be done in close coordination with the discovery teams
633 and the harmonisation process. The following defines a recommended process for
634 conducting technical assessment and approval of all newly submitted and changed
635 core components.
- 636 Technical assessment procedures define the processing steps that shall be followed by
637 the joint development groups, the Harmonisation group, submission entry points, the
638 Technical Assessment group, and the secretariat as related to the review of core
639 components. The result of this process is the final publication of approved core
640 components.
- 641 These procedures were developed in order to facilitate the process of reviewing and
642 approving submissions to the core component library. In order to minimise the
643 requirements for technical assessment and harmonisation, and to expedite the review
644 and approval process, core component development groups should work with the
645 Technical Assessment group, and the Harmonisation group during the early
646 development stages of component discovery.
- 647 Step 1 All CC work that is ready to be reviewed needs to be submitted through the
648 appropriate submission entry point for pre-assessment and forwarding the
649 approved CC submission to the secretariat.
- 650 Step 2 The secretariat will then enter the submission into the CC database. The
651 secretariat will electronically send the CC submission to the Harmonisation
652 Group for its review.
- 653 Step 3 The Harmonisation Group will conduct its review following its own
654 procedures. Once the Harmonisation Group has completed its review, it will
655 return harmonised components to the secretariat sufficiently prior to a
656 Technical Assessment meeting
- 657 Step 4 The Technical Assessment group conducts its final review and approval.

658 Step 5 Once approved by the Technical Assessment group, the approved Core
659 Component(s) will be submitted for entry into the appropriate CC Registry
660 using procedures described elsewhere.

661 At every step in the process, the secretariat should advise all entry points of
662 submissions received and actions taken.

663 For additional information on the draft UN/CEFACT Technical Assessment process,
664 see the current *UN/CEFACT Core Component Technical Review and Approval*
665 *Procedures* document.

666 **5.6 Context in the Discovery Process**

667 The information that exists inside a defined business process is already in a business
668 context. Therefore the initial analysis will be performed on a set of BIEs (both BBIEs
669 and ABIEs) and not on a set of core components (See Figure 5-1). The analysis that
670 produces core components is — among other things — a process of identifying the
671 various context categories and values, to determine those properties that exist in all
672 possible contexts.

673 The guidelines presented here facilitate the analysis of BIEs to determine core
674 business semantics, or provide a mechanism to describe BIEs when they are published
675 in a repository.

676 When doing analysis, there is a key question: “Is a particular property of a BIE
677 derived from its contextual business use, or is it a core property of the component?”

678 The answer to this question can be found by looking at as many different instances of
679 that BIE as possible. If there is a single semantic property of that BIE that is found in
680 every example available for analysis, then it can be assumed that the property in
681 question is in fact a core semantic, and is not derived from the contextual business
682 use.

683 If there are any instances of the BIE in which the property in question is not present,
684 then this raises the issue of identity: Is the BIE which lacks that property really the
685 same BIE, just used in a different context?

686 If the answer to this question is “yes,” then that property is not part of the core
687 component, but is derived contextually, and the property should be removed from the
688 BCC or ACC being discovered.

689 If the answer is “no”, then it is possible that a second, different core component has
690 been discovered.

[Issue]

There is a question regarding properties. A counter position held by some members of the team is that the process that is described by two different BIEs will have all properties. They are just not used in all contexts (or they are specialisations). Under this alternate position, all properties would be stored and BIEs would only be derived by restrictions. This issue will be resolved based on input during the comment period.

691 **5.6.1 Guidelines for Analyzing BIEs in Context**

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

- 694 • *Business Process Context*: This is the classification of the business process
695 as described in the Catalogue of Common Business processes. It is the
696 primary context category, and provides many useful distinctions in the
697 analysis of core components.
- 698 • *Product Classification Context*: There are many types of information that
699 are specific to products or services being traded or referred to in a business
700 process.
- 701 • *Industry Classification Context*: Traditionally, business vocabularies are
702 divided up into industry verticals. This context category specifies a
703 particular industry vertical.
- 704 • *Geopolitical Context*: Specifies the Semantic and structural variation. This
705 is often the result of regional or cultural factors.
- 706 • *Official Constraints Context*: Specifies the legal or contractual influences
707 upon business semantics.
- 708 • *Business Process Role Context*: Every partner in a business process data
709 exchange has a particular role – buyer, seller, etc. These roles are
710 described in the Catalogue of Common Business Processes. Depending on
711 the business process, the nature of these roles may require that certain
712 semantics and data be employed in the messages exchanged. In any
713 Business Process Role context, one must either be a sender or receiver of
714 data in that particular exchange – otherwise, role is described by the
715 Supporting Role Context.
- 716 • *Supporting Role Context*: Parties in a business process who are neither
717 senders nor receivers of data in a particular exchange, may place
718 requirements on the data exchanged by partners who are sending or
719 receiving of data in that exchange. These non-sending, non-receiving
720 parties in this exchange play a supporting role, and are described by the
721 Supporting Role Context.

[Issue]

The decision to split the underlying concept behind Business Process and Supporting Role Context into two separate context drivers needs to be verified with the current Business Process Team approach.

- 722
- 723
- 724
- *System Capabilities Context*: When a particular semantic or structure is primarily the result of system constraints, or compliance with a standard, then it is attributable to the System Capabilities Context.

725 5.6.2 Context Categories

726 Using the criteria given in section 5.6.1 for determining that a particular property of a
727 BIE is in fact the product of its use in context, the analyst must ascertain and
728 document the applicable context categories. To accomplish this, the analyst should list
729 all the context categories, and assign a value or values to each category for that
730 component. If a context category has no particular value or values, then the Analyst
731 should assign a value of “In All Contexts” (for all contexts except Official
732 Constraints) or “None (for Official Constraints). As this analysis is conducted,
733 different context categories might appear to be in competition for application. The
734 analyst must ascertain which context category is responsible.” This section provides
735 some guidelines for answering this question in a systematic and consistent fashion, by
736 examining the typical ambiguities that arise.

737 It is possible that a particular property of a BIE may be the result of several context
738 factors. These context factors are identified by analysis of differences and similarities
739 across particular contexts. For example, comparing the same BIE as used in different
740 regions of the world, variation will probably be the result of a geopolitical context or
741 official constraints context (see below). If a single BIE differs between business
742 processes, then the business process context is probably the cause. For each non-core
743 property of every BIE analysed the relevant influences and hence context factors
744 should be identified.

745 The following guidelines apply:

746 1) Geopolitical Context versus Official Constraints Context

747 If a property can be traced to a specific body of law or international treaty then it is
748 the result of an official constraint. For example, if a warning about hazardous
749 goods is required as part of a goods description, and it is required on all uses of that
750 goods description within the United States, then both Geopolitical and Official
751 Constraints are involved. The value of an Official Constraint Context should
752 always be the body of law or treaty that is being cited. The value of a Geopolitical
753 Context always expresses the region or regions that are relevant.

754 2) Product Classification Context versus Industry Classification Context

755 When a particular variation on a given product or service is specific to a particular
756 industry, then the Industry Classification Context is adequate to specify the
757 context. If all examples of the particular product or service are described by the
758 same unique set of properties across industries, then only a Product Classification
759 Context is required. In other cases, a value or values should be supplied for both
760 context categories.

761 3) Business Process Context versus Business Role Context

762 Business Process Role-based Context is employed when one actor in the business
763 process has an information requirement and the other does not. If both actors have
764 the same information requirement, then it is a Business Process Context.

765 4) System Capability Context categories

766 This context is the result of system or classes of systems that *primarily* influence
767 data variation.

[Example]

If a specific Enterprise Resource Planning (ERP) provider's proprietary data formats use a particular field, and no other applications use that field, then the presence of the data can be attributed to the processing capabilities of that specific system. One way of classifying systems is through their compliance with a particular standard.

768 The following detailed example illustrates the process of assigning values for all
769 context categories as part of the BIE analysis process:

[Example]

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

The following set of values could be ascribed to this child field for this BIE:

- Business Process = “In All Contexts”
- Product Classification = “In All Contexts”
- Industry Classification = “In All Contexts”
- Geopolitical = “United States”
- Official Constraint = “None”
- Business Process Role = “In All Contexts”
- Supporting Role = “In All Contexts”
- System Capabilities = “In All Contexts”

These values were selected based on the following analysis:

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

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

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

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

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

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

770 If, in the above example, the address was taken from a French standard, it might be
771 that some child elements are common across a number of countries in the same

772 region, and perhaps even in multiple regions. Providing the value “France” as a
773 Geopolitical Context here would be incorrect – every known valid value should be
774 given.

775 **6 Technical Details**

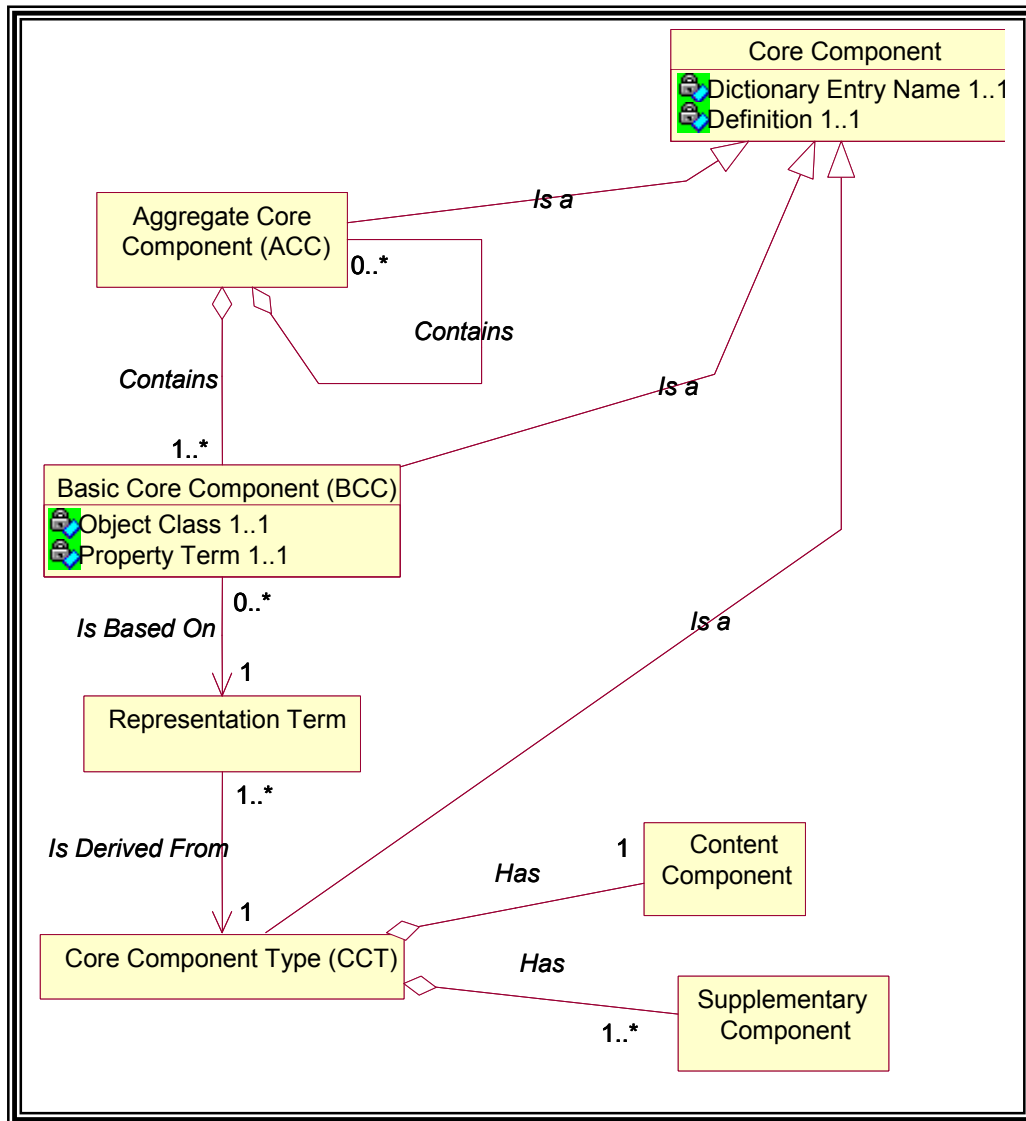
776 This section provides a detailed technical explanation of the Core Component,
777 Business Process integration, storage and metamodel elements of the UN/CEFACT
778 core component concept.

779 **6.1 Core Components and Business Information Entities**

780 This section defines Core Component rules and naming conventions, allowable core
781 component types, content and supplementary component types and representation
782 types. This section also specifies relationships for Core Components and Business
783 Information Entities (BIEs). This section also includes details required for
784 constructing the Core Components Catalogue and a larger Core Component Library.

785 **6.1.1 Core Components**

786 A Core Component is a building block for the creation of a semantically correct and
787 meaningful information exchange 'parcel'. It contains only the information pieces
788 necessary to describe a specific concept. There are three categories of Core
789 Components: Basic Core Component, Core Component Type and Aggregate Core
790 Component. Figure 6-1 illustrates these three categories and their relationships.

791 **Figure 6-1. Core Components Basic Definitions**

792 The following general rules must be followed in discovering and documenting the
 793 three types of core components:

794 [C1] Each Core Component Type, Basic or Aggregate Information Entity must
 795 have its own business semantic definition. The definition shall be developed
 796 first and the Dictionary Entry Name shall be extracted from it. Remarks can be
 797 used to further clarify the definition, to provide examples and/or to reference a
 798 recognised standard.

799 [C2] Within an aggregate, all embedded entities shall be related to the concept of
 800 the aggregate.

801 [C3] There shall be no semantic overlap between the Core Components embedded
802 within the same aggregate.

803 [C4] The representation of the information in a Core Component of the Core
804 Component Type "Code" should use a standard issued by a recognised
805 standards body, whenever a standard exists. If international standards are not
806 used a business driven justification shall be provided.

807 [C5] An aggregate information entity must contain at least one Basic Core
808 Component.

[Issue]

The issue of allowing aggregates without at least one Basic Core Component was discussed in detail and the majority opinion is that problems with infinite loops and miscast aggregates would arise without the restriction to have at least one Basic Core Component in each aggregate. There is a minority opinion that we may discover there are valid technical reasons where occurrences of Aggregate Core Components may only be comprised of other aggregates. A final position on this issue will be developed based on the comments received during the public review period.

809 [C6] For the purpose of exchanging information a practical compromise on the
810 level of detail of a Basic Core Component is required. This compromise shall
811 be based on the Business Need. There is no need to always have absolute
812 detail, which decomposes a piece of information down to its lowest level.

813 [C7] Used Core Component Types will be taken from the approved list of Core
814 Component Types

815 Table 6-1 provides a complete list of the approved Core Component Types.

816 [Ed. Note: The UIDs in Table 6-1 are interim in nature and will be finalized prior to
817 release of this document in specification status.]

818 **Table 6-1 Core Component Types (CCT)**

819

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000105	Amount. Type	A number of monetary units specified in a currency where the unit of currency is explicit or implied.		Amount	Type	<ul style="list-style-type: none"> Amount. Content (000106) Amount Currency. Identification. Code (000107)

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000089	Code. Type	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute together with relevant supplementary information.		Code	Type	<ul style="list-style-type: none"> Code. Content (000091) Code List. Identifier (000092) Code List. Agency. Identifier (000093) Code List. Version. Identifier (000099) Code. Name (000100) Language. Code (000075)
000066	Date Time. Type	A particular point in the progression of time together with relevant supplementary information.	Can be used for a date and/or time.	Date Time	Type	<ul style="list-style-type: none"> Date Time. Content (000067) Date Time. Format. Text (000068)
000200	Graphic. Type	A diagram, graph, mathematical curves, or similar representation.		Graphic	Type	<ul style="list-style-type: none"> Graphic. Content Graphic. Format. Text
000101	Identifier. Type	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme together with relevant supplementary information.		Identifier	Type	<ul style="list-style-type: none"> Identifier. Content (000102) Identification Scheme. Name (000103) Identification Scheme Agency. Name (000104) Language. Code (000075)
000180	Indicator. Type	A list of two, and only two, values which indicate a condition such as on/off; true/false etc. (synonym: "Boolean").		Indicator	Type	<ul style="list-style-type: none"> Indicator. Content (000181) Indicator. Format. Text
000152	Measure. Type	The size, volume, mass, amount or scope derived by performing a physical measure together with relevant supplementary information.		Measure	Type	<ul style="list-style-type: none"> Measure. Content (000153) Measure Unit. Code (000154)
000182	Numeric. Type	A representation of a number.	May or may not be decimal	Numeric	Type	<ul style="list-style-type: none"> Numeric. Content (000183) Numeric. Format. Text
000201	Picture. Type	A visual representation of a person, object, or scene.		Picture	Type	<ul style="list-style-type: none"> Picture. Content Picture. Format. Text

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000108	Quantity. Type	A number of non-monetary units together with relevant supplementary information.		Quantity	Type	<ul style="list-style-type: none"> Quantity. Content (000109) Quantity. Unit. Code (000110) Quantity Unit. Code List. Identifier (000111) Quantity Unit. Code List Agency. Identifier (000112)
000090	Text. Type	A character string with or without a specified language.		Text	Type	<ul style="list-style-type: none"> Text. Content (000094) Language. Code (000075)

820 Table 6-2 presents the definitive set of Core Component Type Content and
821 Supplementary Components. The asterisk (*) in the property term column indicates
822 cases where the property term is the same as either the representation term or object
823 class, and is consequently dropped from the dictionary entry name.

824 [C8] Used Content and Supplementary Component Definitions will be taken from
825 the approved list of Content and Supplementary Component Definitions

826 **Table 6-2. CCT Content and Supplementary Components**

827

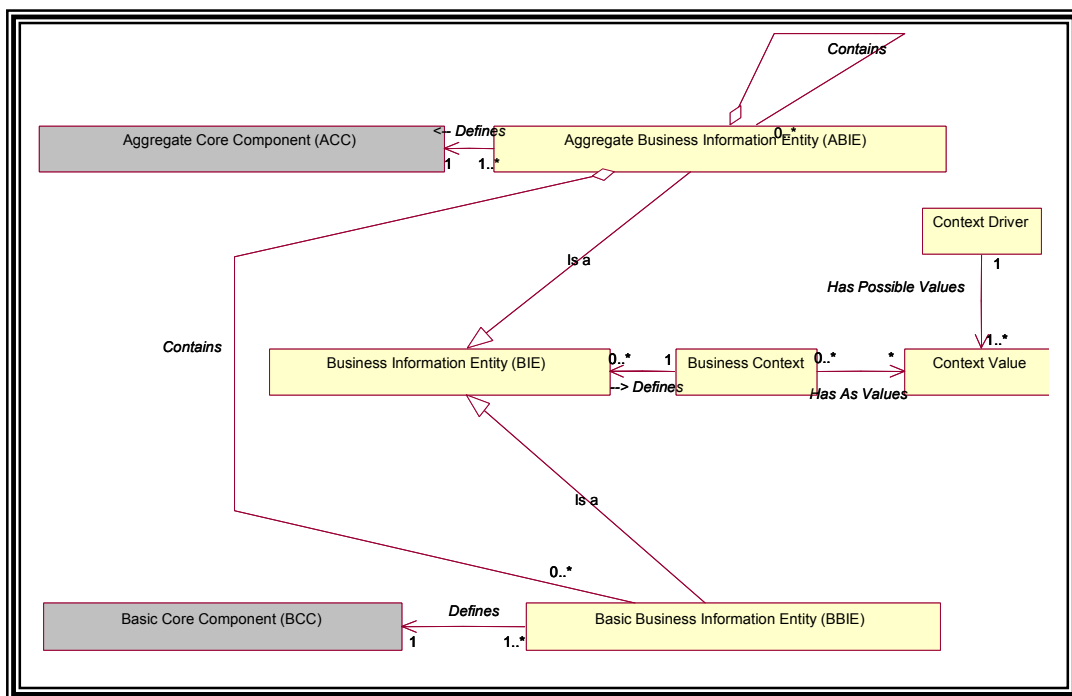
UID	Name	Data-type	Definition	Remarks
000106	Amount. Content	decimal	A number of monetary units specified in a currency where the unit of currency is explicit or implied	
000107	Amount Currency. Identification. Code	string	The currency of the amount	Reference ISO 4217.
000091	Code. Content	string	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute	
000093	Code List. Agency. Identifier	string	An agency that maintains one or more code lists	
000092	Code List. Identifier	string	The name of a list of codes	Can be used to identify the URL of a source that defines the set of currently approved permitted values
000099	Code List. Version. Identifier	string	The version of the code list	
000100	Code. Name	string	The textual equivalent of the code content	If no code content exists, the code name can be used on its own
000067	Date time. Content	string	The particular point in the progression of time	
000068	Date Time. Format. Text	string	The format of the date/time content	Reference ISO 8601
000202	Graphic. Content	binary	A diagram, graph, mathematical curves, or similar representation	
000203	Graphic. Format. Text	string	The format of the graphic content	

UID	Name	Data-type	Definition	Remarks
000104	Identification Scheme Agency. Name	string	The agency that maintains the identification scheme	
000103	Identification Scheme. Name	string	The name of the identification scheme	
000102	Identifier. Content	string	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme	
000181	Indicator. Content	string	The value of the indicator	For example on, off, true, false
	Indicator. Format. Text	String	Whether the indicator is numeric, textual or binary	
000075	Language. Code	string	The identifier of the language used in the corresponding text string	Reference ISO 639: 1998
000153	Measure. Content	decimal	The size, volume, mass, amount or scope derived by performing a physical measure	For example, 20 kilograms (20 is the measure content)
000154	Measure Unit. Code	string	The type of unit of measure	Reference UN/ECE Recommendation #20 and X12 355. For example, for \$10/100 km use CCT quantity type and for a measured distance of 20 kilometres use CCT measure type
000183	Numeric. Content	As defined by Numeric. Format.	The representation of a number	May be decimal
000204	Numeric. Format. Text	string	Whether the number is an integer, decimal, real number or percentage	
000205	Picture. Content	binary	A visual representation of a person, object, or scene	
000206	Picture. Format. Text	string	The acronym of the coding scheme used to record the picture	
000109	Quantity. Content	decimal	A number of non-monetary units	
000110	Quantity. Unit. Code	string	The unit of the quantity	May use UN/ECE Recommendation #20 and X12 355, but for actual measurements use the CCT measure type. For example, for \$10/100 km use CCT quantity type and for a measured distance of 20 kilometers use CCT measure type
000112	Quantity Unit Code List Agency. Identifier	string	The agency which maintains the quantity unit code list	
000111	Quantity Unit Code List. Identifier	string	The quantity unit code list	
000094	Text. Content	string	A character string generally in the form of words	

828 **6.1.2 Business Information Entities**

829 A Business Information Entity is a piece of business data or a group of pieces of
 830 business data with a unique business semantic definition. A BIE can be either a Basic
 831 Business Information Entity (BBIE) or an Aggregate Business Information Entity
 832 (ABIE). A BBIE is derived from a Basic Core Component (BCC). An ABIE is a re-
 833 use of an Aggregate Core Component (ACC) in a specified business context. Figure
 834 6-2 describes the BIE types and shows relationships to the core component
 835 counterparts.

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



[Definition] Business Context
 A Unique combination of values for all defined Context categories.

837 [B1] A Business Information Entity is always a "Basic Business Information
 838 Entity" (BBIE) or an Aggregate Business Information Entity (ABIE).

839 [B2] All BBIEs that relate to the same context-free concept form the basis of the
 840 definition of a BCC.

841 [B3] All ABIEs that relate to the same context-free concept form the basis of the
842 definition of an ACC.

843 [B4] An ABIE will consist of two or more BBIEs and/or ABIEs.

844 **6.1.3 Naming Convention**

845 A naming convention is necessary to gain consistency in the naming of all core
846 components and business information entities. The result of consistency facilitates
847 comparison during the discovery and analysis process, and precludes creating multiple
848 core components with different names that have the same semantic meaning.

849 The naming rules are derived from the guidelines and principles described in
850 document ISO 11179 Part 5 -- *Naming and Identification Principles For Data*
851 *Elements*. In certain instances, these guidelines have been adapted to the Core
852 Component environment. In particular, the guidelines have been extended to cover not
853 only the naming of business information entities or data elements but also to cover the
854 naming of Aggregate Information Entities and Core Component Types.

[Issue]

The Naming Convention needs to be examined to see if the rules adequately cover the naming of Basic Business Information Entities (BBIEs) and Aggregate Business Information Entities (ABIEs), particularly in combining context and core naming.

855 6.1.3.1 Dictionary Information

856 Each Core Component contains the following dictionary information that is impacted
857 by the naming rules:

- 858 • **Dictionary Entry Name** (Mandatory). This is the unique official name of
859 the Core Component in the dictionary.
- 860 • **Definition** (Mandatory). This is the unique semantic business meaning of
861 that Core Component.
- 862 • **Business term** (Optional). This is a synonym term under which the Core
863 Component is commonly known and used in the business. A Core
864 Component may have several business terms or synonyms.

Example:

- Dictionary Entry Name (e.g. Account Identifier; Purchase Order Identifier)
- Business Term (e.g. Account Number; Order Number, PO Number)

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

- 866 • **Object Class.** This represents the logical data grouping or aggregation (in
867 a logical data model) to which a data element belongs. The Object Class
868 thus is the part of a Core Component's Dictionary Entry Name that
869 represents an activity or object in a specific context.
- 870 • **Property Term.** This identifies one of the data elements belonging to the
871 Object Class.
- 872 • **Representation Term.** This defines the type of valid values for an
873 information entity.

874 6.1.3.2 Rules

875 The following subsections define all naming convention rules.

876 6.1.3.2.1 General Rules

- 877 [C9] The dictionary content shall be in English Language following the
878 primary Oxford Dictionary English spellings to assure unambiguous
879 spelling.

[Note]

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

880 6.1.3.2.2 Rules for Definitions

- 881 [C10] The definition shall provide an understandable definition, which should also
882 be translatable to other languages.
- 883 [C11] The definition shall take into account the fact that the users of the Core
884 Component dictionary are not necessarily native English speakers. It shall
885 therefore contain short sentences, using normal words. Wherever synonym
886 terms are possible, the definition shall use the preferred term as identified in
887 the Core Components glossary of terms.
- 888 [C12] The definition of a Core Component shall use a structure that is based on the
889 existence of the *Object Class*, the *Property Term*, and its *Representation Term*.
- 890 [C13] Whenever both the definite (i.e. "the") and indefinite article (i.e. "a") are
891 possible in a definition, preference shall be given to the indefinite article (i.e.
892 "a").

[Note]

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

893 6.1.3.2.3 Rules for Dictionary Entry Names

894 [C14] The Dictionary Entry Name shall be unique.

895 [C15] The Dictionary Entry Name shall be extracted from the Core Component
896 definition.

897 [C16] The Dictionary Entry Name of a Core Component Type shall consist of a
898 meaningful type name followed by a dot and the term “Type”.

[Example]

Amount. Type, Date Time. Type

899 [C17] The Dictionary Entry Name of an Aggregate Core Component shall consist of
900 a meaningful aggregate name followed by a dot and the term “Details”. The
901 aggregate name may consist of more than one word.

[Example]

Postal Address. Details, Party. Details

902 [C18] The Dictionary Entry Name of a Core Component shall consist of the name of
903 an *Object Class*, the name of a *Property Term* and the name of a
904 *Representation Term*.

[Example]

Tax. Description. Text

905 [C19] A Dictionary Entry Name shall be concise and shall not contain consecutive
906 redundant words.

907 [C20] The name of an *Object Class* refers to an activity or object within a business
908 context. It shall be unique throughout the dictionary and may consist of more
909 than one word.

910 [C21] The name of a *Property Term* shall occur naturally in the definition and may
911 consist of more than one word. A name of a *Property Term* shall be unique within the
912 context of an *Object Class* but may be reused across different *Object Classes*.

[Example]

“Car. Colour. Code” and “Shirt. Colour. Code” may both exist

913 [C22] If the name of the *Property Term* uses the same word as the *Representation*
914 *Term* (or an equivalent word), this *Property Term* shall be removed from
915 *Dictionary Entry Name*. The *Representation Term* word in this case only will

[Example]

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”.

916 remain.

917 [C23] The name of the *Representation Term* shall be one of the terms specified in the
918 “list of *Representation Terms*” as included in this document (See section
919 6.1.3.3).

920 [C24] The name of the *Representation Term* shall not be truncated in the *Dictionary*
921 *Entry Name*.

922 [C25] A *Dictionary Entry Name* and all its components shall be in singular form
923 unless the concept itself is plural.

924

[Example]

“Goods”

925 [C26] The components of a Dictionary Entry Name shall be separated by dots. The
 926 space character shall separate words in multi-word *Object Classes* and/or
 927 multiword *Property Terms*. Every word shall start with a capital letter. To
 928 allow spell checking of the Directory Entry Names' words, the dots after
 929 Object Class and property terms shall be followed by a space character.

[Note]

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

- It must be clear that Dictionary Entry Names may not be suitable to be used as syntax specific metadata names
- Use of CamelCase will not allow the use of spell checkers
- Strict use of CamelCase makes it impossible to use separators (“.”) and therefore doesn't allow an unambiguous identification of the composing parts of the Dictionary Entry Name

930 [C27] Non-letter characters shall only be used if required by language rules.

931 [C28] Dictionary Entry Names shall only contain verbs, nouns and adjectives (i.e. no
 932 words like “and”, “of”, “the”, etc.). This rule may not be valid for other
 933 languages but English language.

934 [C29] Abbreviations and acronyms that are part of the Dictionary Entry Name shall
 935 be expanded or explained in the definition.

936 6.1.3.2.4 Rules for Business Terms

937 Business terms are those terms that are commonly used for day-to-day information
 938 exchanges within a given domain. As such, no specific naming rules apply to
 939 Business Terms.

940 6.1.3.3 List of Representation Terms

941 The representation term is the part of a Core Component name that describes the form
 942 of representation of a data item. For instance all basic Core Components representing
 943 a monetary amount shall be named “[Name]. Amount” where [Name] represents a
 944 specialisation of the generic amount and Amount is the representation term. Table 6-3
 945 lists the permissible *Representation Terms*.

946 [C30] Used Representation Terms will be taken from the list of permissible
 947 Representation Terms

948 **Table 6-3 Permissible Representation Terms**

949

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

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

950 Table 6-4 contains the permissible representation terms that apply to Aggregate Core
951 Components or Core Component types.

952 [C30] Used Representation Terms for Aggregate Core Components or Core
953 Component Types will be taken from the list of permissible Representation
954 Terms

955 ***Table 6-4 Permissible Representation Terms for Aggregate Core Components or***
956 ***Core Component Types***

957

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

958 **6.1.4 Catalogue of Core Components**

959 Under the ebXML architecture concept, all core components will be recorded in an
960 ebXML compliant registry and stored in a related repository. However, small and
961 medium enterprise (SME) organisations may not be able to readily access just such an

962 architecture. As such, it is important that the full range of UN/CEFACT core
963 components be published in a catalogue. This catalogue must convey the full details
964 of each core component consistent with how those components are stored as UML
965 objects in the repository. Table 6-5 identifies a proper format for the catalogue and
966 contains representative entries from the existing UN/CEFACT CC catalogue.

967 The catalogue is intended to be part of a larger core component library. The core
968 component library will consist of the following parts:

- 969 • Core Component Types
- 970 • Catalogue of Core Components, including BCCs/BBIEs and ACCs
- 971 • Catalogue of Aggregate Business Information Entities

972

972 **Table 6-5. Core Component Catalogue**

973

UID	Disjunctive Binary Name	CCT Used	Basic Aggregate	Definition	Example	Object Class	Inquiry Term	Representation Term	Business Term	Core Component Children
000034	Address Type Code	Code Type	State	The typical 'the address'	For example a business address or a home address. May be 'Postal' or 'the address'	Address	Type	Code		
000147	Base Charge Price Quantity	Quantity Type	State	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 ¹	Quantity		
000139	Base Currency Measurement Code	Code Type	State	The currency measurement side of the rate of exchange.	The base currency measurement by the currency exchange rate gives the second currency amount.	Base Currency	Measurement	Code		

974
975

975 **6.1.5 Catalogue of Business Information Entities**

976 Predefined Business Information Entities are not provided. Rather, the working
977 registries and the groups defining business messages will define them. However, for
978 the same reasons that a catalogue of Core Components is necessary, a Catalogue of
979 BIE's is also required. The groups defining business messages will be responsible for
980 developing a Catalogue of BIE's that is comparable to the Catalogue of Core
981 Components.

982 **6.2 Context**

983 This section fully describes applicable rules and applications for the use of context in
984 core component discovery, analysis, and use to include context categories and their
985 values, and the Constraint Language.

986 [Ed. Note: This section still requires conversion of prose into specific rules. A few
987 representative transformations have taken place to help guide the context team. The
988 context team will conduct this work during the first comment period.]

989 **6.2.1 Overview of Context Specification**

990 Whenever business collaboration takes place between specific trading partners, data is
991 exchanged in the form of business messages. That data exists in a particular business
992 context. In its simplest form, this is the idea of "context" as used in ebXML. The
993 context in which the business process takes place can be specified by a set of
994 categories and their associated values.

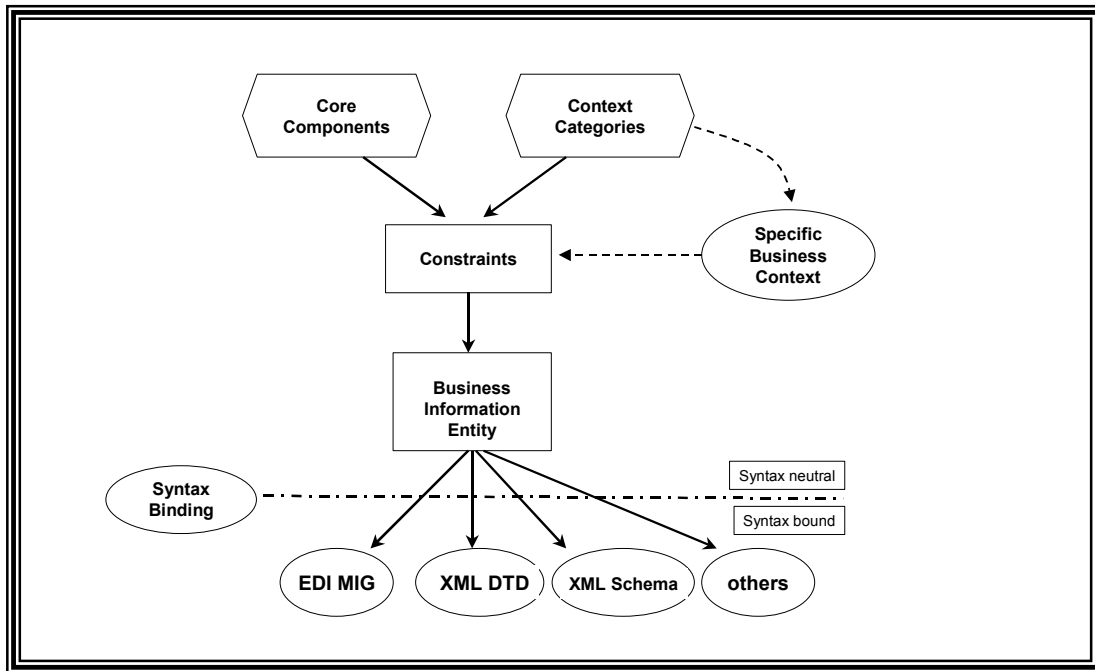
995 The core components have no context independent of their use. The Context
996 mechanism provides a full semantic qualification for the core component used in a
997 business process. In Figure 6-3, the operation of the Context mechanism is illustrated.

998 The ebXML framework provides a mechanism for describing how semantic meaning
999 is given to core components when they are used in a specific business process, that is,
1000 in a specific context. The Business Information Entity resulting from this process can
1001 be manifested as a model, which in turn can be used as the basis of a syntax-bound
1002 business message description (an EDI message implementation guide, an XML DTD
1003 or schema, etc.)

1004 The following sections address the context categories, and the constraint language
1005 more closely.

1006

1006 **Figure 6-3. Operation of The Context Mechanism**



1007 6.2.1.1 Approved Context Categories

1008 [C31] Applied Context will be from an approved category defined in Table 6-6

1009 **Table 6-6. Approved Context Categories**

Context Category	Description
Business Process	The business process as described using the ebXML Catalogue of Common Business Processes and extension mechanism.
Product Classification	Factors influencing semantics that are the result of the goods or services being exchanged, handled, or paid for, etc. (e.g. the buying of consulting services as opposed to materials)
Industry Classification	Semantic influences related to the industry or industries of the trading partners (e.g., product identification schemes used in different industries).
Geopolitical	Geographical factors that influence business semantics (e.g., the structure of an address).
Official Constraints	Legal and governmental influences on semantics (e.g. hazardous materials information required by law when shipping goods).

Context Category	Description
Business Process Role	The actors conducting a particular business process, as identified in the Catalogue of Common Business Processes.
Supporting Role	Semantic influences related to non-partner roles (e.g., data required by a third-party shipper in an order response going from seller to buyer.)
System Capabilities	This context category exists to capture the limitations of systems (e.g. an existing back office can only support an address in a certain form).

1010 6.2.1.2 Constraint Language

1011 A constraint language is used to express the relationship between specific business
1012 contexts and how semantics are applied to the core components to produce Business
1013 Information Entities. The scope of this language covers two functional parts:

- 1014 • “Assembly” of a large aggregate (the “document”);
- 1015 • Refinement of the assembly as appropriate. Refinement is both the
1016 addition of semantics specific to the business process, and the restriction
1017 and extension of the semantic model.

1018 This separation is a convenience for implementation (it simplifies the creation of
1019 processing tools) and creation of “standard” assemblies that can then be refined by
1020 specific users (a process that resembles how EDI standards and message
1021 implementation guides function today).

1022 Both constraint language parts allow, for example, simple commands indicating how
1023 core components will be used, how they will be named for these specific uses, how to
1024 refine the cardinality (if necessary). Further, conditional relationships can be
1025 expressed. Specific context values or sets of values can be tied to the actions

[Example]

If the Geopolitical Process context has a value of “Anywhere in the European Union,” and the specific business context value indicates that the business process occurs in France, then the context-appropriate BIE can be assembled by modifying the correct core component.

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

1026 performed on core components to produce Business Information Entities.

1027 6.2.1.3 Syntax Binding

1028 The Business Information Entity is a model that has no relationship to a specific
1029 syntax. It is intended that any given Business Information Entity can be expressed in
1030 any number of syntaxes. This process is called “syntax binding,” and it may be
1031 possible to express this in an algorithm.

1032 6.2.2 Context Categories Specification

1033 This section specifies the categories used to describe business contexts.

1034 Context categories exist to allow users to uniquely identify and distinguish between
1035 different business contexts. Each of the identified categories – unless otherwise stated
1036 - uses a standard classification to provide values for the category. Note that constraint
1037 rules – and therefore, BIEs – are tied to a particular set of standard classifications for
1038 identifying and distinguishing contexts.

1039 When describing a specific context, a set of values will be assigned to the business
1040 situation being formally described.

1041 6.2.2.1 Business Process Context

1042 In describing a business situation, generally the most important aspect of that situation
1043 is what business activity is being engaged in by those performing it. Business Process
1044 context gives a way to identify the business activity unambiguously.

1045 The Business Process Context category has a standard classification: The
1046 classification provided as part of the UN/CEFACT Catalogue of Core Business
1047 Processes. This classification is hierarchical. Business Process Context values may be
1048 expressed as a single business process at any level, or a set of business processes at
1049 any level. Additionally, these values may be taken from extensions to the business
1050 processes described in the Catalogue of Core Business Processes as provided for in
1051 that document. When extensions are used, they will include full information for each
1052 value sufficient to unambiguously identify which extension is providing the value
1053 used. The recommended value is:

1054 • Catalogue of Common Business Processes

1055 - Custodian: UN/CEFACT

1056 6.2.2.2 Product Classification Context

1057 The Product Classification Context describes those aspects of a business situation
1058 related to the goods or services being exchanged by, or otherwise manipulated, or
1059 concerned, in the business process. A single value or set of values may be used in a

1060 Product Classification. If a hierarchical system of values is used, then these values
1061 may be at any level of the hierarchy. It is necessary to have an additional value
1062 specifying which classification scheme has supplied the values used, if more than one
1063 classification system is being employed.

1064 The following consist of the initial set of recommended product classification context
1065 values. It should be understood that numerous other values also exist, and their use
1066 should be evaluated in the framework of consistency.

- 1067 • Universal Standard Product and Service Specification (UNSPSC)
1068 - Custodian: ECCMA
- 1069 • Standard International Trade Classification (SITC Rev .3)
1070 - Custodian: United Nations Statistics Division (UNSD)
- 1071 • The "Harmonized Commodity Description and Coding System" (HS)
1072 - Custodian: WTO
- 1073 • Classification Of the purposes of non Profit Institutions serving
1074 households (COPI)
1075 - Custodian: UNSD (This provides a mapping between the first three.)

1076 6.2.2.3 Industry Classification Context

1077 The Industry Classification Context provides a description of the industry or sub-
1078 industry in which the business process takes place. An Industry Classification Context
1079 may contain a single value or set of values at any appropriate level of the value
1080 hierarchy. The value hierarchy must be identified. The recommended sets of values
1081 are:

- 1082 • International Standard Industrial Classification (ISIC)
1083 - Custodian: UNSD
- 1084 • Universal Standard Product and Service Specification (UNSPSC) Top-
1085 level Segment [digits 1 and 2] used to define industry.
1086 - Custodian: ECCMA

[Note]

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

1087 6.2.2.4 Geopolitical Context

1088 Geopolitical contexts allow description of those aspects of the business context that
1089 are related to region, nationality, or geographically based cultural factors.

1090 The regional classification allows one or more values to be associated with any
1091 business message or component, according to the following structure.

1092 Global

1093 [Continent]

1094 [Economic Region]

1095 [Country] - ISO 3166.1

1096 [Region] - ISO 3166.2

1097 At any level of the hierarchy, a value may be a single value, a named aggregate, or
1098 cross-border value. These values are structured as follows:

- 1099 • **Single Value:** A single value indicating a single continent, economic
1100 region, country, or region, depending on position within the hierarchy.
- 1101 • **Named Aggregate:** A related group of values (which may themselves be
1102 single values, named aggregates, or cross-border constructions), which
1103 have been related and assigned a name. A named aggregate contains at
1104 least two values.
- 1105 • **Cross-Border:** One or more pairs of values, designated "To", "From", or
1106 "Bi-directional", indicating the direction of cross-border context. Values
1107 may be named aggregates or single values.

[Example]

The following example shows an extract of the basic, single-value hierarchy of recommended values, based on the common ISO 3166 Country Codes. (The value at the top of any hierarchy is always understood to be "Global".)

Europe

 Eastern Europe

 AL – ALBANIA

 AM – ARMENIA

1108 Points in the hierarchy are specified by the use of the node value, or by the full or
1109 partial path. There are cases where the full path is required to understand the
1110 hierarchy, as a result of the use of the more complex constructs. A single-point
1111 specification is understood to inherit all of the properties of the single-value hierarchy
1112 except where otherwise specified.

1113 [C32] Geopolitical Values will be taken from ISO 3166.1 and 3166.2

1114 6.2.2.5 Official Constraints Context

1115 The Official Constraints Context category describes those aspects of the business
1116 situation that result from legal or regulatory requirements and similar "official"
1117 categories. This category is outlined as follows:

- 1118 • Regulatory and legislative (includes customs)
- 1119 • Conventions and treaties (these are different from regulatory and
1120 legislative)

1121 [C33] The Official Constraints Context will consist of at least two values:

- 1122 • Identification of the legal or other classification used to identify the
1123 context values.
- 1124 • Identification of the official constraint itself. These values may represent a
1125 hierarchical structure depending on the official constraints system being
1126 referenced.

1127 Because there is no known global classification of all Official Constraints as used
1128 here, any implementation must provide a set of recognized official constraints
1129 classifications for use within the ebXML Registry implementation.

1130 6.2.2.6 Business Process Role Context

1131 The Business Process Role Context describes those aspects of a business situation that
1132 is specific to an actor or actors within the business process. Its values are taken from
1133 the set of Role values provided by the Catalogue of Core Business Processes. A
1134 Business Process Role Context is specified by using a value or set of values from this
1135 source. The recommended value is:

- 1136 • Catalogue of Common Business Processes
- 1137 - Custodian: UN/CEFACT

1138 6.2.2.7 Supporting Role Context

1139 The Supporting Role Context identifies those parties that are not active participants in
1140 the business process being conducted but who are interested in it. A Supporting Role
1141 Context is specified with a value or set of values from a standard classification. The
1142 recommended value is:

- 1143 • UN/EDIFACT code list for DE 3035Party Roles
- 1144 - Custodian: UN/EDIFACT Technical Assessment

1145 6.2.2.8 System Capabilities Context

1146 This context category identifies a system, a class of systems or standard in the
1147 business situation. The System Capabilities Context requires a least one pair of
1148 values: an identification of the classification scheme being used and a value from that
1149 scheme. A valid System Capabilities Context may include more than one such pair of
1150 values.

[Issue]

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

1151 6.2.3 Context Values

1152 A specific business context is formally described using a set of context values. Every
1153 context category must have a valid value, even if this value is “In All Contexts” or
1154 “None”. None is appropriate for Official constraints because there are no official
1155 constraints.

1156 [C34] The “In All Contexts” value is a valid value for every context category except
1157 for Official Constraints, where the value of “None” is used.

1158 6.2.4 Constraints Language

1159 The constraints language exists to allow users to express the relationships between
1160 specific business situations and the specific structure and meaning of business data
1161 used in that situation. The constraints language refers to specific contexts as described
1162 in the Context Categories specification and uses UIDs to refer to Core Components
1163 semantic models. This section specifies the actions that may be performed on the Core
1164 Components in specific business contexts to produce BIEs.

1165 The constraints language is presented in Table 6-7 as a table of names and a definition
1166 of their constraints, to avoid tying the definition of the constraints to a given syntax.

1167 An “Assembly” is the overall expression of a single set of Assembly Rules, which
1168 groups a set of unrefined BIEs in to a larger structure. When working with pre-
1169 assembled standard document sets, it should not be necessary for users to create these.

1170 [Ed. Note: The next version will have two UML class diagrams inserted here, one for
1171 Assembly and one for ContextRules.]

1172 A “ContextRules” is the overall expression of a single set of Context Rules. These add
1173 the full semantic and structural refinement to the core components to produce BIEs.
1174

1175 *Table 6-7 Constraints Language*

Construct	Component Constructs	Description
Assembly		An Assembly contains at least one Assemble, optionally either an @id or an @idref, and optionally one @version Note: An Assembly is the top level construct in a set of Assembly Rules
	Assemble	List of assembled Core Components 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.
Assemble		An Assemble contains at least either a CreateBIE or a CreateGroup, optionally either an @id or an @idref, and one @name
	CreateBIE	List of Core Components
	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
CreateGroup		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
CreateBIE		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 Core Component

Construct	Component Constructs	Description
	MinOccurs	Minimum occurrences for the BIE created
	MaxOccurs	Maximum occurrences for the BIE created. One possible value (other than integer) is 'unbounded'.
	@id	Id of the created BIE
	@idref	Reference to the ID of another created BIE
	Name	Name of the BIE to be assembled
	@location	Location of the BIE to be assembled (i.e. query to the registry)
	Rename	Renames children of the created BIE
	Condition	Condition under which this rule should apply
	Annotation	Insert Annotation
Name		A Name contains only a string of characters
Type		A Type contains only a string of characters. It represents a type in the output – representation class or core component, depending on where used.
Rename		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
ContextRules		ContextRules contains one or more Rules Note: A ContextRules is the top level construct in a set of Context Rules
	Rule	List of refinement and qualification rules to be applied
	@id	Id of the ContextRules rule
	@idref	Reference to the ID of another ContextRules rule
	@version	Version of the ContextRules document.
Rule		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

Construct	Component Constructs	Description
	Taxonomy	List of taxonomies used in a Rule that employs hierarchical conditions.
Taxonomy		A Taxonomy contains a @context and a @ref, and optionally an @id or an @idref
	@ref	Pointer to a taxonomy.
	@context	Name of the context category to which this Taxonomy applies
	@id	Id of the Taxonomy rule
	@idref	Reference to the ID of another Taxonomy rule
Condition		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.
Action		An Action contains at least one of Add or Occurs or Subtract or Condition or Comment or Rename, one @applyTo and optionally an @id or an @idref
	@applyTo	Node to apply action to
	Add	Add a component to the content model
	Subtract	Subtract a component from the content model
	Occurs	Constrain or expand the number of occurrences of the component
	Condition	When rule should be run
	Comment	Add a comment
	Rename	Rename a component
	@id	Id of the Condition rule
	@idref	Reference to the ID of another Condition rule
	@applyTo	Name of the component to apply this rule to
Add		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

Construct	Component Constructs	Description
	MinOccurs	Minimum number of times that the new instance must occur
	MaxOccurs	Maximum number of times that the new instance can occur
	@before	Specifies before which component the addition should occur.
	@after	Specifies after which component the subtraction should occur.
	CreateGroup	Create a group of BIEs
	BIE	Adds a new BIE to the content model.
	Attribute	Adds a new non-BIE property to the content model
	Annotation	Insert Annotation
	@id	Id of the Add rule
	@idref	Reference to the ID of another Add rule
Subtract		Subtract contains one or more of BIE or Attribute, and optionally an @id or an @idref
	BIE	Removes a BIE from the content model.
	Attribute	Removes a non-BIE property from the content model
	@id	Id of the Subtract rule
	@idref	Reference to the ID of another Subtract rule
Occurs		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
BIE		A BIE contains a Name, followed by an optional Type, followed by zero or more Attribute, followed by zero or more Annotation, and optionally an @id or an @idref
	Name	Name of BIE to be modified
	Type	Type of BIE – the Core Component - required only if contained in an Add tag
	Attribute	Attribute(s) of this BIE
	Annotation	Insert Annotation
	@id	Id of the BIE rule

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

Construct	Component Constructs	Description
		@id or an @idref
	@id	Id of the Documentation
	@idref	Reference to the ID of another Annotation
Appinfo		Documentation contains optionally an @id or an @idref
	@id	Id of the Appinfo
	@idref	Reference to the ID of another Appinfo

1176

[Note]

Table Key: @ indicates properties of the construct being defined (@id, @idref and @version are properties of Assembly)

1177 6.2.4.1 Notes about Assembly

1178 [C35] The MinOccurs and MaxOccurs constructs in the CreateBIE construct specify
 1179 the occurrence that the created BIE will have in the resulting semantic model.
 1180 Thus, a BIE created with MinOccurs = 1 and MaxOccurs = 1 should
 1181 be specified in the resulting semantic model as occurring only once.

1182 [C36] An Assembly may contain more than one assembled top-level semantic
 1183 model.

1184 6.2.4.2 Notes about Context Rules

1185 Several built-in variables are used to access context information. These variables
 1186 correspond to the identified context categories. All of these variables have string
 1187 values.

1188 The “Apply” attribute of the “Rule” construct type is used for determining the
 1189 behaviour of rules that use hierarchical values. Possible values are:

- 1190 • “exact” - match only if the value in the provided context is precisely the
 1191 same as that specified in the rule
- 1192 • “hierarchical” - match if the value provided is the same or a child of that
 1193 specified in the rule.

[Example]

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

1194 The “Attribute” construct has four optional children in its content model, of which at
1195 least one must be present.

1196 When the “Attribute” construct is used to refine an existing “Attribute”, then a value
1197 must be specified for @applyTo on that “Attribute” construct.

1198 When writing Rules, they must refer to the names of the core components, and not the
1199 names given to the resulting BIEs elsewhere in the Rules.

[Example]

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.

1200 6.2.4.3 Output Constraints

1201 Semantic models and document definitions produced through the application of
1202 Assembly and Context Rules must contain the metadata about the rules and context
1203 that produced them.

1204 6.2.4.4 Ordering and Application

1205 There is an explicit “Order” property on the “Rule” construct that applies a sequence
1206 to the application of a set of rules. It is an error for two “Rule” constructs to have the
1207 same value for the property “Order.” In a single set of Context Rules, users should be
1208 careful not to sequence rules in a way that would preclude their execution. (e.g.
1209 adding an attribute to a BIE that has not been added yet by the rules).

1210 **7 Technical Details - Core Component Repository**

1211 **Storage**

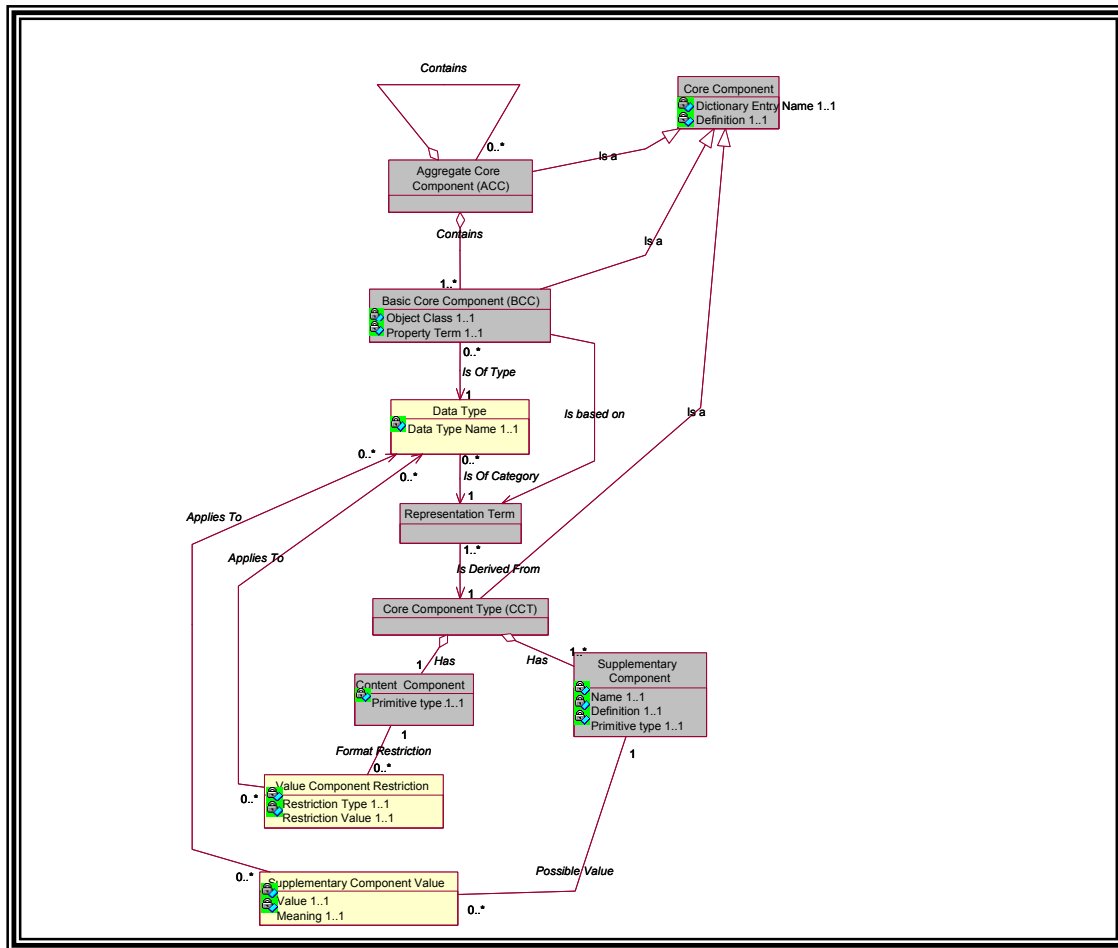
1212 Section 6.1 specifies the Core Component basic definition. This section details exact
1213 information required to create UML objects to store core components in the repository
1214 and to store relevant metadata about the core components. Both parts contain
1215 requirements that must be addressed by developers of core components and users of core
1216 components. Further, both parts contain requirements that must be satisfied in the
1217 supported registry and repository suite of technical specifications and any corresponding
1218 overarching information technology framework that uses core components as the linchpin
1219 between process modeling and trade.

1220 **7.1 Storing Core Components**

1221 This section fully describes Core Component storage details. Figure 7-1 is the UML
1222 model of all aspects of Core Components and fully describes the types of Core
1223 Components and their relationships as a requirement of storage.

1224

1224 **Figure 7-1. Core Components - Full Definition**



1225 **7.1.1 Stored Core Components**

1226 [S1] Stored Core Components will always be defined as one of the three recognized
 1227 types—Basic Core Component, Aggregate Core Component
 1228 or Core Component Type.

1229 [S2] Stored Core Components will include the following attributes:

- 1230 • Dictionary Entry Name 1..1: where the Dictionary Entry Name is the unique
 1231 official name of the Core Component in the dictionary.
- 1232 • Definition 1..1: where the Definition is is the unique semantic business
 1233 meaning of the Core Component.

1234 7.1.2 Stored Basic Core Components

1235 [S3] Stored Basic Core Components will always be based on three elements: (1) an
1236 Object Class, which defines the overall business concept to which the BCC
1237 belongs, (2) a Property Term, which defines the specific characteristic of the
1238 business concept that is covered by the BCC and (3) a Data Type.

1239 [S4] Stored Basic Core Components will include the following Attributes:

1240 • **Object Class 1..1:** where the Object Class represents the logical data
1241 grouping (in a logical data model) to which a data element belongs
1242 (ISO11179). The Object Class is the part of a Core Component's Dictionary
1243 Entry Name that represents an activity or object in a specific context.

1244 • **Property Term 1..1:** where the Property Term identifies one of the
1245 characteristics belonging to the Object (Class)

1246 [S5] Stored Basic Core Components will reflect association with a stored Core
1247 Component Type.

1248 [S6] Stored Basic Core Component Data Types will be based on a Representation
1249 Term derived from a Core Component Type.

1250 7.1.3 Stored Core Component Types

1251 [S7] Stored Core Component Types will include one Content Component that defines
1252 the primitive type and one or more Supplementary Components that give meaning
1253 to the Content Component.

1254 [S8] Stored Core Component Types will not reflect business meaning.

1255 7.1.4 Stored Aggregate Core Components

1256 [S9] Stored Aggregate Core Components will consist of two or more Basic core
1257 Components, or at least one Basic Core Component plus one or more Aggregate
1258 Core Components.

[Note]

As shown in Figure 7-1, when the CCT is used as the basis for a particular data type, the content component and supplementary components can be restricted. This is expressed in the diagram through the existence of the classes Supplementary Component Value" and "Content component Restriction" and their "Applies To" relation with "Data Type".

1259 [S10] Stored Aggregate Core Components will reflect relationships between the Basic
1260 Core Components and Aggregate Core Components from which it is constructed.

1261 **7.1.5 Stored Content Components and Supplementary Components**

1262 [S11] Restrictions on Stored ContentComponents and Supplementary Components
1263 will be identified when the Core Component Type is used as basis for a
1264 particular Data Type.

1265 **7.1.6 Stored Data Types**

1266 [S12] Stored Data Types will define the full range of valid values that can be used for a
1267 particular Basic Core Component and will include the following attribute:

- 1268 • **Data Type Name 1..1:** Official name of the Data Type.

1269 **7.1.7 Stored Representation Term**

1270 [S13] Stored Representation Terms for Basic Core Components will define the type of
1271 valid value and will include the following attribute:

- 1272 • **Representation Term Name 1..1:** Official name of the Representation Term

1273 **7.1.8 Stored Supplementary Components**

1274 [S14] Stored Supplementary Components will be associated with the
1275 ContentComponent in the overarching Core component Type and will include the
1276 following attributes:

- 1277 • **Name 1..1:** Name is the official name of the supplementary component of a
1278 Core Component Type.
- 1279 • **Definition 1..1:** Definition is a clear, unambiguous and complete explanation
1280 of the meaning of a Supplementary Component and its relevance for the
1281 related Core Component Type.
- 1282 • **Primitive type 1..1:** Primitive type to be used for the representation of the
1283 value of a Supplementary Component.

[Note]

Possible values are String, Decimal, Integer, Boolean, Date.

1284 **7.1.9 Stored Supplementary Component Value**

1285 [S15] A stored Supplementary Component Value shall define an enumerated list of
1286 possible values of a Supplementary Component.

- 1287 [S16] A stored SCV will only be stored if the values can be defined by an enumeration
1288 (e.g. list of quantity units).

[Note]

The list of possible Stored Supplementary Component values can be further restricted when a Core Component Type is used for a particular Basic Component. Example: the Core Component Type "Quantity" has a supplementary component "Quantity Unit" with possible values like gram and second. A Basic Component like "Person. Weight. Quantity" will not accept "second" as quantity unit.

The list can still be further restricted when used in a particular context.

1289 **7.1.10 Stored Supplementary Component Values**

- 1290 [S17] Stored Supplementary Component Values will contain the following Attributes:

- 1291
- Value 1..1: Value is a possible value of a Supplementary Component.
- 1292
- Meaning 1..1: Meaning describes the meaning of the Supplementary
- 1293 Component when it has a particular Value.

1294 **7.1.11 Stored Content Components**

- 1295 [S18] Stored Content Components will contain the primitive type that must be used to
1296 express the value of a CCT.

- 1297 [S19] Stored Content Components will contain the following attribute:

- 1298
- **Primitive type 1..1:** Primitive type to be used for the representation of the
- 1299 value of a Core Component Type.

1300 **7.1.12 Stored Content Component Restrictions**

[Note]

Possible values are String, Decimal, Integer, Boolean, Date.

- 1301 [S20] Stored Content Component Restrictions will define a format restriction that
1302 applies to the possible values of a Content Component.

- 1303 [S21] Stored Content Component Restrictions will only exist if the values can be
1304 defined by a format restriction such as string pattern, minimum or maximum
1305 length, or enumeration.
- 1306 [S22] Stored Content Component Restrictions will contain the following attributes:
- 1307 • Restriction Type 1..1: Restriction Type defines the type of format restriction
1308 that must be applied to the Content Component.
- 1309 • Restriction Value 1..1: Restriction Value is the actual value of the Restriction
1310 Type that applies to a Content Component.

[Note]

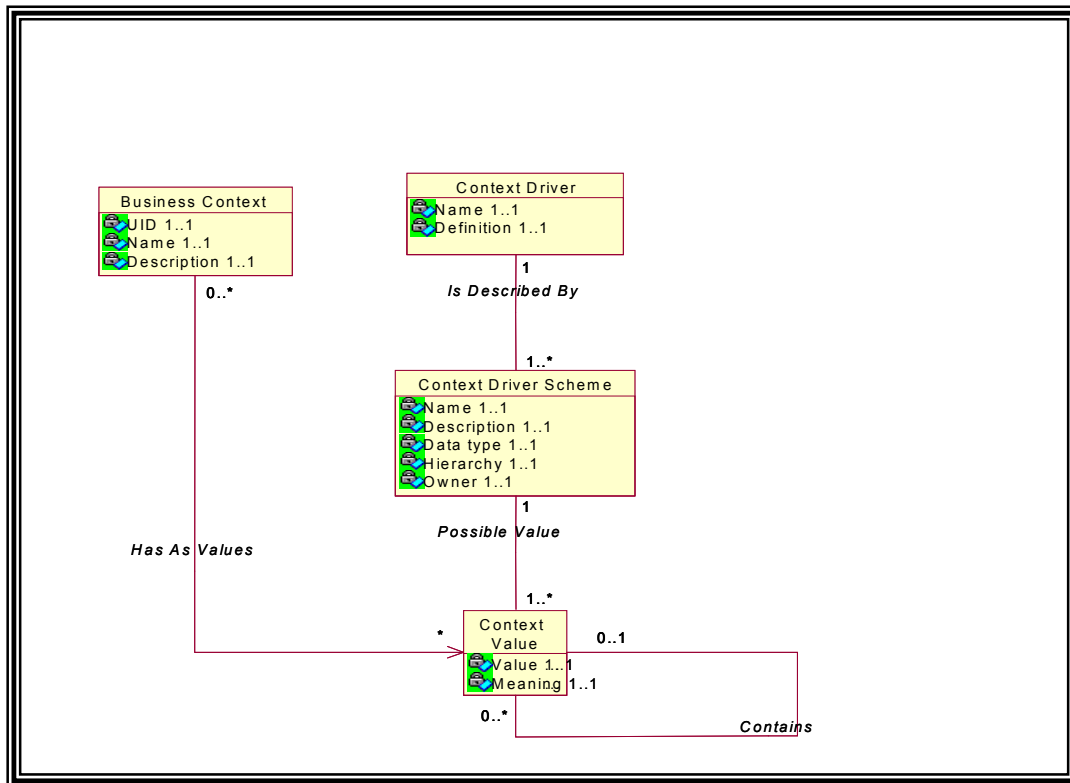
Possible values include pattern, length, minimum length, maximum length, enumeration, and others to be identified.

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

1311 **7.2 Stored Context**

1312 Figure 7-2 models the function of context for storage. It shows that there are a number of
1313 Context Categories (e.g. Region, Product), which can each be described by one or more
1314 Schemes (e.g. UN scheme for products, WTO scheme for products, ...). For each Scheme
1315 the list of possible values (and their meaning) is defined. A "Business Context" is then
1316 defined as a unique and meaningful combination of Context Values.

1317

1317 **Figure 7-2 Core Components Context Definition Model**

1318 [S23] Stored Business Context will contain the combination of values for context
 1319 categories so as to define a unique and meaningful business context.

1320 [S24] Stored Business Contexts will contain the following attributes:

- 1321 • **UID 1..1:** Unique Identifier of a Business Context
- 1322 • **Name 1..1:** Name of the Business Context
- 1323 • **Description 1..1:** Description of the meaning of a Business Context

1324 **7.2.1 Context categories**

1325 [S25] Stored Context Categories will be in conformance with the officially accepted
 1326 categories of Core Component contexts.

1327 [S26] Stored Context Categories will contain the following attributes:

- 1328 • **Name 1..1:** Name is the official name of the Context categories.

- 1329 • Definition 1..1: Definition gives the meaning of the Context categories for
1330 Core Components.

1331 **7.2.2 Context categories Scheme**

- 1332 [S27] Stored Context categories Scheme will be an officially supported Scheme to
1333 describe a given Context categories.

- 1334 [S28] A Context categories may be described by one or more Context categories
1335 Schemes.

[Issue]

Should Context categories Scheme be included. In other words, can a scheme be defined and used. At least some members of the team believe that they should be named so that users can define locally significant extensions to schemes defined by other people—such as extending ISO 3166 to allow the separate regions of the United Kingdom to be identified as South-West, South-East,

- 1336 [S29] Stored Context Categories Schemes will contain the following attributes:

- 1337 • **Name 1..1:** Name under which the Context categories Scheme is known.
- 1338 • **Description 1..1:** Description of the Context categories Scheme.
- 1339 • **Data type 1..1:** Data type is the primitive type that is used for the
1340 representation of a value in the Context categories Scheme.

[Note]

Possible values are String, Decimal, Integer, Boolean, Date.

- 1341 • **Hierarchy 1..1:** Indicator describing whether the Context categories Scheme
1342 supports a hierarchical description of the context.
- 1343 • **Owner 1..1:** Organisation that is responsible for the Context categories
1344 Scheme

1345 **7.2.3 Context Value**

- 1346 [S30] Stored Context Value(s) will describe a particular context in a given Context
1347 categories according to a particular Context categories Scheme. If the Context
1348 categories Scheme allows a hierarchy, the "Contains" value describes this
1349 hierarchy.

1350 [S31] Stored Context Value(s) will contain the following attributes:

1351 • Attributes:

1352 • Value 1..1: Value describing a particular context.

1353 • Meaning 1..1: Description of the meaning of the corresponding value.

1354 **7.3 Stored Business Information Entities**

1355 Figure 7-3 models the types of Business Information Entities and their relationships. A
1356 Business Information Entity is defined as "a piece of business data or a group of pieces of
1357 business data with a unique business semantic definition in a specified business context".

1358 A Business Information Entity is always of one of the following types:

1359 • A "Basic Business Information Entity" (BBIE) is a piece of business
1360 information with a unique concept having a single business semantic
1361 definition in a specified business context

1362 • An "Aggregate Business Information Entity" (ABIE) is a collection of related
1363 pieces of business information that together convey a distinct business
1364 meaning in a specified business context.

1365 A Business Context is defined as a unique combination of values for all defined Context
1366 Categories. For a given business context it is possible to define business terms and
1367 examples, to specify an alternative definition and name and to restrict the data type (only
1368 for BBIE).

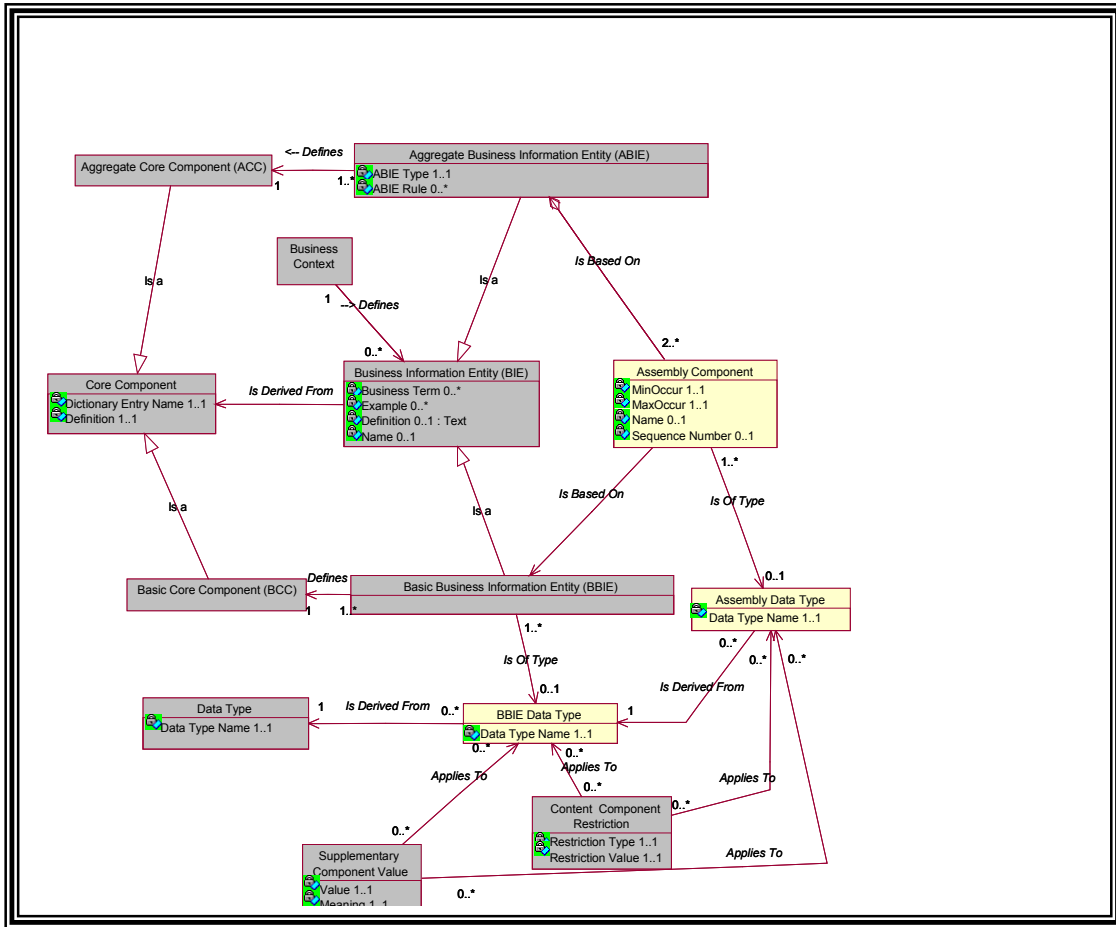
1369 All BBIEs that relate to the same context-free concept form the basis of the definition of a
1370 BCC.

1371 All ABIEs that relate to the same context-free concept form the basis of the definition of
1372 an ACC.

1373 An ABIE is either a sequence or a choice and will consist of two or more Assembly
1374 Components, which are either BBIEs or ABIEs. Each Assembly Component has a certain
1375 cardinality (i.e. it is mandatory, optional and/or repetitive) and - in case of a sequence - a
1376 sequence number. When used as an Assembly Component, it is possible to change the
1377 name of the composing ABIE or BBIE and to restrict the data type of a composing BBIE.

1378

1378 **Figure 7-3. Business Information Entities – Full Definition**



1379

[Issue]

Figure 7-3 shows the Aggregate and Basic Business Information Entities “define” the Core Components they are derived from. Some in the team believe that Whilst they may be used to identify holes in the CC list, they are in general, more likely to be derived from existing CCs rather than the source of new ones. (Note that Data Type is marked as is derived from, which is inconsistent with the other CC components.)

1380 **7.3.1 Stored Aggregate Business Information Entities**

1381 [S32] Stored Aggregate Business Information Entities will consist of a collection of
1382 related pieces of business information and will convey a distinct business meaning
1383 in a specified business context.

1384 [S33] Stored Aggregate Business Information Entities shall contain the following
1385 attributes:

1386 • **ABIE Type 1..1**: ABIE Type indicates whether the composing components of
1387 the ABIE form a sequence (i.e. all composing components may occur when
1388 the ABIE is used) or a choice (i.e. only one of the composing components may
1389 occur when the ABIE is used).

1390 • **ABIE Rule 0..***: ABIE Rule describes a restriction that relates to various
1391 Assembly Components of the ABIE.

1392 **7.3.2 Stored Assembly Component**

1393 [S34] A stored Assembly Component will be either a ABIE or a BBIE that is a
1394 component of an ABIE. It will specify the cardinality, and may specify the
1395 alternative name and the sequence number to be used.

1396 [S35] Stored Assembly Components will contain the following attributes:

1397 • **MinOccur 1..1**: Minimum number of occurrences that a composing BIE must
1398 occur when used in an ABIE. If the minimum is zero, the component is
1399 optional. If the minimum is one or more, the component is mandatory.

1400 • **MaxOccur 1..1**: Maximum number of occurrences that a composing BIE may
1401 occur when used in an ABIE. If the maximum is zero, the component is not
1402 allowed. If the maximum is more than one, the component is repetitive.
1403 Remark that the defined maximum must always be greater than or equal to the
1404 defined minimum.

1405 • **Name 0..1**: Optional alternative name to be used for a BIE when used in an
1406 ABIE.

1407 • **Sequence Number 0..1**: Position of the Assembly Component in an ABIE of
1408 type Sequence.

1409 **7.3.3 Stored Assembly Data Type**

1410 [S36] Stored Assembly Data Types will define the set of valid values that can be used
1411 for a particular BBIE when used in a particular ABIE.

1412 [S37] Stored Assembly Data Types will be defined by specifying restrictions on the
1413 Content Component and Supplementary Components.

1414 [S38] Stored Assembly Data Types will contain the following attribute:

- 1415 • **Data Type Name 1..1**: Official name of the Data Type.

1416 **7.3.4 Basic Business Information Entity (BBIE)**

1417 [S39] Stored Basic Business Information Entities will define a piece of business
1418 information with a unique concept having a single business semantic definition in
1419 a specified business context.

1420 **7.3.5 BBIE Data Type**

1421 [S40] Stored BBIE Data Types will define the set of valid values that can be used for a
1422 particular BBIE. It will be defined by specifying restrictions on the Content
1423 Component and Supplementary Components.

1424 [S41] Stored BBIE Data Types shall contain the following attributes:

- 1425 • **Data Type Name 1..1**: Official name of the Data Type.

1426 **7.3.6 Business Information Entity**

1427 [S42] Stored Business Information Entities (BIE) will have a unique business semantic
1428 definition in a specified business context.

1429 [S43] Stored BIEs will be categorized as either a Basic Business Information Entity
1430 (BBIE) or an Aggregate Business Information Entity (ABIE).

1431 [S44] A stored Business Information Entity will contain the following attributes:

- 1432 • **Business Term 0..***: A synonym term under which the Core Component is
1433 commonly known and used in the business. A Core Component may have
1434 several business terms or synonyms.
- 1435 • **Example 0..***: An example of a possible value of a Core Component in a
1436 given business context
- 1437 • **Definition 0..1**: Context dependent definition of a Core Component
- 1438 • **Name 0..1**: Context dependent name of a Core Component

1439 **7.3.7 Data Type**

1440 [S45] Stored Data Types will contain the set of valid values that can be used for a
1441 particular BCC.

1442 [S46] Stored Data Types will be defined by specifying restrictions on the CCT that
1443 forms the basis of the Representation Term from which the Data Type is derived.

1444 **7.3.8 Supplementary Component Value**

1445 [S47] Stored Supplementary Component Value(s) will define an enumerated list of
1446 possible values of a Supplementary Component. This will only exist if the values
1447 can be defined by an enumeration (e.g. list of quantity units).

1448 [S47] Stored Supplementary Component Values will contain the following attributes:

- 1449 • **Value 1..1:** Value is a possible value of a Supplementary Component.
- 1450 • **Meaning 1..1:** Meaning describes the meaning of the Supplementary
1451 Component when it has a particular Value.

[Note]

The list of possible values can be further restricted when a Core Component Type is used for a particular Basic Component. Example: the Core Component Type "Quantity" has a supplementary component "Quantity Unit" with possible values like gram and second. A Basic Component like "Person. Weight. Quantity" will not accept "second" as quantity unit.

The list can still be further restricted when used in a particular context.

1452 **7.3.9 Stored Content Component Restriction**

1453 [S48] Stored Content Component Restrictions will define a format restriction that
1454 applies to the possible values of a Content Component.

1455 [S49] Stored Content Component Restrictions will only exist if the values can be
1456 defined by a format restriction such as string pattern, minimum or maximum
1457 length, or enumeration.

1458 [S50] Stored Content Component Restrictions will contain the following attributes:

- 1459 • **Restriction Type 1..1:** Restriction Type defines the type of format restriction
1460 that must be applied to the Content Component. Possible values include
1461 pattern, length, minimum length, maximum length, enumeration, etc.
- 1462 • **Restriction Value 1..1:** Restriction Value is the actual value of the Restriction
1463 Type that applies to a Content Component. The possible values depend on the
1464 restriction type (e.g. integer for a "length" restriction type, list of possible
1465 values for an "enumeration" restriction type, ...).

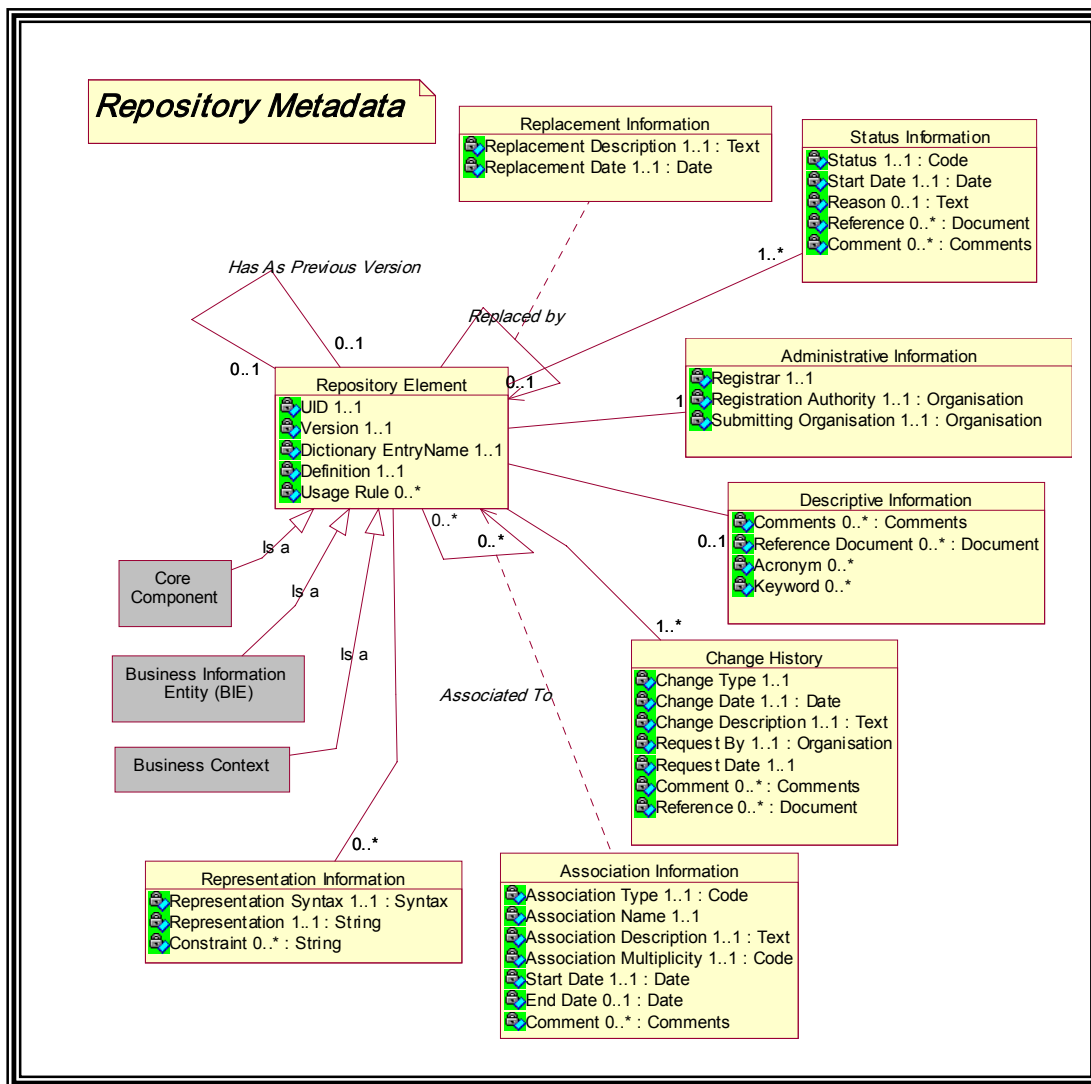
1466 **7.4 Core Component Storage Metadata**

1467 Core Components and Business Information entities are the linchpins for developing
 1468 standardized process models and business documents. Storing these artefacts so that they
 1469 are able to meet this role requires rich metadata storage as well.

1470 Figure 7-4 focuses on the "meta-information" that needs to be defined for a Repository
 1471 Element (i.e. all information needed to store for Core Components and for Business
 1472 Information Entities). To simplify the diagram all information regarding the structure of a
 1473 Core Component and a Business Information Entity has been hidden.

1474 **Figure 7-4. Repository Metadata**

1475 [Ed. Note: Figure 7-4 requires correction to reflect storage of context as part of the



1476 repository element stored data]

1477 As shown in Figure 7-4, the following metadata categories will be required:

- 1478 • Version Information: even though at any given point in time only one version
1479 of a Repository Element can be valid, multiple previous versions may have
1480 existed and a future version may be in preparation. The "Version" association
1481 makes it possible to link the consecutive versions of a Repository Element.
1482 There will not be branches in the versioning; only a linear versioning will be
1483 supported.
- 1484 • Replacement Information: a Repository Element may be replaced by another
1485 Repository Element at some point in time (e.g. because a duplicate is
1486 discovered). The "Replaced by" association makes it possible to do this and
1487 "Replacement Information" makes it possible to document the date and reason
1488 of replacement.
- 1489 • Status Information: information about the live status of a Repository Element
1490 Administrative Information: information about the registration of the
1491 Repository Element.

[Example]

To indicate that there is a relation between an Organisation and a Postal Address. The "Associated To" association can be used for this and "Association Information" can be used to document additional information about the association.

- 1492 • Descriptive Information: additional descriptive information about a Repository
1493 Element, giving further clarification about its meaning.
- 1494 • Change History: information about all changes that are made to a Repository
1495 Element.
- 1496 • Association Information: a Repository Element may be associated to multiple
1497 other Repository Elements.
- 1498 • Representation Information: information about the physical representation of a
1499 Repository Element in a particular syntax (e.g. to document the XML-tag).

1500 **7.4.1 General Metadata Storage Rules**

- 1501 [S51] Stored Repository Elements will include a unique identifier (UID).
- 1502 [S52] Stored Repository Elements will include a Version number to keep track of the
1503 evolution over time of a Repository Element.
- 1504 [S53] Stored Repository Elements may include one or more Usage Rules, describing
1505 how and/or when to use the Repository Element.
- 1506 [S54] Except for the first version of a Repository Element, each stored version will be
1507 linked to its previous version.

1508 [S55] Except for the last version of a Repository Element, each stored version will be
1509 linked to its next version.

1510 [S56] Stored Repository Elements will include the history of the status lifecycle of each
1511 Version.

1512 **7.4.2 Administrative Information**

1513 [S57] Stored Repository elements will contain administrative information and will
1514 include the following attributes:

- 1515 • Registrar 1..1: Name of the responsible person who has created the Repository
1516 Element in the repository
- 1517 • Registration Authority 1..1: Organisation authorised to register the Repository
1518 Element.
- 1519 • Submitting Organisation 1..1: The organisation that has submitted / requested
1520 the Repository Element

1521 **7.4.3 Association Information**

1522 [S58] Stored Repository Elements will include all associations they have with other
1523 stored Repository Elements and will include the following attributes:

- 1524 • Association Name 1..1: Name of the association
- 1525 • Association Description 1..1: Descriptive text explaining the meaning of the
1526 association
- 1527 • Association Type 1..1: Type of association (e.g. aggregation, specialisation,
1528 generalisation, simple association ...)
- 1529 • Association Multiplicity 1..1: Cardinality of the association (i.e.
1530 optional/mandatory and repetition)
- 1531 • Start Date 1..1: Date at which the association becomes valid
- 1532 • End Date 0..1..1*: Date from which the association is no longer valid
- 1533 • Comment 0..*: Relevant information about the association (e.g. reason why it
1534 has been removed, ...)

1535 **7.4.4 Change History**

1536 [S59] Stored Repository Elements will include the history of all modifications related to
1537 each version to include the following attributes:

- 1538 • Change Type 1..1: Purpose of the Change—such as “new element”, “new
1539 version”, “element modification”, “status modification”, “element
1540 replacement”.
- 1541 • Change Date 1..1: Date on which the modification has been made
- 1542 • Change Description 1..1: Description of why and how the Repository Element
1543 has been modified.
- 1544 • Request By 1..1: Name of the organisation that has requested the modification
1545 of the Repository Element
- 1546 • Request Date 1..1: Date on which the modification was requested.
- 1547 • Comment 0..*: Remark about the Repository Element modification.
- 1548 • Reference 0..*: External Document(s) containing relevant information about
1549 the modification.

1550 7.4.5 Descriptive Information

- 1551 [S60] Stored Repository Elements may optionally include additional descriptive
1552 information to include the following attributes:
- 1553 • **Comments 0..***: Comments is additional information about a Repository
1554 Element, which is not part of the definition but that is considered relevant for
1555 clarification.
 - 1556 • **Reference Document 0..***: Reference Document is a reference (e.g. URL) to
1557 external documentation that contains relevant additional information about a
1558 Repository Element.
 - 1559 • **Acronym 0..***: Acronym is an abbreviation or code under which the Semantic
1560 Information Component is commonly known.
 - 1561 • **Keyword 0..***: Keyword is one or more significant words used for the search
1562 and retrieval of a Semantic Information Component.

1563 7.4.6 Replacement Information

- 1564 [S61] For each stored pair of Repository Elements where one Repository Element
1565 replaces the other, the stored information will specify replacement information to
1566 include the following attributes:
- 1567 • Replacement Description 1..1: Reason for the Repository Element being
1568 replaced
 - 1569 • Replacement Date 1..1: Date from which the replacement is effective.

1570 [S62] If a Repository Element has been replaced by another Repository Element, it will
1571 be linked to the Repository Element by which it has been replaced.

1572 [S63] If a Repository Element replaces one or more other Repository Element, it will be
1573 linked to the Repository Element(s) it replaces.

1574 **7.4.7 Representation Information**

1575 [S64] Stored Repository Elements may optionally include information about the
1576 representation of the Repository Element in one or more syntaxes to include the
1577 following attributes.

- 1578 • Representation Syntax 1..1: Identification of the representation syntax
- 1579 • Representation 1..1: Physical representation of the Repository Element (e.g.
1580 XML-tag)
- 1581 • Constraint 0..*: Description of additional constraints that apply to the
1582 representation of the Repository Element in the given syntax (e.g. maximum
1583 length, ...)

1584 **7.4.8 Status Information**

1585 [S65] Stored Repository Elements will contain status information to include the
1586 following attributes:

- 1587 • Status 1..1: Status of the Repository Element (i.e. draft, provisionally
1588 registered, registered, to be retired, retired, ...)
- 1589 • Start Date 1..1: Date on which the status comes into effect
- 1590 • Reason 0..1: Description of why the Repository Element status has been
1591 changed.
- 1592 • Reference 0..*: External Document(s) containing relevant information about
1593 the status change.
- 1594 • Comment 0..*: Remark about the Repository Element status.
1595

1596 **8 Definition of Terms**

1597 **Aggregate Business Information Entity – (ABIE)** A collection of related pieces of
1598 business information that together convey a distinct business meaning in a specified
1599 business context.

1600 **Aggregate Composition Value Restriction –** A Restriction on the possible values for
1601 a Supplementary Component of a Core Component Type when the corresponding
1602 Business Information Entity is used indirectly (i.e. via another Aggregate Information
1603 Entity) in an Aggregate Information Entity.

Example:

The Business Information Entity "Financial Account.Country.Identifier" could restrict the allowed value of the "Identification.Scheme.Name" to "ISO list of country codes."

Remarks:

There are two possibilities:

- If the value of the Supplementary Component is fixed the Representation Term can be specialised (e.g. "ISO Country Identifier").
- If the value of the Supplementary Component is not fixed, the user will have to specify the value of the Supplementary Component each time he uses the Business Information Entity.

1604 **Aggregate Core Component - (ACC)** – A collection of Core Components that
1605 convey a distinct business meaning. An ACC will consist of two or more Basic Core
1606 Components, or at least one Basic Core Component plus one or more Aggregate Core
1607 Components.

1608 **Aggregate-Aggregate Composition Information –** Specifies additional information
1609 when an Aggregate Information Entity is used in another Aggregate Information
1610 Entity. This gives the ability to define the cardinality of the component in the
1611 aggregate as either mandatory, optional, repetitive.

1612 **Aggregate-Aggregate Context Information –** The Influence of a particular context
1613 on the additional information when an Aggregate Information Entity is used in
1614 another Aggregate Information Entity.

1615 **Aggregate-Basic Composition Information –** Specifies additional information when
1616 the Business Information Entity is used in an Aggregate Information Entity.

1617 **Basic Business Information Entity** – A Basic Business Information Entity is derived
1618 from a Basic Core Component.

1619 **Basic Composition Value Restriction** – Restriction on the possible values for a
1620 Supplementary Component of a Core Component Type when the corresponding
1621 Business Information Entity is used in an Aggregate Information Entity.

Example:

The Business Information Entity "Financial Account.Country.Identifier" could restrict the allowed value of the "Identification.Scheme.Name" to "ISO list of country codes."

Remarks:

There are two possibilities:

If the value of the Supplementary Component is fixed the Representation Term can be specialised (e.g. "ISO Country Identifier").

If the value of the Supplementary Component is not fixed, the user will have to specify the value of the Supplementary Component each time he uses the Business Information Entity.

1622 **Basic Core Component** – A Core Component with a unique concept having a single
1623 business semantic definition. It must be constructed by using a Core Component
1624 Type.

1625 **Basic-Aggregate Context Info** – The influence of a particular context on the
1626 additional information when a Basic Information Entity is used in an Aggregate
1627 Information Entity.

1628 **Business Information Entity (BIE)** – A Business Information Entity is a piece of
1629 business data or a group of pieces of business data with a unique business semantic
1630 definition. A BIE can be either a Basic Business Information Entity (BBIE) or an
1631 Aggregate Business Information Entity (ABIE).

1632 **Business Term** – This is a synonym term under which the Core Component is
1633 commonly known and used in the business. A Core Component may have several
1634 business terms or synonyms.

1635 **Constraint Language** – A formal expression of actions occurring in specific contexts
1636 to assemble, structurally refine, and semantically qualify core components. The result
1637 of applying the constraint language to a set of core components in a specific context is
1638 a set of BIEs.

1639 **Content Component** - Defines the primitive type used to express the content of a
1640 CCT.

1641 **Context** – The formal description of a specific business circumstance as identified by
1642 the values of a set of context categories, allowing different business circumstances to
1643 be uniquely distinguished.

1644 **Context Basic Composition Value Restriction** – The influence of a particular
1645 context on the restriction on the possible values for a Supplementary Component of a
1646 Core Component Type when the corresponding Business Information Entity is used in
1647 an Aggregate Information Entity.

[Example]

The Business Information Entity "Financial Account.Country.Identifier" could restrict the allowed value of the "Identification.Scheme.Name" to "ISO list of country codes."

Remarks:

There are two possibilities:

- If the value of the Supplementary Component is fixed the Representation Term can be specialised (e.g. "ISO Country Identifier").
- If the value of the Supplementary Component is not fixed, the user will have to specify the value of the Supplementary Component each time he uses the Business Information Entity.

1648 **Context Category** – A group of one or more related values used to express one
1649 characteristic of a business circumstance.

1650 **Context Information Entity** – The influence of a particular context on the restriction
1651 on a reusable semantic building block for the exchange of business-related
1652 information.

1653 **Context Value Composition Restriction** – The influence of a particular context on
1654 the restriction on the possible values for a Supplementary Component of a Core
1655 Component Type when the corresponding Business Information Entity is used
1656 indirectly (i.e. via another Aggregate Information Entity) in an Aggregate Information
1657 Entity.

[Example]

The Business Information Entity "Financial Account.Country.Identifier" could restrict the allowed value of the "Identification.Scheme.Name" to "ISO list of country codes."

Remarks:

There are two possibilities:

If the value of the Supplementary Component is fixed the Representation Term can be specialised (e.g. "ISO Country Identifier").

If the value of the Supplementary Component is not fixed, the user will have to specify the value of the Supplementary Component each time he uses the Business Information Entity.

1658 **Core Component** – A building block for the creation of a semantically correct and
1659 meaningful information exchange ‘parcel’. It contains only the information pieces
1660 necessary to describe a specific concept. A Core Component will always be defined as
1661 a Basic Core Component, a Core Component Type, or an Aggregate Core
1662 Component.

1663 **Core Component Administrative Information** – Administrative information
1664 regarding a core component

1665 **Core Component Association Information** – Information about the association
1666 between two core components.

1667 **Core Component Change History** – History of the modifications applied to a core
1668 component version.

1669 **Core Component Replacement Information** – Information about the replacement of
1670 a core component by another.

1671 **Core Component Representation Information** – Information about the physical
1672 representation of a core component in a particular syntax.

1673 **Core Component Status Information** – History of the lifecycle of a particular
1674 version of a core component.

1675 **Core Component Type** – A core component that consists of one and only one
1676 Content Component that carries the actual content plus one or more supplementary
1677 components giving an essential extra definition to the content component. Core
1678 Component Types do not have business meaning.

1679 **Data Type** – Defines the set of valid values that can be used for a particular BCC. It

[Example]

If the content component carries “12” this has no meaning on its own. But “12
Kilometres” or “12 Euros” do have meaning.

1680 is defined by specifying restrictions on the CCT that forms the basis of the
1681 Representation Term from which the Data Type is derived.

1682 **Definition** - This is the unique semantic business meaning of a Core Component

1683 **Dictionary Entry Name** – This is the unique official name of a Core Component in
1684 the dictionary.

1685 **Information Entity** – A reusable semantic building block for the exchange of
1686 business-related information.

- 1687 **Object Class** – The logical data grouping (in a logical data model) to which a data
1688 element belongs (ISO11179). The Object Class is the part of a Core Component's
1689 Dictionary Entry Name that represents an activity or object in a specific context.
- 1690 **Primitive Type** – Primitive type used for the representation of the value of a
1691 Supplementary Component. Possible values are String, Decimal, Integer, Boolean,
1692 Date.
- 1693 **Property Term** – This identifies one of the characteristics belonging to the Object
1694 Class
- 1695 **Representation Term** – The type of valid values for a Basic Core Component.
- 1696 **Supplementary Component** – Gives meaning to the Content Component in the CCT.
- 1697 **User Community** – A user community is a group of users, with a publicised contact
1698 address, who may define context profiles relevant to their area of business. Users
1699 within the community do not create, define or manage their individual context needs
1700 but conform to the community's standard. Such a community should liaise closely
1701 with other communities and with general standards-making bodies to avoid
1702 overlapping work and to avoid creating multiple 'Towers of Babel'.

[Note]

A community may, of course, be as small as two consenting organisations!

- 1703 **Value Restriction** - Restriction on the possible values for a Supplementary
1704 Component of a Core Component Type when the corresponding Business Information
1705 Entity is based on this Core Component Type.
1706

[Example]

The Business Information Entity "Financial Account.Country.Identifier" could restrict the allowed value of the "Identification.Scheme.Name" to "ISO list of country codes."

Remarks:

There are two possibilities:

If the value of the Supplementary Component is fixed the Representation Term can be specialised (e.g. "ISO Country Identifier").

If the value of the Supplementary Component is not fixed, the user will have to specify the value of the Supplementary Component each time he uses the Business Information Entity.

1706 9 Disclaimer

1707 The views and specification expressed in this document are those of the authors and
1708 are not necessarily those of their employers. The authors and their employers
1709 specifically disclaim responsibility for any problems arising from correct or incorrect
1710 implementation or use of this design.

1711 10 Contact Information

1712 Team Leader

1713 Name Hartmut Hermes
1714 Company Siemens AG
1715 Street Richard Strauss Strasse 76
1716 City, state, zip/other 81679 Munich
1717 Nation Germany
1718
1719 Phone: (089) 92 21-4564
1720 Email: hartmut.hermes@mch11.siemens.de
1721

1722 Editor

1723 Name Mark Crawford
1724 Company Logistics Management Institute
1725 Street 2000 Corporate Ridge
1726 City, state, zip/other McLean, Virginia 22102
1727 Nation USA
1728
1729 Phone: +01 703 917 7177
1730 Email: mcrawford@lmi.org
1731

1732 **Copyright Statement**

1733

1734 Copyright © UN/CEFACT 2001. All Rights Reserved.

1735

1736 This document and translations of it MAY be copied and furnished to others, and
1737 derivative works that comment on or otherwise explain it or assist in its
1738 implementation MAY be prepared, copied, published and distributed, in whole or in
1739 part, without restriction of any kind, provided that the above copyright notice and this
1740 paragraph are included on all such copies and derivative works. However, this
1741 document itself MAY not be modified in any way, such as by removing the copyright
1742 notice or references to ebXML, UN/CEFACT, or OASIS, except as required to
1743 translate it into languages other than English.

1744

1745 The limited permissions granted above are perpetual and will not be revoked by
1746 UN/CEFACT or its successors or assigns.

1747

1748 This document and the information contained herein is provided on an "AS IS" basis
1749 and UN/CEFACT DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1750 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF
1751 THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY
1752 IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1753 PARTICULAR PURPOSE.

1754

1755