



UN/CEFACT

DRAFT

United Nations Centre for Trade Facilitation and Electronic Business

Core Components Technical Specification, Part 1

15 January 2002

Version 1.75



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 final eBTWG release for comment as
18 defined in Step 4 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 1.75 of 15 January
24 2002

25 Previous version: *Core Components* Technical specification, Version 1.7 of 31
26 October 2001

27 **2 eBTWG CC Project Team Participants**

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

30	Project Team Leader:	Hartmut Hermes	Siemens
31	Lead Editor:	Mark Crawford	Logistics Management Institute
32	Editing Team	Mike Adcock	APACS
33		Alan Stitzer	Marsh, Inc.
34		James Whittle	e Centre
35			
36	Contributors:	Mary Kay Blantz	Iona Technologies
37		Bernd Boesler	DIN
38		Kerstien Celis	Seagha c.v.
39		Marianne Cockle	APACS
40		Scott Coulthurst	State Farm
41		Alain Dechamps	CEN/ISSS
42		Arofan Gregory	CommerceOne
43		Eduardo Gutentag	Sun Microsystems
44		Paula Heilig	Worldspan
45		Stig Korsgaard	Danish Bankers Association
46		Richard May	Marsh, Inc
47		Dale McKay	Logicon/Northrop Grumman
48		Bill Murray	General Motors
49		Margaret Pemberton	Diskray
50		Sue Probert	CommerceOne
51		Andreas Schultz	DKV
52		Lisa Seaburg	CommerceOne
53		Gunther Stuhec	SAP AG
54		Hisanao Sugamata	ECOM-Japan
55		Herbert Thomas	AustriaPro
56		Frank VanDamme	SWIFT
57		Tom Warner	Boeing
58		James Werner	Boeing
59		Nigel Wooden	Accord
60		Joe Zurlo	Logistics Management Institute
61			

62	3	Table of Contents	
63	1	Status of This Document.....	2
64	2	eBTWG CC Project Team Participants	3
65	3	Table of Contents	4
66	4	Introduction.....	7
67	4.1	Scope and Focus	7
68	4.2	Structure of this Specification.....	8
69	4.2.1	Notation.....	8
70	4.3	Related Documents	9
71	4.4	Overview	10
72	4.5	Core Component Key Concepts.....	10
73	5	Working Process and Methodology.....	15
74	5.1	Overview	15
75	5.1.1	Discovery	15
76	5.1.2	How to use UN/CEFACT Core Components	16
77	5.1.2.1	Core Components and Semantic Interoperability	16
78	5.1.2.2	Steps for Use	17
79	5.2	Core Components Discovery	20
80	5.2.1	Core Component Discovery – Preparation Steps.....	20
81	5.2.2	Core Components Discovery – Search Registry/Repository	22
82	5.2.3	Core Component Discovery – Basic Business Information Entities	24
83	5.3	Preparation for Submission.....	24
84	5.3.1	Applying the Naming Convention to a New Item	25
85	5.3.2	Preparation for Submitting New Aggregate Core Components	27
86	5.3.3	Preparation Steps for Requesting a New Basic Core Component	28
87	5.3.4	Preparation for Requesting a New Aggregate Business Information Entity which re-uses an Existing Aggregate Core Component	29
89	5.4	Harmonisation.....	29
90	5.5	Technical Assessment and Approval	30
91	5.6	Context in the Discovery Process	31
92	5.6.1	Context Categories.....	32
93	5.6.2	Guidelines for Analyzing Business Information Entities in Context.....	33
94	6	Technical Details	37
95	6.1	Core Components and Business Information Entities	37
96	6.1.1	Core Components.....	37
97	6.1.2	Business Information Entities	40
98	6.1.3	Naming Convention	41
99	6.1.3.1	Dictionary Information	41
100	6.1.3.2	Rules	42
101	6.1.3.2.1	General Rules.....	42
102	6.1.3.2.2	Rules for Definitions.....	43
103	6.1.3.2.3	Rules for Dictionary Entry Names.....	43
104	6.1.3.2.4	Rules for Business Terms	46
105	6.1.3.3	List of Representation Terms	46
106	6.1.4	Catalogue of Core Components	48
107	6.1.5	Catalogue of Business Information Entities	49
108	6.2	Context.....	50
109	6.2.1	Overview of Context Specification.....	50

110	6.2.1.1	Context Categories	51
111	6.2.1.2	Constraint Language	51
112	6.2.1.3	Syntax Binding	52
113	6.2.2	Approved Context Categories	52
114	6.2.2.1	Business Process Context	53
115	6.2.2.2	Product Classification Context	53
116	6.2.2.3	Industry Classification Context	54
117	6.2.2.4	Geopolitical Context	55
118	6.2.2.5	Official Constraints Context	56
119	6.2.2.6	Business Process Role Context	56
120	6.2.2.7	Supporting Role Context	57
121	6.2.2.8	System Capabilities Context	57
122	6.2.3	Context Values	58
123	6.2.4	Core Components Context Constraints Language	58
124	6.2.4.1	Assembly Construct	64
125	6.2.4.2	ContextRule Construct	64
126	6.2.4.3	Output Constraints	65
127	6.2.4.4	Ordering and Application	65
128	7	Technical Details - Core Component Repository Storage	66
129	7.1	Storing Core Components	66
130	7.1.1	Stored Core Components	68
131	7.1.2	Stored Basic Core Components	68
132	7.1.3	Stored Core Component Types	68
133	7.1.4	Stored Aggregate Core Components	69
134	7.1.5	Stored Content Components and Supplementary Components	69
135	7.1.6	Stored Data Types	69
136	7.1.7	Stored Representation Term	69
137	7.1.8	Stored Supplementary Components	70
138	7.1.9	Stored Supplementary Component Value	70
139	7.1.10	Stored Supplementary Component Values	70
140	7.1.11	Stored Content Components	71
141	7.1.12	Stored Content Component Restrictions	71
142	7.2	Stored Context	71
143	7.2.1	Context categories	73
144	7.2.2	Context categories Scheme	73
145	7.2.3	Context Value	74
146	7.3	Stored Business Information Entities	74
147	7.3.1	Stored Aggregate Business Information Entities	76
148	7.3.2	Stored Assembly Component	76
149	7.3.3	Stored Assembly Data Type	77
150	7.3.4	Basic Business Information Entity (BBIE)	77
151	7.3.5	Basic Business Information Entity Data Type	77
152	7.3.6	Business Information Entity	77
153	7.3.7	Data Type	78
154	7.3.8	Supplementary Component Value	78
155	7.3.9	Stored Content Component Restriction	78
156	7.4	Core Component Storage Metadata	79
157	7.4.1	General Metadata Storage Rules	81
158	7.4.2	Management Information	81
159	7.4.2.1	Administrative Information	81

160	7.4.2.2	Status Information.....	81
161	7.4.2.3	Change History	82
162	7.4.2.4	Replacement Information.....	82
163	7.4.3	Content Information.....	83
164	7.4.3.1	Descriptive Information	83
165	7.4.3.2	Representation Information	83
166	7.4.3.3	Association Information.....	83
167	8	Approved Core Component Type, Content, and Supplementary	
168		Components	85
169	8.1	Approved Core Component Types	85
170	8.2	Approved Core Component Type Content and Supplementary	
171		Components	87
172	9	Definition of Terms.....	90
173	10	References.....	93
174	11	Disclaimer	95
175	12	Contact Information	96
176		Copyright Statement	97
177			

4 Introduction

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

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

4.1 Scope and Focus

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

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

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

4.2 Structure of this Specification

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

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

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

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

4.2.1 Notation

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

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

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

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

[Note] – Explanatory information. Notes are informative.

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

4.3 Related Documents

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

- ebXML Technical Architecture Specification v1.04ebXML Business Process Specification Schema v1.01
- ebXML Registry Information Model v1.0
- ebXML Registry Services Specification v1.0
- ebXML Requirements Specification v1.06
- ebXML Collaboration-Protocol Profile and Agreement Specification v1.0
- ebXML Message Service Specification v1.0ebXML Technical Reports
- Business Process and Business Information Analysis Overview v1.0Business Process Analysis Worksheets & Guidelines v1.0 -
- E-Commerce Patterns v1.0
- Catalog of Common Business Processes v1.0*Core Component* Overview v1.05
- *Core Component* Discovery and Analysis v1.04
- Context and Re-Usability of *Core Components* v1.04
- Guide to the *Core Components* Dictionary v1.04
- Naming Convention for *Core Components* v1.04
- Document Assembly and Context Rules v1.04
- Catalogue of Context Categories v1.04
- *Core Component* Dictionary v1.04
- *Core Component* Structure v1.04
- Information Technology — Metadata registries: Framework for the Specification and Standardization of Data Elements, International Standardization Organization, ISO 11179-1
- Information Technology — Metadata registries: Classification of Concepts for the Identification of Domains, International Standardization Organization, ISO 11179-2
- Information Technology — Metadata registries: Registry Metamodel, International Standardization Organization, ISO 11179-3
- Information Technology — Metadata registries: Rules and Guidelines for the Formulation of Data Definitions, International Standardization Organization, ISO 11179-4

- 280 — Information Technology — Metadata registries: Naming and Identification
281 Principles for Data Elements, International Standardization Organization, ISO
282 11179-5
- 283 — Information Technology — Metadata registries: Framework for the
284 Specification and Standardization of Data Elements, International
285 Standardization Organization, ISO 11179-6

286 **4.4 Overview**

287 This *Core Components Technical Specification* provides a way to identify, capture
288 and maximize the reuse of business information to support and enhance information
289 interoperability across multiple business situations. The specification focuses both on
290 human-readable and machine-processible representations of this information.

291 The *Core Components* approach described in this document is more flexible than
292 current standards in this area because the semantic standardisation is done in a syntax-
293 neutral fashion. UN/CEFACT can guarantee that two trading partners using different
294 syntaxes (e.g. XML and EDIFACT) are using business semantics in the same way on
295 condition that both syntaxes have been based on the same *Core Components*. This
296 enables clean mapping between disparate message definitions across syntaxes,
297 industry and regional boundaries.

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

305 **4.5 Core Component Key Concepts**

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

- 307 • *Core Component* — The *Core Component* is a semantic building block
308 that is used as a basis to construct all electronic business messages. A
309 technical specification for creating a *Core Component* library is provided.

310 [Definition] *Core Component* (CC)

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

- 314 • *Business Context* – *Business Context* is a mechanism for qualifying and
315 refining *Core Components* according to their use within a particular

316 process. Once business contexts are identified, the appropriate *Core*
 317 *Components* can be selected or created and differentiated to indicate any
 318 necessary qualification and refinement needed to support the business
 319 process in a given business context.

320 [Definition] *Business Context*

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

324 • *Business Information Entity* –When a *Core Component* is used in a real
 325 business situation it is used to define a *Business Information Entity*. The
 326 *Business Information Entity* is the result of using a *Core Component*
 327 within a specific business context.

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

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

332 There are three different categories of *Core Components*: *Basic Core Component*,
 333 *Core Component Type* and *Aggregate Core Component*. The following definitions
 334 explain each of these:

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

336 A *Core Component* that represents a singular business concept with a unique business
 337 semantic definition. A *Basic Core Component* is constructed by using a *Core*
 338 *Component Type*. *Basic Core Components* are used in developing *Aggregate Core*
 339 *Components*.

340

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

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

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

348

[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.

[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

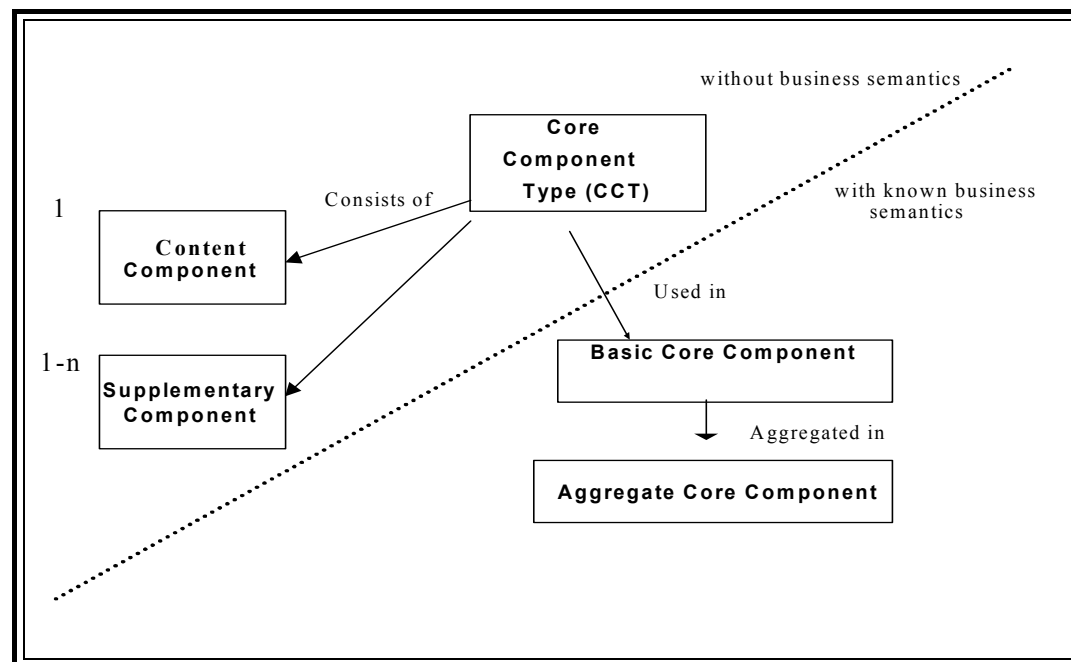
- ◆ at least one *Basic Core Component* plus one or more *Aggregate Core Components*

[Example] - *Aggregate Core Components*

Account. Details; Party. Details

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

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



366
 367 A specific relationship exists between *Core Components* and *Business Information*
 368 *Entities*. Core and business elements are complementary in many respects. *Core*
 369 *Components* are intended to be the linchpin for creating interoperable business
 370 process models and business documents using a controlled vocabulary.

371 [Definition] *Basic Business Information Entity*

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

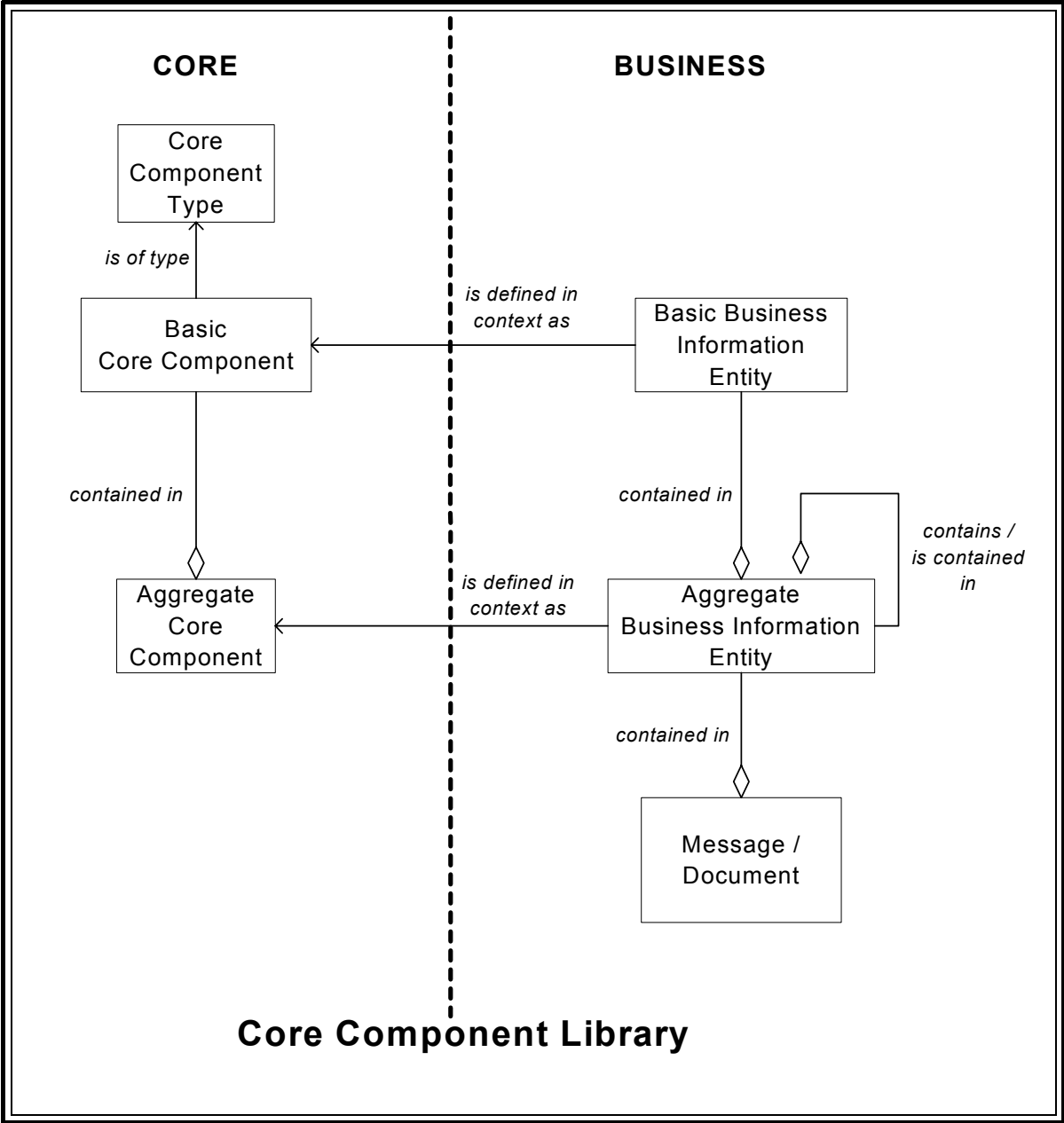
374

375 [Definition] *Aggregate Business Information Entity*

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

378 The features of the relationship between *Core Components* and business information
 379 entities are described in Figure 4-2.

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



[Note]

From previous words it should be clear that the term *Core Component* is used as a generic term that encompasses *Basic Core Components* and *Aggregate Core Components*. Equally the term *Business Information Entity* is used as a generic term encompassing *Basic Business Information Entities* and *Aggregate Business Information Entities*.

5 Working Process and Methodology

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

5.1 Overview

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

5.1.1 Discovery

A business process should be modeled using a standard approach. UN/CEFACT requires the *UN/EDIFACT Modeling Methodology* (UMM) as the approach.³ One of the results is a model (e.g. a class diagram) that shows the business information and its inter-relationships. *Business Information Entities* can be identified from the class diagram.

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

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

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

³ 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* that uses the generic *Amount*, which is a *Basic Core Component*. 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.

[Issue]

The preceding example may not adequately convey the intended information. Based on comments received during the final eBTWG review cycle, it may be reworked.

Just as each *Business Information Entity* must ultimately be based on *Basic Core Components*, each *Aggregate Business Information Entity* must ultimately be based on an existing *Aggregate Core Component*. The underlying *Aggregate Core Component* identifies the generic, standard definition of business information that is being used in the *Aggregate Business Information Entity*. The *Aggregate Business Information Entity* inherits the generic description, which is then modified and enhanced to be specific to the business process in which the *Aggregate Business Information Entity* is used. An *Aggregate Business Information Entity* is thus directly tied to a specific business process, or to a *Business Context*. (See Section 5.7 for a fuller understanding of context.)

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

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

5.1.2 How to use UN/CEFACT Core Components

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

5.1.2.1 Core Components and Semantic Interoperability

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

456 (MIG), an XML Schema, or similar syntax specific representation. UN/CEFACT will
457 produce standards based representations of these artefacts for implementation.

458 Under the *Core Components* concept, defining and storing *Core Components* and
459 associated context mechanisms occur prior to the creation of a MIG or a Schema. In
460 this manner, the focus of the user changes from examining the MIG or DTD, and
461 moves to an examination of the semantic models. Accordingly, interoperability
462 between syntaxes no longer depends on analysing specific instances, but naturally
463 occurs during the business process model definition phase.

464 5.1.2.2 Steps for Use

465 The overall discovery and document creation process can be thought of as a series of
466 steps that starts with determining the availability of existing business process
467 definitions and ultimately results in standard business documents. Figure 5-1
468 illustrates this process. Specific steps to be followed are further described below.
469

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

473 • If no existing business process is found to be appropriate, then the new
474 business process should be modelled and submitted to the registry. The
475 process includes conducting a thorough analysis of the business
476 information requirements by following the *Core Component Discovery*
477 *Steps* (Section 5.2.2).

478 • If an existing business process is located that will be used, the new use
479 should be identified to the registry. If the searcher does not have access
480 to the registry, the *Catalogue of Common Business Processes* (CCBP)
481 can be substituted. The searcher continues with Step 2.

482 Step 2: Identify relevant context categories - Access the registry interface and
483 identify the relevant context categories of the selected business process by
484 determining the following:

485 • *Product Classification Context* – Determine the goods or services
486 concerned in the collaboration.

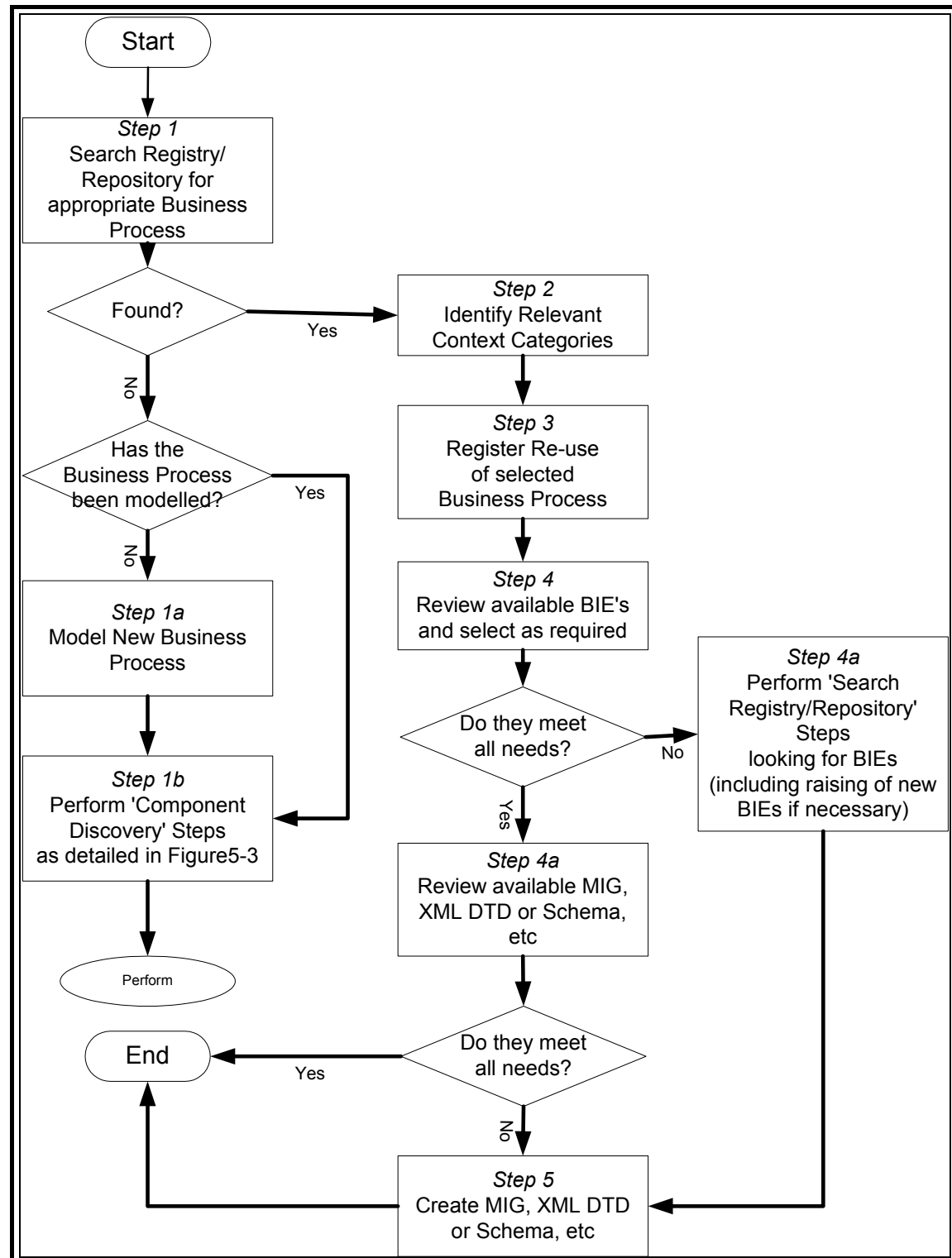
487 • *Industry Classification Context* – Determine the relevant trading partner
488 industries.

489 • *Geopolitical Context* – Determine where the business process is to be
490 conducted. Determine if the business process crosses international
491 boundaries.

492 • *Official Constraints Context* – Determine any legal restrictions or
493 requirements on this business process.

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

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



- 498 [Issue]
- 499 The next edition of this document may have a modified version of the above diagram
- 500 and the step descriptions that follow, to allow harmonisation and seamless knitting
- 501 together of the *Business Process* and *Core Component* specifications.
- 502 • *Supporting Role Context* – Determine what other significant parties will be
 - 503 using the data in the messages. Determine their role in the overall process.
 - 504 • *System Capabilities Context* – Determine any major restrictions derived
 - 505 from legacy systems. Identify the type of system.
- 506 The registry will provide a list of pre-defined *Business Information Entities*
- 507 that are available to the selected business process, and which meet the
- 508 context criteria specified. These will come with links to the *Core*
- 509 *Components* that they are based on and the constraint rules/values that fully
- 510 qualify them. The registry should also return partial matches with an
- 511 indication of how closely they match the specified context.
- 512 Step 3: Register re-use of selected *Business Process* in the set of contexts in which it
- 513 is being used. Registration of each re-use ensures the gradual development
- 514 of a library of re-uses that will be available to the widening user base.
- 515 Step 4: Review the available *Business Information Entities* and select the
- 516 appropriate subset for use that meets the needs of the business process
- 517 requirement that is being developed.
- 518 If the *Business Information Entities* available for the specific
- 519 business process do not address all of the data requirements, the
- 520 repository of all *Business Information Entities* should be searched to
- 521 see if the appropriate *Business Information Entities* already exist. The
- 522 procedure for this is described under Search Repository (Section 5.2)
- 523 which includes the steps to raise any new *Business Information*
- 524 *Entities* required because no appropriate *Business Information*
- 525 *Entities* can be found.
- 526 Step 4a: If all required *Business Information Entities* are already available, review the
- 527 available MIG, XML Document Type Definition or Schema, etc and select
- 528 the appropriate one(s) for use that meet the technical implementation/
- 529 solution requirements identified.
- 530 If no appropriate technical implementation/solution is already
- 531 available, continue with Step 5 to create new ones.
- 532 Step 5: Create MIG, XML DTD or Schema, etc. – The resulting semantic model
- 533 (the set of *Business Information Entities*) is manually or programmatically
- 534 rendered into a syntax-specific message description. The resulting MIG,
- 535 DTD or Schema is submitted to the repository where it is associated with the
- 536 *Business Information Entities* it represents.

[Note]

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

5.2 Core Components Discovery

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

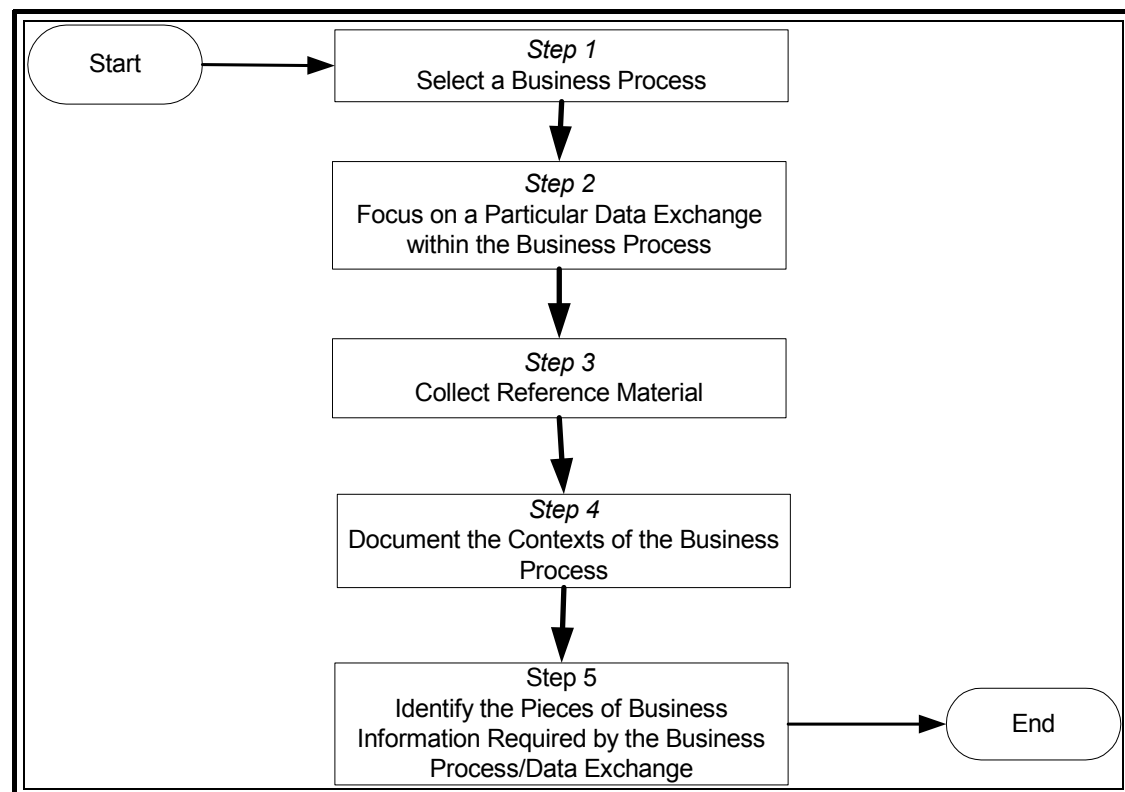
5.2.1 Core Component Discovery – Preparation Steps

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

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

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

582 **Figure 5-2 Preparation Steps**



583
584

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

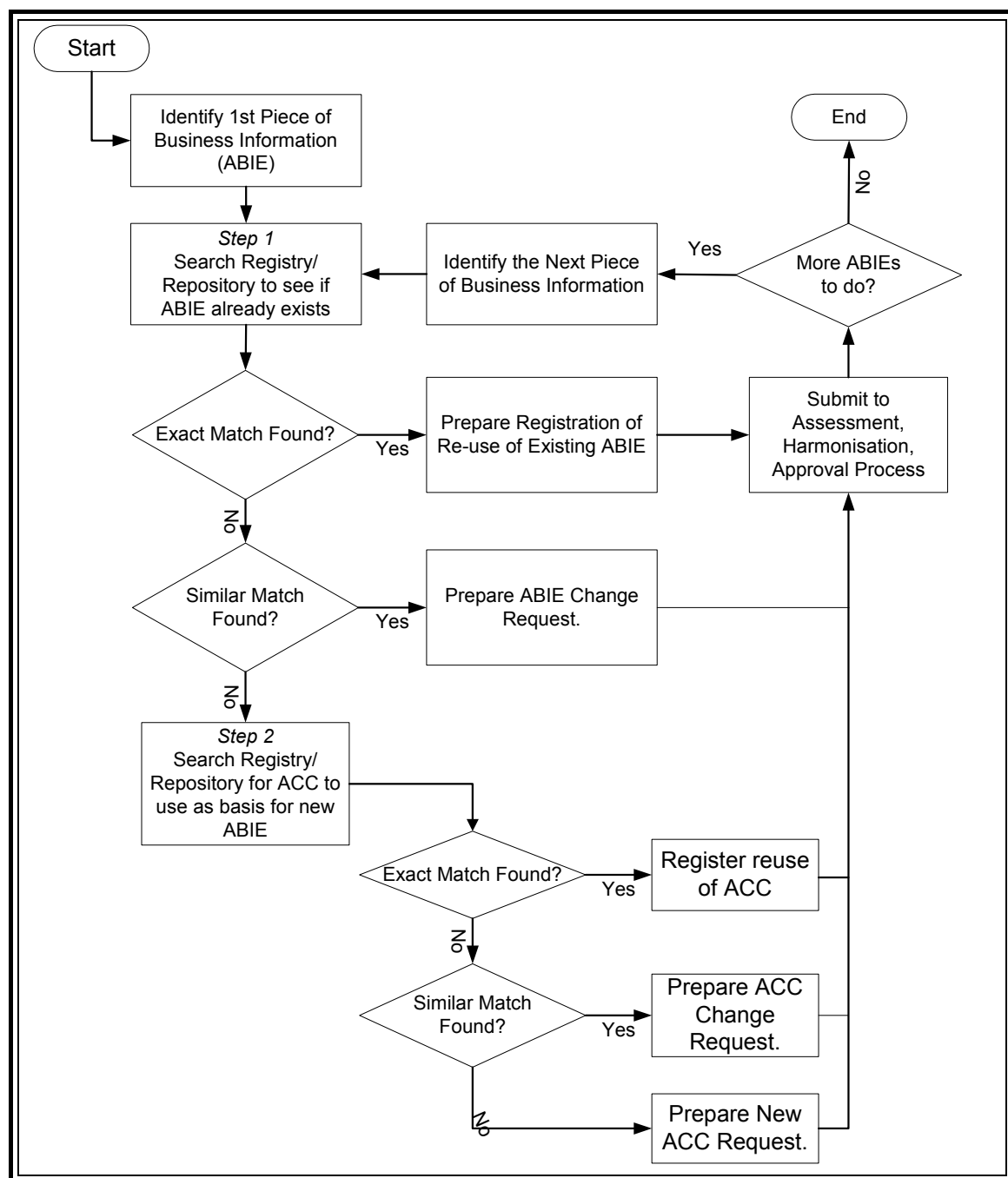
- 586 • If starting from a model (UN/CEFACT recommends UMM models of
- 587 business processes), identify the objects (*Aggregate Business Information*
- 588 *Entities*) that are needed.
- 589 • If not starting from a model, collect the pieces of information into object-
- 590 like groups (*Aggregate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

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

5.2.2 Core Components Discovery – Search Registry/Repository

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

Figure 5-3 Search Steps



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

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

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

618 [Note]

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

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

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

629 • If there is an existing *Aggregate Core Component* with a definition and
 630 structure that meets the business needs³, register the re-use of the
 631 *Aggregate Core component* as an *Aggregate Business Information Entity*
 632 including the business context and any business terms. (Go to next
 633 *Aggregate Business Information Entity*)

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

- If there is not an *Aggregate Core Component* with a suitable definition and structure, prepare a new *Aggregate Core Component* request for submission to the harmonisation and approval process. Include the re-use of the *Aggregate Core Component* as an *Aggregate Business Information Entity*, the business context and any business terms. (Go to next *Aggregate Business Information Entity*)

5.2.3 Core Component Discovery – Basic Business Information Entities

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

5.3 Preparation for Submission

Following the search of the *Core Component Library*, there may be a need to prepare submissions for the harmonisation and approval process.

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

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

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

- To register a Re-use of an Existing *Aggregate Business Information Entity*
- To make a Change Request for an *Existing Aggregate Business Information Entity*
- To make a Change Request for an Existing *Aggregate Core Component*

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

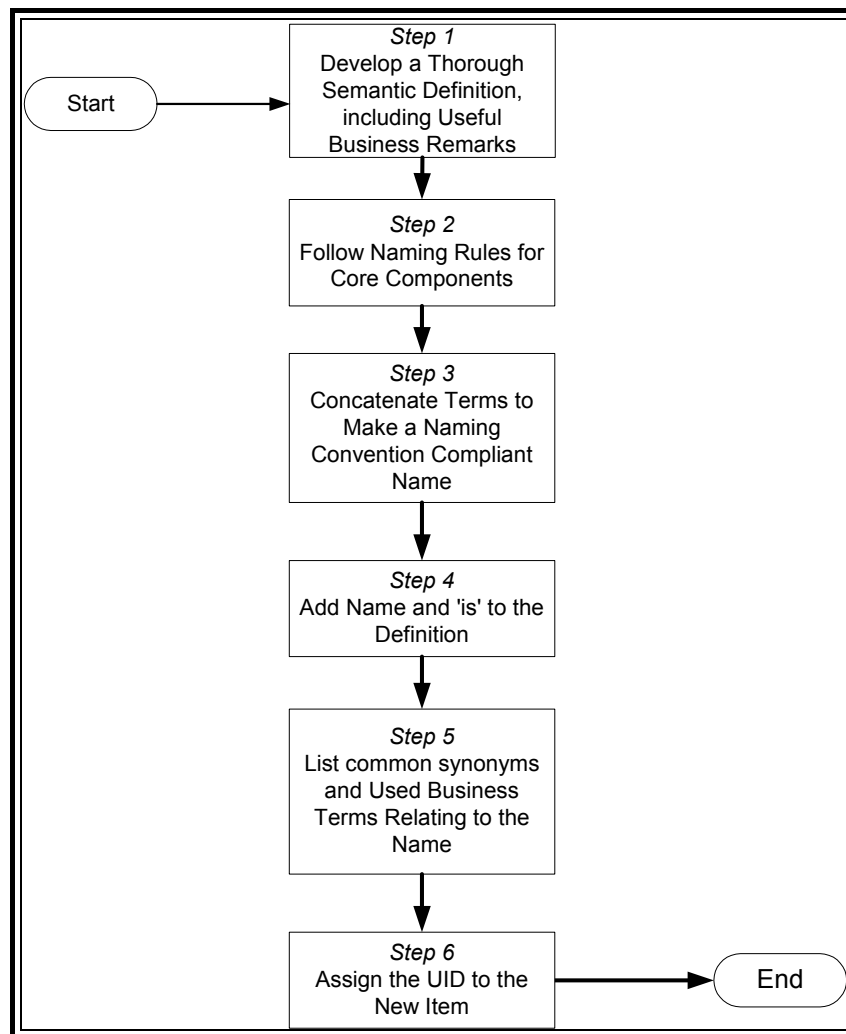
- Preparation for Requesting a new *Aggregate Core Component*
- Preparation for Requesting a new *Core Component*
- Preparation for Requesting a new *Aggregate Business Information Entity* which re-uses an Existing *Aggregate Core Component*

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

5.3.1 Applying the Naming Convention to a New Item

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

Figure 5-4 Applying the Naming Convention



680

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

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

- 688 • be applicable across multiple industries or domains, and
- 689 • be simple and clear to enable unambiguous translation to other languages
- 690 Step 2. Follow the *Naming Rules for Core Components* (Section 6.1.3) to assign:
- 691 • *Representation Type*
- 692 • *Property Term*
- 693 • *Object Class*
- 694 Step 3. Concatenate the terms to create a *Naming Convention* compliant name.

695 [Note]

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

- 699 Step 4. Check the quality of the definition by adding the words “[*Dictionary Name*]
700 is” to the front of the definition, where [*Dictionary Name*] is the agreed
701 name.

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

705 [Note]

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

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

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

Temp UID	Definition	Remarks	Business Terms	CCT	Dictionary Entry Name			
					Name	Object Class	Property Term *to be suppressed according to rule 5	Represent- ation Term
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	

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

Same Business Term

5.3.2 Preparation for Submitting New Aggregate Core Components

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

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

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

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

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

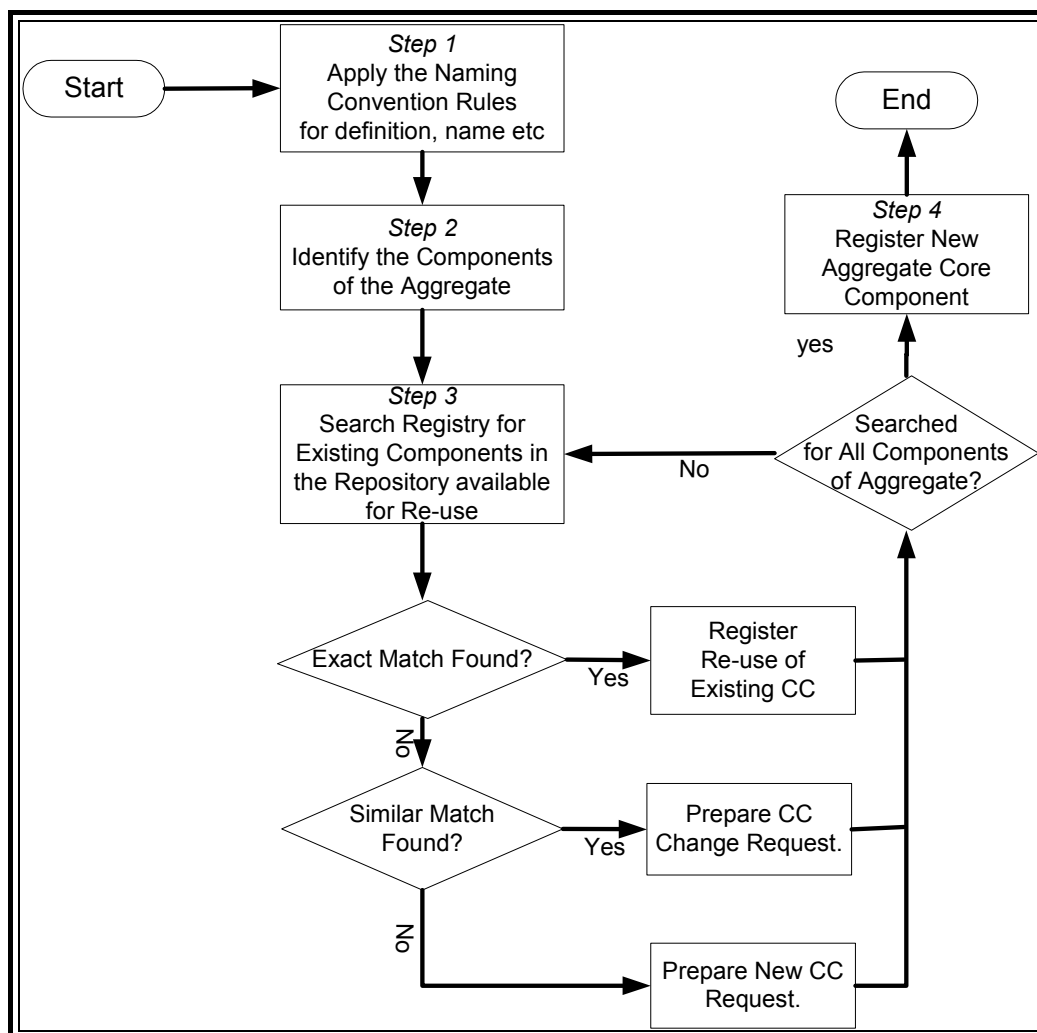
- If there is an existing *Core Component* with a definition and structure that meets the requirement, register this re-use of the *Core Component* including the context in which it is used.
- If there is an existing *Core Component* with a definition and structure that potentially could be modified to meet the requirement, prepare an *Core Component* change request for submission to the harmonisation and approval process, including the re-use of the *Core Component* and the context in which it is used.
- If there is not an existing *Core Component* with a suitable definition and structure, prepare a new *Core Component* request for submission to the harmonisation and approval process, including the re-use of the *Core Component* and the context.

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

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

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

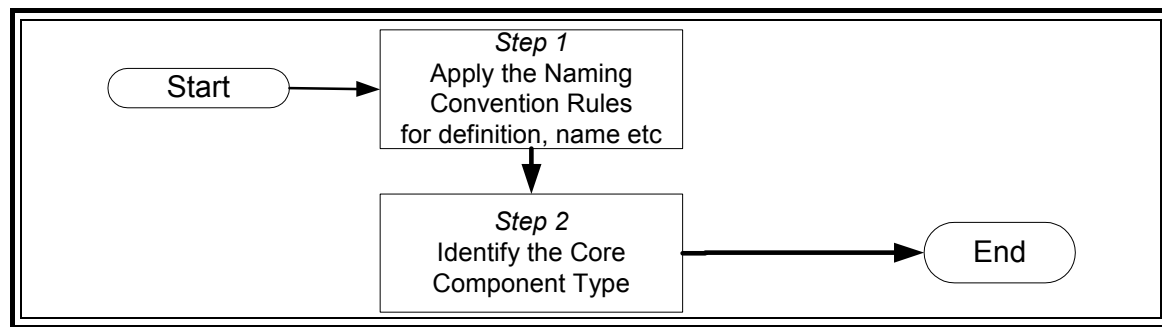
744 **Figure 5-6 Preparation for requesting a new Aggregate Core Component**



746 5.3.3 Preparation Steps for Requesting a New Basic Core Component

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

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



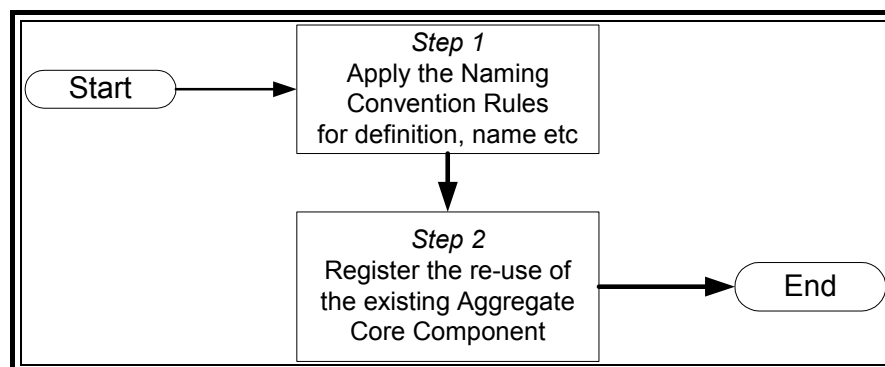
751 Step 1. Apply the *Naming Convention* and *Rules* to arrive at the name of the new
752 *Basic Core Component*

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

755 **5.3.4 Preparation for Requesting a New Aggregate Business** 756 **Information Entity which re-uses an Existing Aggregate Core** 757 **Component**

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

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



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

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

767 **5.4 Harmonisation**

768 The purpose of harmonisation is to take a set of proposed *Core Components* or
769 *Business Information Entities* from different domains, identify differences and

770 similarities between the various submissions, and produce a single, complete cross-
771 domain set of *Core Components*. Harmonisation is a critical step in the overall *Core*
772 *Component* procedures. The following describes the recommended areas that
773 harmonisation procedures should cover.

- 774 1. Evaluate each submitted *Core Component* for consistent application of the
775 Discovery methodology. Resolve any questions or issues by discussion with the
776 submitting groups.
- 777 2. Compare the definition and structure of each submitted *Core Component* with
778 what already exists in the *Core Component Library*.
 - 779 • If the submitted *Core Component* is the same or similar, compare the
780 properties of each to identify any differences. If the submitted *Core*
781 *Component* has properties missing in the existing one, enforce a
782 harmonised form that contains the properties of each. If the submitted
783 *Core Component* is a subset of the existing *Core Component* definition,
784 then recommend the use of the existing one.
 - 785 • If the definition of the *Core Component* does not match any existing ones,
786 then go to 3.
- 787 3. Publish the results of harmonisation to the submitting groups for review and
788 finalisation.

789 [Note]

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

795 Once the submitted material has passed the harmonisation procedure, it may now be
796 submitted for assessment and approval.

797 **5.5 Technical Assessment and Approval**

798 Technical assessment must be done in close coordination with the discovery teams
799 and the harmonisation process in order to minimise domain re-working after
800 *Technical Assessment* and *Harmonisation* review. This section, 5.5, defines a
801 recommended process for conducting technical assessment and approval of all newly
802 submitted and changed *Core Components*.

803 Technical assessment procedures define the processing that shall be followed by the
804 joint development groups, the *Harmonisation* group, submission entry points, the
805 *Technical Assessment* group, and the secretariat as related to the review of *Core*

806 *Components*. The result of this process is the final publication of approved *Core*
807 *Components*.

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

814 In outline, the procedures cover:

- 815 1) Submission of *Core Component* work that is ready to be reviewed to a designated
816 secretariat.
- 817 2) Recording of all *Core Component* submissions and distribution to the
818 Harmonisation Group members.
- 819 3) Review procedures and criteria followed by the *Harmonisation* group.
- 820 4) Return of harmonised *Core Component* submissions for *Technical Assessment*.
- 821 5) Review procedures and criteria followed by the *Technical Assessment* group.
- 822 6) Registration of the approved *Core Component(s)* in the appropriate *Core*
823 *Component Registry*.

824 [Issue]

825 The *UN/CEFACT Technical Review and Approval Procedures* need to be developed
826 and published as soon as possible by the appropriate working group.

827 **5.6 Context in the Discovery Process**

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

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

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

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

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

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

855 [Issue]

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

862 **5.6.1 Context Categories**

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

- 865 • *Business Process Context*: This is the classification of the business
 866 process, business collaboration, or business transaction as described in the
 867 *Catalogue of Common Business Processes*. It is the primary context
 868 category, and provides many useful distinctions in the analysis of *Core*
 869 *Components*.
- 870 • *Product Classification Context*: There are many types of information that
 871 are specific to products or services being traded or referred to in a business
 872 process.
- 873 • *Industry Classification Context*: Traditionally, business vocabularies are
 874 divided up into industry verticals. This *Context Category* specifies a
 875 particular industry vertical.
- 876 • *Geopolitical Context*: Specifies the semantic and structural variation. This
 877 is often the result of regional or cultural factors.

- 878 • *Official Constraints Context*: Specifies the legal or contractual influences
879 upon business semantics.
- 880 • *Business Process Role Context*: Every partner in a business process data
881 exchange has a particular role – buyer, seller, etc. These roles are
882 described in the *Catalogue of Common Business Processes* and in other
883 *Business Libraries* (libraries of business process models). Depending on
884 the business process, the nature of these roles may require that certain
885 semantics and data be employed in the messages exchanged. In any
886 *Business Process Role Context*, one must either be a sender or receiver of
887 data in that particular exchange – otherwise, role is described by the
888 *Supporting Role Context*.
- 889 • *Supporting Role Context*: Parties in a business process who are neither
890 senders nor receivers of data in a particular exchange, may place
891 requirements on the data exchanged by partners who are sending or
892 receiving of data in that exchange. These non-sending, non-receiving
893 parties in this exchange play a supporting role, and are described by the
894 *Supporting Role Context*.

895 [Issue]

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

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

902 5.6.2 Guidelines for Analyzing Business Information Entities in Context

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

914 It is possible that a particular property of a *Business Information Entity* may be the
915 result of several context factors. These context factors are identified by analysis of
916 differences and similarities across particular contexts. For example, comparing the
917 same *Business Information Entity* as used in different regions of the world, variation

918 will probably be the result of a geopolitical context or official constraints context (see
919 below). If a single *Business Information Entity* differs between business processes,
920 then the business process context is probably the cause. For each non-core property of
921 every *Business Information Entity* analysed the relevant influences and hence context
922 factors should be identified.

923 The following guidelines apply:

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

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

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

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

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

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

943 4) *System Capability Context* categories

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

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

951 [Example]

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

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

958 *Business Process = In All Contexts*

959 *Product Classification = In All Context*

960 *Industry Classification = In All Contexts*

961 *Geopolitical = United States*

962 *Official Constraint = None*

963 *Business Process Role = In All Contexts*

964 *Supporting Role = In All Contexts*

965 *System Capabilities = In All Contexts*

966 These values were selected based on the following analysis:

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

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

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

979

980 [Example] (Continued)

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

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

988 The same reasoning holds for the *Supporting Role Context*.

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

995
996 If, in the above example, the address was taken from a French standard, it might be
997 that some child elements are common across a number of countries in the same
998 region, and perhaps even in multiple regions. Providing the value *France* as a
999 *Geopolitical Context* here would be incomplete – every known valid value should be
1000 given.

⁴ The UN/CEFACT Modeling Methodology (UMM) is a methodology for business process and information modeling that is based on the Unified Modeling Language.

1001 **6 Technical Details**

1002 This section provides a detailed technical explanation of the *Core Component*,
1003 *Business Process* integration, storage and metamodel elements of the UN/CEFACT
1004 *Core Components* concept.

1005 **6.1 Core Components and Business Information Entities**

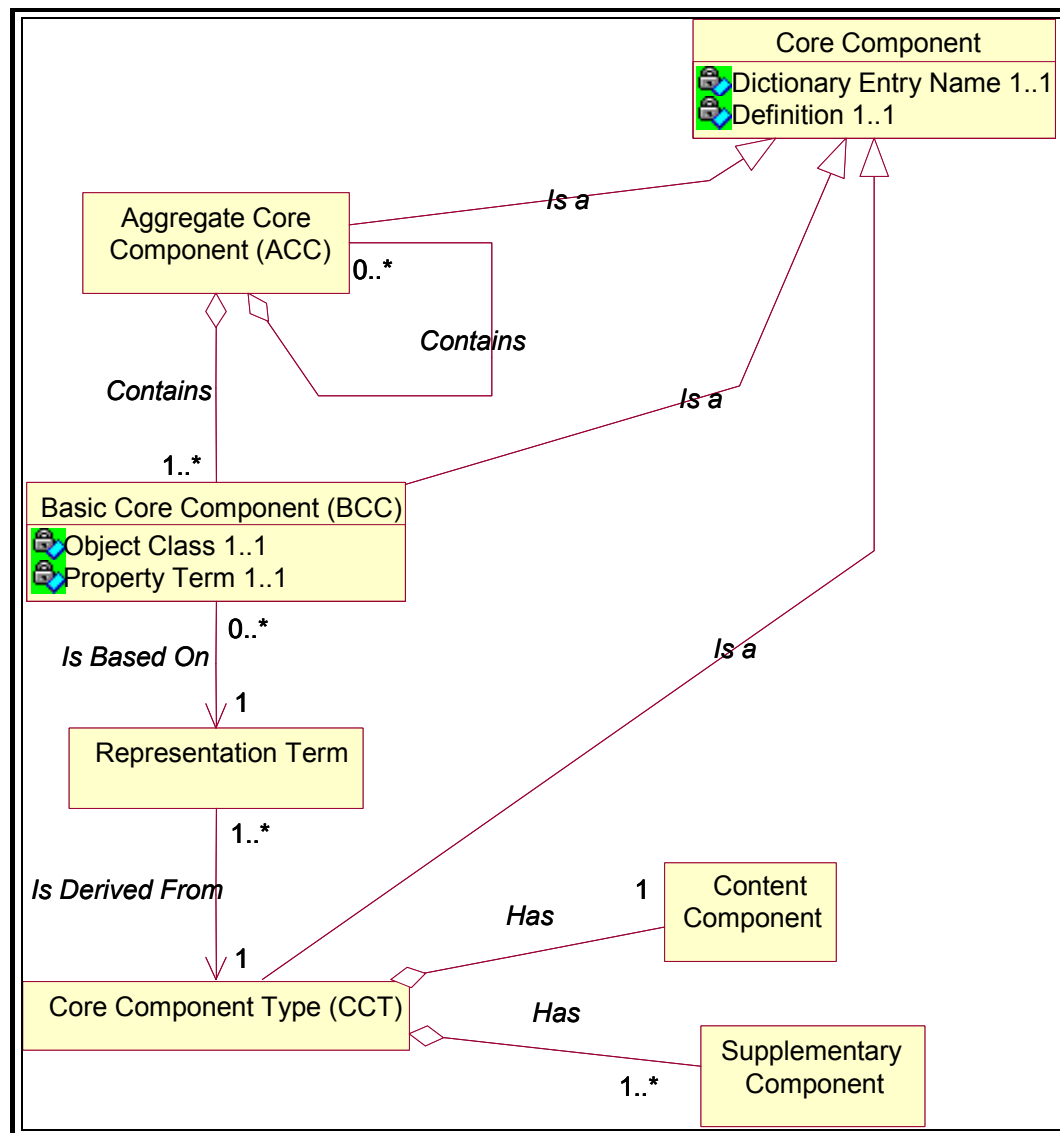
1006 This section defines the following:

- 1007 • *Core Component* rules
- 1008 • *Naming Conventions*,
- 1009 • allowable *Core Component Types*
- 1010 • *Content* and *Supplementary Component* types, and
- 1011 • *Representation Terms*.

1012 This section also specifies relationships for *Core Components* and *Business*
1013 *Information Entities* and includes details required for constructing the *Core*
1014 *Components Catalogue* and a larger *Core Component Library*.

1015 **6.1.1 Core Components**

1016 A *Core Component* is a building block for the creation of a semantically correct and
1017 meaningful business information exchange ‘parcel’, containing the information pieces
1018 needed to describe a specific concept. There are three categories of *Core Components*:
1019 *Basic Core Component*, *Core Component Type* and *Aggregate Core Component*.
1020 Figure 6-1 illustrates these three categories and their relationships.

1021 **Figure 6-1. Core Components Basic Definitions**

1022 The following general rules must be followed in discovering and documenting the
 1023 three types of *Core Components*:

1024 [C1] Each *Core Component Type*, *Basic* or *Aggregate Information Entity* must have
 1025 its own business semantic definition. The definition shall be developed first
 1026 and the *Dictionary Entry Name* shall be extracted from it. Remarks can be
 1027 used to further clarify the definition, to provide examples and/or to reference a
 1028 recognised standard.

1029 [C2] Within an *Aggregate Core Component*, all embedded entities shall be related
 1030 to the concept of the aggregate.

1031 [C3] There shall be no semantic overlap between the *Core Components* embedded
1032 within the same *Aggregate Core Component*.

1033 [C4] The representation of the information in a *Core Component* of the *Core*
1034 *Component Type Code* should use a standard issued by a recognised standards
1035 body, whenever a standard exists. If international standards are not used a
1036 business driven justification shall be provided.

1037 [C5] An *Aggregate Information Entity* must contain at least one *Basic Core*
1038 *Component*.

1039 [Issue]

1040 The issue of allowing aggregates without at least one *Basic Core Component* was
1041 discussed in detail and the majority opinion is that problems with infinite loops and
1042 miscast aggregates would arise without the restriction to have at least one *Basic Core*
1043 *Component* in each aggregate. There is a minority opinion that we may discover there
1044 are valid technical reasons where occurrences of *Aggregate Core Components* may
1045 only be comprised of other aggregates. A final position on this issue will be developed
1046 based on the comments received during the final eBTWG review period.

1047

1048 [Note]

1049 For the purpose of exchanging information a practical compromise on the level of
1050 detail of a *Basic Core Component* is required. This compromise shall be based on the
1051 Business Need. It is not necessary to have absolute detail, which decomposes a piece
1052 of information down to its lowest level.

1053 [C6] The *Core Component Type* shall be one of the approved *Core Component*
1054 *Types*

1055 [Note]

1056 Table 8-1 provides a complete list of the currently approved *Core Component Types*.
1057 This Table will subsequently be published separately to facilitate maintenance outside
1058 the body of this specification.

1060 [C7] The *Content Component* shall be one of the approved *Content Components*

1061 [C8] The *Supplementary Component* shall be one of the approved *Supplementary*
1062 *Components*

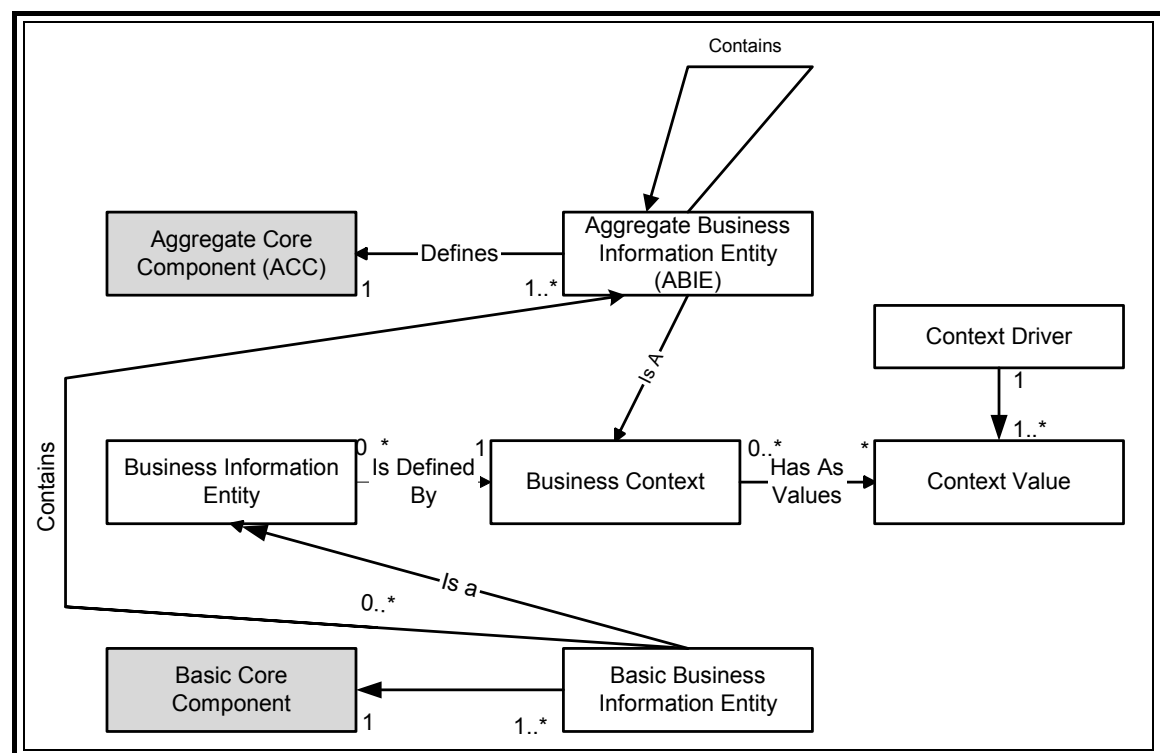
[Note]

Table 8-3 provides a complete list of the currently approved *Content Components* and *Supplementary Components*. This Table will subsequently be published separately to facilitate maintenance outside the body of this specification.

6.1.2 Business Information Entities

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

Figure 6-2. Business Information Entities Basic Definition Model



[B1] A *Business Information Entity* shall always be either a *Basic Business Information Entity* or an *Aggregate Business Information Entity*.

[B2] All *Basic Business Information Entities* that relate to the same context-free concept shall form the basis of the definition of a *Basic Core component*.

1085 [B3] All *Aggregate Business Information Entities* that relate to the same context-
 1086 free concept form the basis of the definition of an *Aggregate Core component*.

1087 [B4] An *Aggregate Business Information Entity* will consist of two or more *Basic*
 1088 *Business Information Entities* and/or *Aggregate Business Information Entities*.

1089 **6.1.3 Naming Convention**

1090 A naming convention is necessary to gain consistency in the naming and defining of
 1091 all *Core Components* and *Business Information Entities*. The resulting consistency
 1092 facilitates comparison during the discovery and analysis process, and precludes
 1093 ambiguity, such as the creation of multiple *Core Components* with different names
 1094 that have the same semantic meaning.

1095 The naming rules are derived from the guidelines and principles described in
 1096 document ISO 11179 Part 5 -- *Naming and Identification Principles For Data*
 1097 *Elements*. In certain instances, these guidelines have been adapted to the *Core*
 1098 *Component* environment. In particular, the guidelines have been extended to cover the
 1099 naming and defining of *Aggregate Business Information Entities* and *Core*
 1100 *Component Types*.

1101 In order to ensure absolute clarity and understanding of the names and definitions it is
 1102 essential to use words from the *Oxford English Dictionary*. A supplementary
 1103 *Controlled Vocabulary* will be developed to define uniquely any words that are
 1104 potentially ambiguous. This will ensure that each word within any of the names and
 1105 definitions is used in a consistent and unambiguous way. The resultant semantic
 1106 integrity will also mean that translation into other languages retains the precise
 1107 original meaning.

1108 [Issue]

1109 The *Naming Convention* needs to be examined to see if the rules adequately cover the
 1110 naming of *Basic Business Information Entities* and *Aggregate Business Information*
 1111 *Entities*, particularly in combining context and core naming. This will be resolved by
 1112 the *Core Component* Team based on inputs received during the final eBTWG review
 1113 process.

1114 **6.1.3.1 Dictionary Information**

1115 Each *Core Component* contains the following dictionary information that is impacted
 1116 by the naming rules:

1117 • **Dictionary Entry Name** (Mandatory). This is the unique official name of
 1118 the *Core Component* in the dictionary.

1119 • **Definition** (Mandatory). This is the unique semantic business meaning of
 1120 that *Core Component*.

- 1121 • **Business Term** (Optional). This is a synonym term under which the *Core*
- 1122 *Component* is commonly known and used in the business. A *Core*
- 1123 *Component* may have several business terms or synonyms.

1124 [Example]

1125 *Dictionary Entry Name* (e.g. *Charge. Price. Amount; Currency Exchange. Rate;*
 1126 *Financial. Account. Identifier*)

1127 *Business Term* (e.g. *Price Amount; Conversion Rate; Account Number*)

1128

1129 [Example]

1130 *Dictionary Entry Name – Person. Tax. Identifier*

1131 Definition – The registered national tax identification of a person

1132 *Business Term – Social security number, national insurance number*

1133 The naming rules are also based on the following concepts as defined in ISO 11179:

- 1134 • **Object Class**. This represents the logical data grouping or aggregation (in a
- 1135 logical data model) to which a data element belongs. The *Object Class*
- 1136 thus is the part of a *Core Component's Dictionary Entry Name* that
- 1137 represents an activity or object in a specific context.

- 1138 • **Property Term**. This represents the distinguishing characteristic or
- 1139 property of the dominant area of interest and shall occur naturally in the
- 1140 definition.

- 1141 • **Representation Term**. This defines the type of valid values for an
- 1142 information entity.

1143 6.1.3.2 Rules

1144 The following subsections define all naming convention rules.

1145 6.1.3.2.1 General Rules

- 1146 [C9] The dictionary content shall be in *English Language* following the
- 1147 primary *Oxford Dictionary* English spellings to assure unambiguous
- 1148 spelling.

1149 [Note]

1150 There may be restrictions in specific languages, which need to be applied when
1151 transforming the *Core Component* dictionary into other languages. These restrictions
1152 shall be formulated as additional rules and added as separate language specific
1153 annexes to this document.

1154 6.1.3.2.2 Rules for Definitions

1155 [C10] The definition shall be consistent with the requirements of ISO 11179-4
1156 Section 4.4 and will provide an understandable meaning, which should also be
1157 translatable to other languages.

1158 [C11] The definition shall take into account the fact that the users of the *Core*
1159 *Component Dictionary* are not necessarily native English speakers. It shall
1160 therefore contain short sentences, using normal words. Wherever synonym
1161 terms are possible, the definition shall use the preferred term as identified in
1162 the *Core Components Glossary of Terms*.

1163 [C12] The definition of a *Core Component* shall use a structure that is based on the
1164 existence of the *Object Class*, the *Property Term*, and its *Representation Term*.

1165 [C13] Whenever both the definite (i.e. *the*) and indefinite article (i.e. *a*) are
1166 possible in a definition, preference shall be given to the indefinite article (i.e.
1167 *a*).

1168 [Note]

1169 To check the quality of the definition, place the *Dictionary Entry Name* followed by *is*
1170 before the definition to ensure that it is not simply a repetition of the *Dictionary Entry*
1171 *Name*.

1172 6.1.3.2.3 Rules for Dictionary Entry Names

1173 [C14] The *Dictionary Entry Name* shall be unique.

1174 [C15] The *Dictionary Entry Name* shall be extracted from the *Core Component*
1175 definition.

1176 [C16] The *Dictionary Entry Name* shall be concise and shall not contain consecutive
1177 redundant words.

1178 [C17] The *Dictionary Entry Name* and all its components shall be in singular form
1179 unless the concept itself is plural.

1180 [Example]

1181 *Goods*

1182 [C18] The *Dictionary Entry Name* shall not use Non-letter characters unless required
1183 by language rules.

1184 [C19] The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e.
1185 no words like *and, of, the*, etc.). This rule shall be applied to the English
1186 language, and may be applied to other languages as appropriate.

1187 [C20] Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall
1188 be expanded or explained in the definition.

1189 [C21] The *Dictionary Entry Name* of a *Core Component* shall consist of the name of
1190 an *Object Class*, the name of a *Property Term* and the name of a
1191 *Representation Term*.

1192 [Example]

1193 *Tax. Description. Text*

1194 [C22] The components of a *Dictionary Entry Name* shall be separated by dots. The
1195 space character shall separate words in multi-word *Object Classes* and/or
1196 multiword *Property Terms*. Every word shall start with a capital letter. To
1197 allow spell checking of the *Dictionary Entry Names'* words, the dots after
1198 *Object Class* and *Property Terms* shall be followed by a space character.

1199 [Note]

1200 The use of CamelCase for *Dictionary Entry Names* has been considered, but has been
1201 rejected for following reasons:

1202 ♦ Use of CamelCase will not allow the use of spell checkers

1203 ♦ Strict use of CamelCase makes it impossible to use separators (“.”) and
1204 therefore doesn’t allow an unambiguous identification of the composing
1205 parts of the *Dictionary Entry Name*

1206 [C23] The name of an *Object Class* refers to an activity or object within a *Business*
1207 *Context*. It shall be unique throughout the dictionary and may consist of more
1208 than one word.

1209 [C24] The name of a *Property Term* shall occur naturally in the definition and may
1210 consist of more than one word. A name of a *Property Term* shall be unique
1211 within the context of an *Object Class* but may be reused across different
1212 *Object Classes*.

1213 [Example]

1214 *Car. Colour. Code* and *Shirt. Colour. Code* may both exist

1215 [C25] If the name of the *Property Term* uses the same word as the *Representation*
1216 *Term* (or an equivalent word), this *Property Term* shall be removed from
1217 *Dictionary Entry Name*. The *Representation Term* word in this case only will
1218 remain.

1219 [Example]

1220 If the *Object Class* is *Goods*, the *Property Term* is *Delivery Date*, and *Representation*
1221 *Term* is *Date*, the *Dictionary Entry Name* is *Goods. Delivery. Date*; the *Dictionary*
1222 *Entry Name* for an identifier of a party (*Party. Identification. Identifier*) will be
1223 truncated to *Party. Identifier*.

1224 [C26] The name of the *Representation Term* shall be one of the terms specified in the
1225 *List of Representation Terms* as included in this document (See section
1226 6.1.3.3).

1227 [C27] The name of the *Representation Term* shall not be truncated in the *Dictionary*
1228 *Entry Name*.

1229 [C28] The *Dictionary Entry Name* of a *Core Component Type* shall consist of a
1230 meaningful type name followed by a dot and the term *Type*.

1231 [Example]

1232 *Amount. Type; Date Time. Type*

1233 [C29] The *Dictionary Entry Name* of an *Aggregate Core Component* shall consist of
1234 a meaningful aggregate name followed by a dot and the term *Details*. The
1235 aggregate name may consist of more than one word.

1236 [Example]

1237 *Postal Address. Details, Party. Details*

1238

1239 6.1.3.2.4 Rules for Business Terms

1240 *Business Terms* are those terms that are commonly used for day-to-day information
 1241 exchanges within a given domain. As such, no specific naming rules apply to *Business*
 1242 *Terms*. Interoperability of *Business Terms* will be given by linking them to *Core*
 1243 *Component* dictionary entries.

1244 6.1.3.3 List of Representation Terms

1245 The *Representation Term* is the part of a *Core Component* name that describes the
 1246 form of valid values in which the business information is expressed in a data item. For
 1247 instance all basic *Core Components* representing a monetary amount shall be named
 1248 *[Name]. Amount* where *[Name]* represents a specialisation of the generic amount and
 1249 *Amount* is the *Representation Term*. Table 6-3 lists the permissible *Representation*
 1250 *Terms*.

1251 [C30] The *Representation Term* shall be one of the list of permissible *Representation*
 1252 *Terms*

1253 **Table 6-3 Permissible Representation Terms**

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

Representation Term	Definition	Links to Core Component Type
Identifier	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme. [Note: Type shall not be used when a person or an object is identified by its name. In this case the <i>Representation Term</i> “Name” shall be used.]	Identifier. Type
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
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

1254 In addition to permissible representation terms for *Core Components*, there are also
1255 permissible representation terms for *Aggregate Core Components* and *Core*
1256 *Component Types*. Table 6-4 contains the permissible representation terms that apply
1257 to *Aggregate Core Components* or *Core Component Types*.

1258 [C31] The *Representation Term* for *Aggregate Core Components* or *Core*
 1259 *Component Types* shall be one of the list of permissible *Aggregate Core*
 1260 *components* or *Core Component Type Representation Terms*

1261 **Table 6-4 Permissible Representation Terms for Aggregate Core Components or**
 1262 **Core Component Types**

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

1263 6.1.4 Catalogue of Core Components

1264 As originally articulated in the ebXML architecture concept and perpetuated in the
 1265 developing UN/CEFACT architecture concept, all *Core Components* will be recorded
 1266 in an ebXML compliant registry and stored in a related repository. However, small
 1267 and medium enterprise (SME) organisations may not be able to readily access such
 1268 architecture. As such, it is important that the full range of *UN/CEFACT Core*
 1269 *Components* be published in a freely available catalogue. This catalogue must convey
 1270 the full details of each *Core Component* consistent with how those components are
 1271 stored as UML objects in the repository. Table 6-5 identifies a proper format for the
 1272 catalogue and contains representative entries from the existing *UN/CEFACT Core*
 1273 *Components Catalogue*.

1274 The catalogue is intended to be part of a larger *Core Component Library*. The *Core*
 1275 *Component Library* will consist of the following parts:

- 1276 • *Core Component Types*
- 1277 • *Catalogue of Core Components*, including *Basic Core Components*, *Basic*
 1278 *Business Information Entities*, and *Aggregate Core Components*
- 1279 • *Catalogue of Aggregate Business Information Entities*

1280

1280 **Table 6-5. Core Component Catalogue**

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

1281

1282 **6.1.5 Catalogue of Business Information Entities**

1283 For the same reasons that a *Catalogue of Core Components* is necessary, a *Catalogue*
1284 *of Business Information Entities* is also required. Predefined BIEs are not provided.
1285 Rather, the working registries and the groups defining business messages will be
1286 responsible for developing a *Catalogue of Business Information Entities* that is
1287 comparable to the *Catalogue of Core Components*.

6.2 Context

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

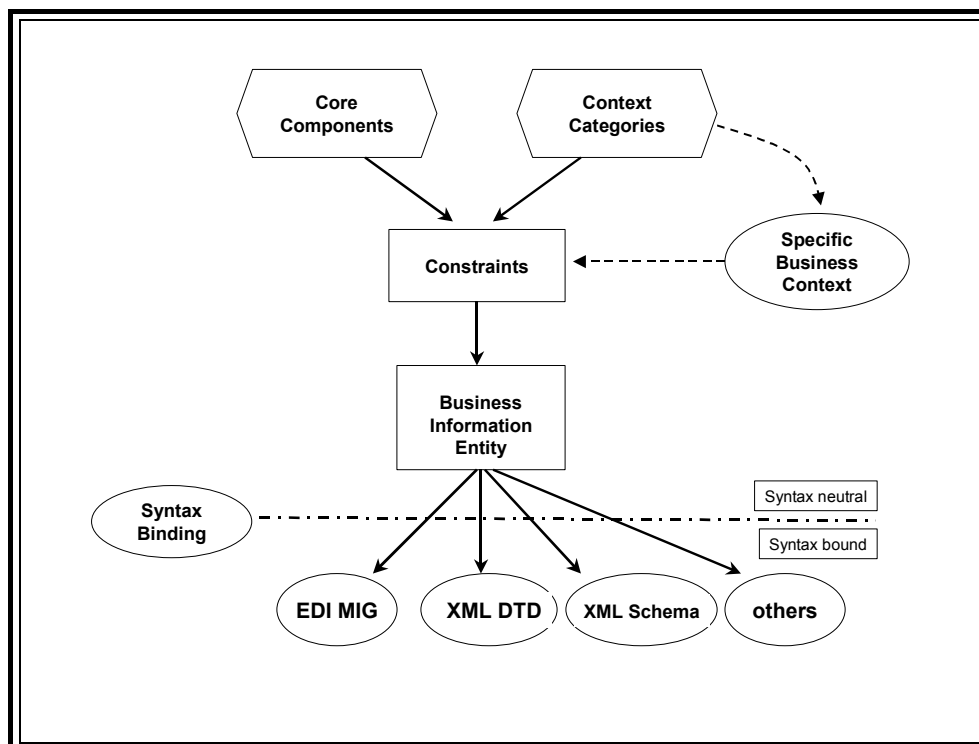
6.2.1 Overview of Context Specification

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

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

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

Figure 6-3. Operation of The Context Mechanism



1310 6.2.1.1 Context Categories

1311 *Context Categories* exist to allow users to uniquely identify and distinguish between
1312 different business contexts. Eight context categories have been identified. Each of the
1313 identified categories – unless otherwise stated - uses a standard classification to
1314 provide values for the category. Constraint rules – and therefore, BIEs – are tied to a
1315 particular set of standard classifications for identifying and distinguishing contexts.

1316

1317 6.2.1.2 Constraint Language

1318 A *Constraint Language* is used to express the relationship between specific *Business*
1319 *Contexts* and how semantics are applied to the *Core Components* to produce *Business*
1320 *Information Entities*. The scope of this language covers two functional parts:

- 1321 • *Assembly* of a large aggregate (the *Document*);
- 1322 • Refinement of the assembly as appropriate. Refinement is both the
1323 addition of semantics specific to the business process, and the restriction
1324 and extension of the semantic model.

1325 This separation is a convenience for implementation (it simplifies the creation of
1326 processing tools) and creation of *standard* assemblies that can then be refined by
1327 specific users (a process that resembles how EDI standards and message
1328 implementation guides function today).

1329 Both *Constraint Language* parts allow, for example, simple commands indicating
1330 how *Core Components* will be used, how they will be named for these specific uses,
1331 and how to refine the cardinality (if necessary). Further, conditional relationships can
1332 be expressed. Specific context values or sets of values can be tied to the actions
1333 performed on *Core Components* to produce *Business Information Entities*.

1334 [Example]

1335 If the *Geopolitical Process Context* has a value of *Anywhere in the European Union*,
1336 and the specific *Business Context Value* indicates that the business process occurs in
1337 France, then the context-appropriate *Business Information Entity* can be assembled by
1338 modifying the correct *Core Component*.

1339 The constraint language would say—If the *Geopolitical Process Context* equals the
1340 *European Union*, then take the core *NameAddress* component and rules to provide the
1341 correct names, cardinality, and arrangement to the fields. To do business in France,
1342 the specific context value for that process will trigger this rule, giving a set of
1343 appropriate business semantics (*Business Information Entities*).

1344 6.2.1.3 Syntax Binding

1345 The *Business Information Entity* is a model that has no relationship to a specific
 1346 syntax. It is intended that any given *Business Information Entity* can be expressed in
 1347 any number of syntaxes. This process is called *syntax binding*, and is independent of
 1348 (has no relationship to) a specific syntax. It may be possible to express *syntax binding*
 1349 in an algorithm.

1350 6.2.2 Approved Context Categories

1351 Table 6-6 contains the eight approved *Context Categories*.

1352 [C32] When describing a specific *Business Context*, a set of values will be assigned
 1353 to the business situation being formally described.

1354 [C33] Applied *Business Context* will be from the list of approved context categories.

1355 **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 an 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).
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

Context Category	Description
	existing back office can only support an address in a certain form).

1356 6.2.2.1 Business Process Context

1357 In describing a business situation, generally the most important aspect of that situation
 1358 is the business activity being conducted. *Business Process Context* provides a way to
 1359 unambiguously identify the business activity. To ensure consistency with business
 1360 process activities, it is important to use a common point of reference. The definitive
 1361 point of reference for international standards is the UN/CEFACT *Catalogue of*
 1362 *Common Business Processes*.

1363 [C34] Assigned *Business Process Contexts* shall be from the standard hierarchical
 1364 classification: provided as part of the UN/CEFACT *Catalogue of Common*
 1365 *Business Processes*.

1366 [C35] *Business Process Context* values may be expressed as a single business
 1367 process at any level, or may be expressed as a set of business processes at any
 1368 level.

1369 [C36] *Business Process Context* values may be taken from extensions to the business
 1370 processes described in the *Catalogue of Common Business Processes* as
 1371 provided for in that document.

1372 [C37] When business process extensions are used, they shall include full information
 1373 for each value sufficient to unambiguously identify which extension is
 1374 providing the value used.

1375 6.2.2.2 Product Classification Context

1376 The *Product Classification Context* describes those aspects of a business situation
 1377 related to the goods or services being exchanged by, or otherwise manipulated, or
 1378 concerned, in the business process. Recognized code lists exist that provide
 1379 authoritative sources of product classification contexts.

1380 [C38] A single value or set of values may be used in a *Product Classification*
 1381 *Context*.

1382 [C39] If a hierarchical system of values is used for *Product Classification Context*,
 1383 then these values may be at any level of the hierarchy.

1384 [C40] If more than one classification system is being employed, an additional value
 1385 specifying which classification scheme has supplied the values used shall be
 1386 conveyed.

1387 [C41] Product classification context code values shall be taken from recognized code
1388 lists to include:

- 1389 • *Universal Standard Product and Service Specification* (UNSPSC)
 - 1390 - Custodian: Electronic Commerce Code Management Association
 - 1391 (ECCMA)
- 1392 • *Standard International Trade Classification* (SITC Rev .3)
 - 1393 - Custodian: United Nations Statistics Division (UNSD)
- 1394 • *Harmonized Commodity Description and Coding System* (HS)
 - 1395 - Custodian: World Trade Organization (WTO)
- 1396 • *Classification Of the purposes of non Profit Institutions serving*
1397 *households* (COPI)
 - 1398 - Custodian: UNSD (This provides a mapping between the first three.)

1399 6.2.2.3 Industry Classification Context

1400 The *Industry Classification Context* provides a description of the industry or sub-
1401 industry in which the business process takes place.

1402 [C42] An *Industry Classification Context* may contain a single value or set of values
1403 at any appropriate level of the value hierarchy.

1404 [C43] The *Industry Classification Context* value hierarchy must be identified.

1405 [C44] *Industry Classification Context* code values shall be taken from recognized
1406 code lists to include:

- 1407 • *International Standard Industrial Classification* (ISIC)
 - 1408 - Custodian: UNSD
- 1409 • *Universal Standard Product and Service Specification* (UNSPSC) Top-
1410 level Segment [digits 1 and 2] used to define industry.
 - 1411 - Custodian: ECCMA

1412 [Note]

1413 There are many other industry classification schemes that may be used for *Industry*
1414 *Classification Context*.

1415 6.2.2.4 Geopolitical Context

1416 *Geopolitical Contexts* allow description of those aspects of the business context that
 1417 are related to region, nationality, or geographically based cultural factors.

1418 [C45] *Geopolitical Context* shall consist of appropriate continent, economic region,
 1419 country, and region identifiers.

1420 [C46] *Geopolitical Regional Classification* may associate one or more values with
 1421 any business message or component.

1422 [C47] *Geopolitical Regional Classification* shall employ the following hierarchical
 1423 structure:

1424 Global

1425 [Continent]

1426 [Economic Region]

1427 [Country] - ISO 3166.1

1428 [Region] - ISO 3166.2

1429 [C48] At any level of the *Geopolitical Regional Classification* hierarchy, a value
 1430 may be a single value, a named aggregate, or cross-border value.

1431 [C49] *Geopolitical Regional Classification* hierarchy values shall structured as
 1432 follows:

- 1433 • **Single Value:** A single value indicating a single continent, economic
 1434 region, country, or region, depending on position within the hierarchy.
- 1435 • **Named Aggregate:** A related group of values (which may themselves be
 1436 single values, named aggregates, or cross-border constructions), which
 1437 have been related and assigned a name. A named aggregate contains at
 1438 least two values.
- 1439 • **Cross-Border:** One or more pairs of values, designated *To*, *From*, or *Bi-*
 1440 *directional*, indicating the direction of cross-border context. Values may be
 1441 named aggregates or single values.

1442 [Example]

1443 The following example shows an extract of the basic, single-value hierarchy of
 1444 recommended values, based on the common ISO 3166.1 *Country Codes*. (The value at
 1445 the top of any hierarchy is always understood to be *Global*.)

1446 Europe

1447 Eastern Europe

1448 AL – ALBANIA

1449 AM – ARMENIA

1450 [C50] Points in the *Geopolitical Regional Classification* hierarchy shall be specified
1451 by the use of the node value, or by the full or partial path.

1452 [C51] The full path of the *Geopolitical Regional Classification* hierarchy must be
1453 used to understand the hierarchy when complex constructs are employed.

1454 [C52] A single-point specification is understood to inherit all of the properties of the
1455 single-value hierarchy except where otherwise specified.

1456 [C53] *Geopolitical Values* will be taken from ISO 3166.1 and 3166.2

1457 6.2.2.5 Official Constraints Context

1458 The *Official Constraints Context* category describes those aspects of the business
1459 situation that result from legal or regulatory requirements and similar *official*
1460 categories. This category contains two distinct parts:

1461 • Regulatory and Legislative. These are normally unilateral in nature and
1462 include such things as customs.

1463 • Conventions and Treaties (These are normally bi- or multilateral
1464 agreements and as such are different from regulatory and legislative
1465 constraints.

1466 [C54] The *Official Constraints Context* will consist of at least two values:

1467 • Identification of the legal or other classification used to identify the
1468 context values.

1469 • Identification of the official constraint itself. These values may represent a
1470 hierarchical structure depending on the official constraints system being
1471 referenced.

1472 Because there is no known global classification of all *Official Constraints Contexts* as
1473 used here, any implementation must provide a set of recognized official constraints
1474 classifications for use within the appropriate *Core Components* Registry
1475 implementation.

1476 [C55] Individual *Core Component* implementations shall register used official
1477 constraint classification schemes with the appropriate supporting *Core*
1478 *Components* Registry implementation.

1479 6.2.2.6 Business Process Role Context

1480 The *Business Process Role Context* describes those aspects of a business situation that
1481 are specific to an actor or actors within the business process. Its values are taken from
1482 the set of *Role* values provided by the *Catalogue of Common Business Processes*. A

1483 *Business Process Role Context* is specified by using a value or set of values from this
1484 source.

1485 [C56] *Business Process Role Context* values shall be taken from an approved list
1486 provided by the business process model library being employed.

1487 [C57] The UN/CEFACT *Catalogue of Common Business Processes* shall be the
1488 definitive source of *Business Process Role Context* values for all
1489 UN/CEFACT *Business Information Entities*.

1490 [Issue]

1491 A normative dictionary of role values is expected and is considered the responsibility
1492 of the library where the business process models are stored.

1493 6.2.2.7 Supporting Role Context

1494 The *Supporting Role Context* identifies those parties that are not active participants in
1495 the business process being conducted but who are interested in it. A *Supporting Role*
1496 *Context* is specified with a value or set of values from a standard classification.

1497 [C58] *Supporting Role Context* values shall be taken from the UN/EDIFACT *Code*
1498 *List for DE 3035 Party Roles*.

1499 [Note]

1500 Users are cautioned that duplication exists in the current version of the required code
1501 list. UN/CEFACT will review this code list to clarify duplicates and identify non-
1502 *Supporting Role Context* values.

1503 6.2.2.8 System Capabilities Context

1504 This category identifies a system, a class of systems or standard in the business
1505 situation. The *System Capabilities Context* requires a least one pair of values: an
1506 identification of the classification scheme being used and a value from that scheme. A
1507 valid *System Capabilities Context* may include more than one such pair of values.

1508 [C59] *Systems Capabilities Context* values shall consist of pairs of values. Each pair
1509 shall be comprised of an identification of the referenced classification scheme
1510 and the value(s) being employed.

1511 [Note]

1512 There is no known classification of all types of information systems and standards. It
 1513 is recommended that a mechanism for the registration of system and standard names
 1514 be provided by the ebXML registry, as valid values for the *System Capabilities*
 1515 *Context*.

1516 6.2.3 Context Values

1517 A specific business context is formally described using a set of context values. Every
 1518 context category must have a valid value, even if this value is *In All Contexts* or *None*.
 1519 The value *None* is appropriate for *Official Constraints Context* because there will be
 1520 instances where there are no official constraints.

1521 [C60] The *In All Contexts* value shall be a valid value for every context category
 1522 except for *Official Constraints Context*.

1523 [C61] The value *None* shall be a valid value for *Official Constraints Context*.

1524 6.2.4 Core Components Context Constraints Language

1525 The *Core Components Context Constraints Language* consists of a set of values (See
 1526 Table 6-7) that allow users to express the relationships between specific business
 1527 situations and the specific structure and meaning of business data used in that
 1528 situation. The constraints language refers to specific contexts as described in the
 1529 *Context Categories* specification and uses UIDs to refer to *Core Components*
 1530 semantic models. The constraints applied to *Core Components* in specific business
 1531 contexts to generate *Business Information Entities* are expressed using the constraints
 1532 language.

1533 [C62] The *Core Components Context Constraints Language* shall be used to describe
 1534 the constraints being applied to *Core Components* to create *Business*
 1535 *Information Entities*.

1536 An *Assembly* is the overall expression of a single set of *Assembly Rules*, which groups
 1537 a set of unrefined *Business Information Entities* in to a larger structure. When working
 1538 with pre-assembled standard document sets, it should not be necessary for users to
 1539 create *Assembly* constraints.

1540 [C63] *Assembly* shall be the top level construct in any set of *Assembly Rules*.

1541 [Ed. Note]

1542 The next version will have two UML class diagrams inserted here, one for *Assembly*
 1543 and one for *ContextRules*.

1544 The *ContextRules* construct is the overall expression of a single set of *Context Rules*.
 1545 These add the full semantic and structural refinement to the *Core Components* to
 1546 produce *Business Information Entities* .

1547 [C64] A single set of context rules shall be described using the *Context Rules*
 1548 expression.

1549 **Table 6-7 Core Components Context 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 <i>Core Components</i> to be grouped together to form BIEs
	@id	ID of an Assembly
	@idref	Reference to an Assembly id
	@version	Version of the Assembly Rules document.
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 <i>Core Components</i>
	CreateGroup	Create a group of BIEs
	@name	Name of the highest-level BIE being assembled
	@id	ID of an Assemble rule
	@idref	Reference to an Assemble id
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

Construct	Component Constructs	Description
		MaxOccurs followed by zero or more CreateGroup or Rename, or UseBIE, or Condition or Annotation, optionally an @id or an @idref, and an optional @location
	Type	Type of BIE to be created – a reference to a <i>Core Component</i>
	MinOccurs	Minimum occurrences for the BIE created
	MaxOccurs	Maximum occurrences for the BIE created. One possible value (other than integer) is ‘unbounded’.
	@id	Id of the created BIE
	@idref	Reference to the ID of another created BIE
	Name	Name of the BIE to be assembled
	@location	Location of the BIE to be assembled (i.e. query to the registry)
	Rename	Renames children of the created BIE
	Condition	Condition under which this rule should 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 <i>Core Component</i> , 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.

Construct	Component Constructs	Description
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
	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

Construct	Component Constructs	Description
	@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
	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

Construct	Component Constructs	Description
		Annotation, and optionally an @id or an @idref
	Name	Name of BIE to be modified
	Type	Type of BIE – the <i>Core Component</i> - required only if contained in an Add tag
	Attribute	Attribute(s) of this BIE
	Annotation	Insert Annotation
	@id	Id of the BIE rule
	@idref	Reference to the ID of another BIE rule
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
MaxOccurs		Maximum number of occurrences in the output
Annotation		An Annotation contains zero or more of either Documentation or Appinfo, and

Construct	Component Constructs	Description
		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 @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

1550

1551 [Note]

1552 Table Key: @ indicates properties of the construct being defined (@id, @idref and
 1553 @version are properties of Assembly.

1554 6.2.4.1 Assembly Construct

1555 The *MinOccurs* and *MaxOccurs* constructs in the *CreateBIE* construct specify the
 1556 occurrence that the created *Business Information Entity* will have in the resulting
 1557 semantic model.

1558 [C65] A *Business Information Entity* created with *MinOccurs* = 1 and *MaxOccurs* =
 1559 1 shall be specified in the resulting semantic model as occurring only once.

1560 [C66] An *Assembly* may contain more than one assembled top-level semantic model.

1561 6.2.4.2 ContextRule Construct

1562 Several built-in variables are used to access context information. These variables
 1563 correspond to the identified context categories. All of these variables have string
 1564 values.

1565 [C67] The *Apply* attribute of the *ContextRule* construct type shall be used for
 1566 determining the behaviour of rules that use hierarchical values.

1567 [C68] Allowed *Apply* attribute values are:

- 1568 • *exact* - match only if the value in the provided context is precisely the
- 1569 same as that specified in the rule
- 1570 • *hierarchical* - match if the value provided is the same or a child of that
- 1571 specified in the rule.

1572 [Example]

1573 If the *ContextRule* specifies the region *Europe*, the value *France* would match only if

1574 the *Apply* attribute is set to *hierarchical* (*exact* being the default).

1575 [C69] The *Attribute* construct has four optional children in its content model, of

1576 which at least one must be present.

1577 [C70] When the *Attribute* construct is used to refine an existing *Attribute*, then a

1578 value must be specified for *@applyTo* on that *Attribute* construct.

1579 [C71] *ContextRules* must refer to the names of the *Core Components*, and not the

1580 names given to the resulting *Business Information Entities* elsewhere in the

1581 Rules.

1582 [Example]

1583 Given a source that contains an optional child type named 'X', a rule can be applied to

1584 rename 'X' to 'Y', but a rule to make 'Y' required, rather than 'X', would be illegal.

1585 6.2.4.3 Output Constraints

1586 [C72] Semantic models and document definitions produced through the application

1587 of *Assembly* and *Context Rules* must contain the metadata about the rules and

1588 context that produced them.

1589 6.2.4.4 Ordering and Application

1590 There is an explicit *Order* property on the *Rule* construct that applies a sequence to

1591 the application of a set of rules. It is an error for two *Rule* constructs to have the same

1592 value for the property *Order*. In a single set of *ContextRules*, users should be careful

1593 not to sequence rules in a way that would preclude their execution—such as adding an

1594 attribute to a business Information Entity that has not been added yet by the rules.

1595 [C73] The *Order* property on the *Rule* construct will determine the sequence for the

1596 application of the applicable set of rules.

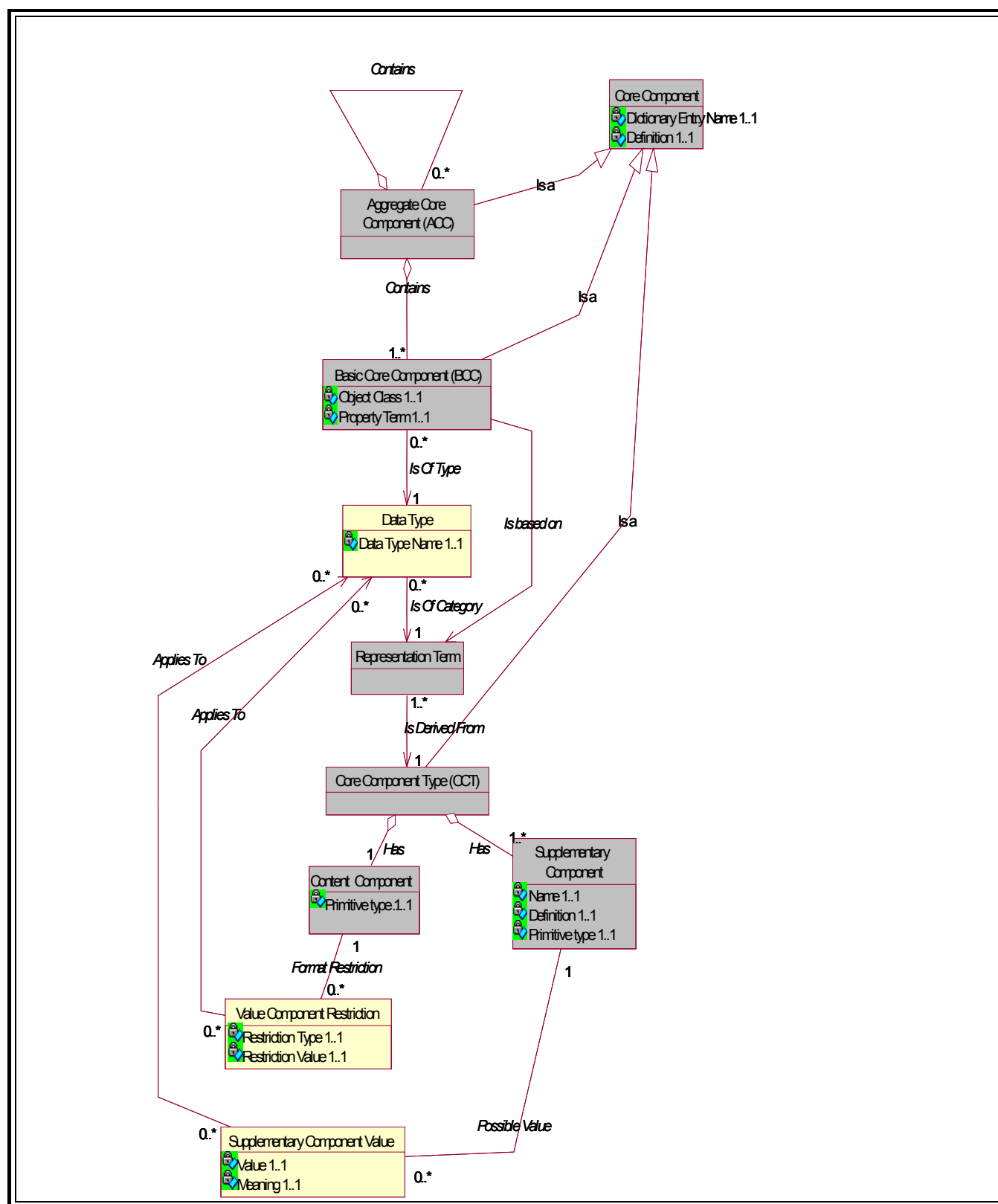
1597 [C74] Two *Rule* constructs must not have the same value for the property *Order*.

7 Technical Details - Core Component Repository Storage

Section 6.1 specifies the *Core Component* basic definition. This section details exact information required for creation of *Universal Modelling Language* objects to store *Core Components* in the repository and to store relevant metadata about the *Core Components*. Both parts contain requirements that must be addressed by developers of *Core Components* and users of *Core Components*. Further, both parts contain requirements that must be satisfied in the supported registry and repository suite of technical specifications and any corresponding overarching information technology framework that uses *Core Components* as the linchpin between process modelling and trade.

7.1 Storing Core Components

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

1612 **Figure 7-1. Core Components - Full Definition**

1613 7.1.1 Stored Core Components

1614 [S1] Stored *Core Components* shall always be defined as one of the three recognized
1615 types—*Basic Core Component*, *Aggregate Core Component*
1616 or *Core Component Type*.

1617 [S2] Stored *Core Components* shall include the following attributes:

- 1618 • *Dictionary Entry Name 1..1*: where the *Dictionary Entry Name* is the unique
1619 official name of the *Core Component* in the dictionary.
- 1620 • *Definition 1..1*: where the *Definition* is the unique semantic business
1621 meaning of the *Core Component*.

1622 7.1.2 Stored Basic Core Components

1623 [S3] Stored *Basic Core Components* shall always be based on three elements: (1) an
1624 *Object Class*, which defines the overall business concept to which the *Basic Core*
1625 *Component* belongs, (2) a *Property Term*, which defines the specific characteristic
1626 of the business concept that is covered by the *Basic Core Component* and (3) a
1627 *Data Type* as expressed as the *Representation Term*.

1628 [S4] Stored *Basic Core Components* shall include the following Attributes:

- 1629 • ***Object Class 1..1***: where the *Object Class* represents the logical data
1630 grouping (in a logical data model) to which a data element belongs
1631 (ISO11179). The *Object Class* is the part of a *Core Component's Dictionary*
1632 *Entry Name* that represents an activity or object in a specific context.
- 1633 • ***Property Term 1..1***: where the *Property Term* identifies one of the
1634 characteristics belonging to the Object (Class)

1635 [S5] Stored *Basic Core Components* shall reflect association with a stored *Core*
1636 *Component Type*.

1637 [S6] Stored *Basic Core Component Data Types* shall be based on a *Representation*
1638 *Term* derived from a *Core Component Type*.

1639 7.1.3 Stored Core Component Types

1640 [S7] Stored *Core Component Types* shall include one *Content Component* that defines
1641 the primitive type and one or more *Supplementary Components* that give meaning
1642 to the *Content Component*.

1643 [S8] Stored *Core Component Types* shall not reflect business meaning.

1644 7.1.4 Stored Aggregate Core Components

1645 [S9] *Stored Aggregate Core Components* shall consist of two or more *Basic Core*
1646 *Components*, or at least one *Basic Core Component* plus one or more *Aggregate*
1647 *Core Components*.

1648 [Note]

1649 As shown in Figure 7-1, when the *Core Component Type* is used as the basis for a
1650 particular data type, the content component and supplementary components can be
1651 restricted. This is expressed in the diagram through the existence of the classes
1652 *Supplementary Component Value* and *Content component Restriction* and their *Applies*
1653 *To* relation with *Data Type*.

1654 [S10] *Stored Aggregate Core Components* shall reflect relationships between the *Basic*
1655 *Core Components* and *Aggregate Core Components* from which it is constructed.

1656 7.1.5 Stored Content Components and Supplementary Components

1657 [S11] Restrictions on *Stored Content Components* and *Supplementary Components*
1658 shall be identified when the *Core Component Type* is used as basis for a
1659 particular *Data Type*.

1660 7.1.6 Stored Data Types

1661 [S12] *Stored Data Types* shall define the full range of valid values that can be used for a
1662 particular *Basic Core Component* and will include the following attribute:

- 1663 • ***Data Type Name 1..1***: Official name of the *Data Type*.

1664 [Note]

1665 Defines the set of valid values that can be used for a particular *Basic Core Component*. It
1666 is defined by specifying restrictions on the *Core Component Type* that forms the basis of
1667 the *Representation Term* from which the *Data Type* is derived.

1668 7.1.7 Stored Representation Term

1669 [S13] *Stored Representation Terms* for *Basic Core Components* shall define the type of
1670 valid value and will include the following attribute:

- 1671 • ***Representation Term Name 1..1***: Official name of the *Representation Term*

1672 7.1.8 Stored Supplementary Components

1673 [S14] *Stored Supplementary Components* shall be associated with the
1674 ContentComponent in the overarching *Core Component Type* and shall include
1675 the following attributes:

- 1676 • Name 1..1: Name is the official name of the *Supplementary Component* of a
1677 *Core Component Type*.
- 1678 • Definition 1..1: Definition is a clear, unambiguous and complete explanation
1679 of the meaning of a *Supplementary Component* and its relevance for the
1680 related *Core Component Type*.
- 1681 • Primitive type 1..1: Primitive type to be used for the representation of the
1682 value of a *Supplementary Component*.

1683 [Note]

1684 Possible values are String, Decimal, Integer, Boolean, Date.

1685 7.1.9 Stored Supplementary Component Value

1686 [S15] A stored *Supplementary Component Value* shall define an enumerated list of
1687 possible values of a *Supplementary Component*.

1688 [S16] A stored *Supplementary Component Value* shall only be stored if the values can
1689 be defined by an enumeration (e.g. list of quantity units).

1690 [Note]

1691 The list of possible *Stored Supplementary Component Values* can be further restricted
1692 when a *Core Component Type* is used for a particular *Basic Component*. Example: the
1693 *Core Component Type Quantity* has a *Supplementary Component Quantity Unit* with
1694 possible values like gram and second. A *Basic Component* like *Person. Weight. Quantity*
1695 will not accept *second* as quantity unit.

1696 The list can be further restricted when used in a particular context.

1697 7.1.10 Stored Supplementary Component Values

1698 [S17] *Stored Supplementary Component Values* shall contain the following Attributes:

- 1699 • Value 1..1: Value is a possible value of a *Supplementary Component*.

- 1700 • Meaning 1..1: Meaning describes the meaning of the *Supplementary*
1701 *Component* when it has a particular Value.

1702 **7.1.11 Stored Content Components**

1703 [S18] Stored *Content Components* shall contain the primitive type that must be used to
1704 express the value of a *Core Component Type*.

1705 [S19] Stored *Content Components* shall contain the following attribute:

- 1706 • **Primitive type 1..1:** Primitive type to be used for the representation of the
1707 value of a *Core Component Type*.

1708 **7.1.12 Stored Content Component Restrictions**

1709 [S20] Stored *Content Component Restrictions* shall define a format restriction that
1710 applies to the possible values of a *Content Component*.

1711 [Note]

1712 Possible values are String, Decimal, Integer, Boolean, Date.

1713 [S21] Stored *Content Component Restrictions* shall only exist if the values can be
1714 defined by a format restriction such as string pattern, minimum or maximum
1715 length, or enumeration.

1716 [S22] Stored *Content Component Restrictions* shall contain the following attributes:

- 1717 • **Restriction Type 1..1:** *Restriction Type* defines the type of format restriction
1718 that must be applied to the *Content Component*.
- 1719 • **Restriction Value 1..1:** *Restriction Value* is the actual value of the *Restriction*
1720 *Type* that applies to a *Content Component*.

1721 [Note]

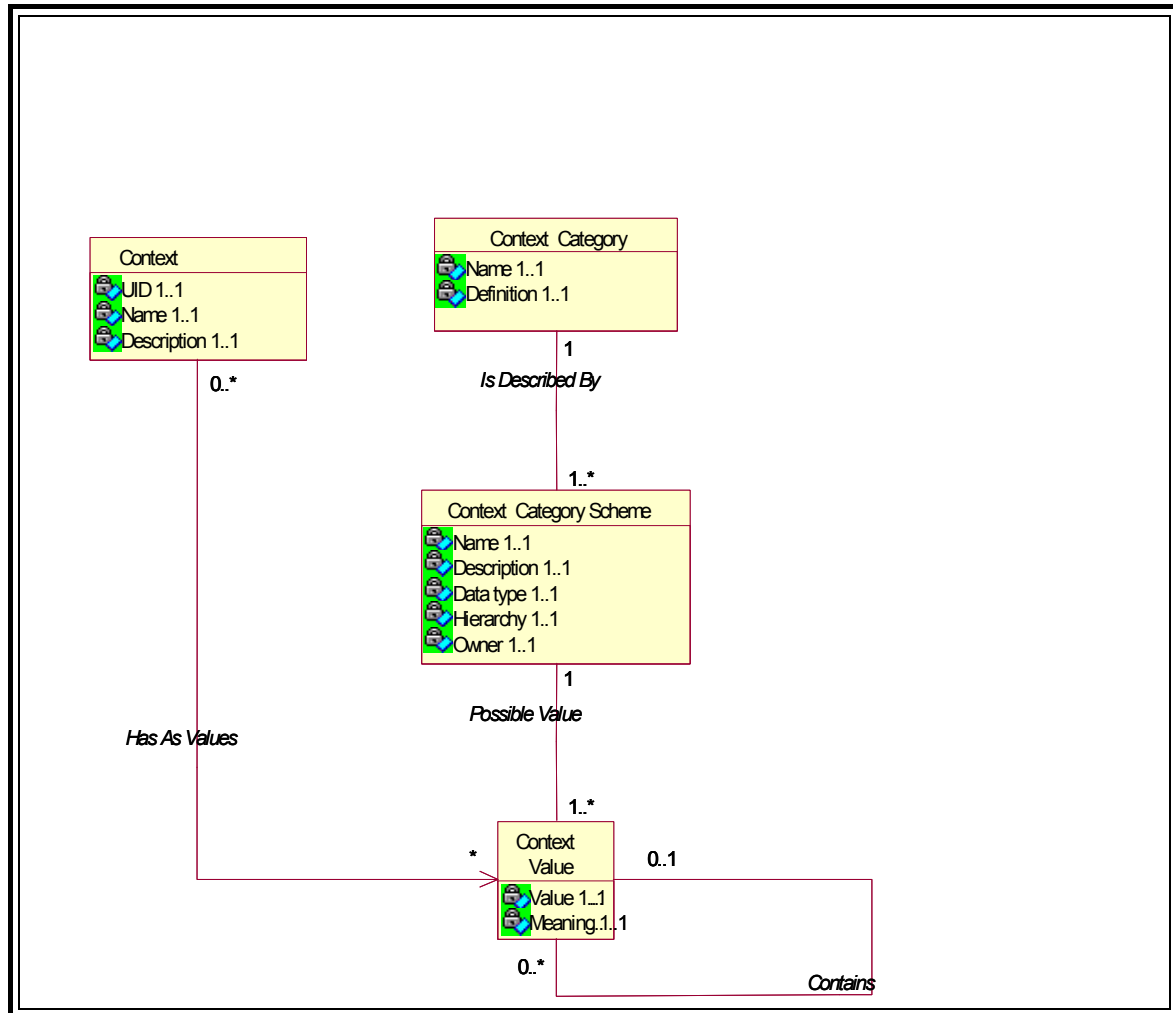
1722 Possible values include pattern, length, minimum length, maximum length, enumeration,
1723 and others to be identified. The possible values depend on the restriction type (e.g. integer
1724 for a *length* restriction type, list of possible values for an *enumeration* restriction type.).

1725 **7.2 Stored Context**

1726 Figure 7-2 models the function of context for storage. It shows that there are a number of
1727 *Context Categories* (e.g. Region, Product), which can each be described by one or more
1728 Schemes (e.g. United Nations scheme for products, World Trade Organization scheme for

1729 products). For each Scheme the list of possible values (and their meaning) is defined. A
 1730 *Business Context* is then defined as a unique and meaningful combination of *Context*
 1731 *Values*.

1732 **Figure 7-2 Core Components Context Definition Model**



1733 [S23] Stored *Business Context* shall contain the combination of values for context
 1734 categories so as to define a unique and meaningful business context.

1737 [S24] Stored *Business Contexts* shall contain the following attributes:

- 1738 • **UID 1..1:** Unique Identifier of a *Business Context*
- 1739 • **Name 1..1:** Name of the *Business Context*
- 1740 • **Description 1..1:** Description of the meaning of a *Business Context*

1741 **7.2.1 Context categories**

1742 [S25] Stored *Context Categories* shall be in conformance with the officially accepted
 1743 categories of *Core Component* contexts.

1744 [S26] Stored *Context Categories* shall contain the following attributes:

- 1745 • **Name 1..1:** Name is the official name of the *Context Categories*.
- 1746 • **Definition 1..1:** Definition gives the meaning of the *Context Categories* for
 1747 *Core Components*.

1748 **7.2.2 Context categories Scheme**

1749 [S27] Stored *Context Categories Scheme* shall be an officially supported *Scheme* to
 1750 describe a given *Context Category*.

1751 [S28] A *Context Category* may be described by one or more *Context Categories*
 1752 *Schemes*.

1753 [Issue]

1754 Should *Context Categories Scheme* be included. In other words, can a scheme be defined
 1755 and used. At least some members of the team believe that they should be named so that
 1756 users can define locally significant extensions to schemes defined by other people—such
 1757 as extending ISO 3166 to allow the separate regions of the United Kingdom to be
 1758 identified as South-West, South-East.

1759 [S29] Stored *Context Categories Schemes* shall contain the following attributes:

- 1760 • **Name 1..1:** Name under which the *Context Categories Scheme* is known.
- 1761 • **Description 1..1:** Description of the *Context Categories Scheme*.
- 1762 • **Data type 1..1:** Data type is the primitive type that is used for the
 1763 representation of a value in the *Context Categories Scheme*.

1764 [Note]

1765 Possible values are String, Decimal, Integer, Boolean, Date.

- 1766 • **Hierarchy 1..1:** Indicator describing whether the *Context Categories Scheme*
 1767 supports a hierarchical description of the context.

- 1768 • **Owner 1..1:** Organisation that is responsible for the *Context Categories*
 1769 *Scheme*.

1770 7.2.3 Context Value

1771 [S30] Stored shall describe a particular context in a given Context categories according
 1772 to a particular Context categories Scheme. If the *Context Category Scheme* allows
 1773 a hierarchy, the *Contains* value describes this hierarchy.

1774 [S31] Stored *Context Value(s)* shall contain the following attributes:

- 1775 • Attributes:
- 1776 • Value 1..1: Value describing a particular context.
- 1777 • Meaning 1..1: Description of the meaning of the corresponding value.

1778 [Note]

1779 The context value is derived from a business process model which presumably uses
 1780 values that have their meaning defined somewhere. For example, if the value is taken
 1781 from a code list (specified in the *Context categories Scheme*), then the meaning of the
 1782 code should be provided by the code list specification. As an alternative solution,
 1783 Meaning could be an optional Uniform Resource Identifier that points to the definition.

1784 7.3 Stored Business Information Entities

1785 Figure 7-3 models the types of *Business Information Entities* and their relationships. A
 1786 *Business Information Entity* is defined as a piece of business data or a group of pieces of
 1787 business data with a unique business semantic definition in a specified business context.

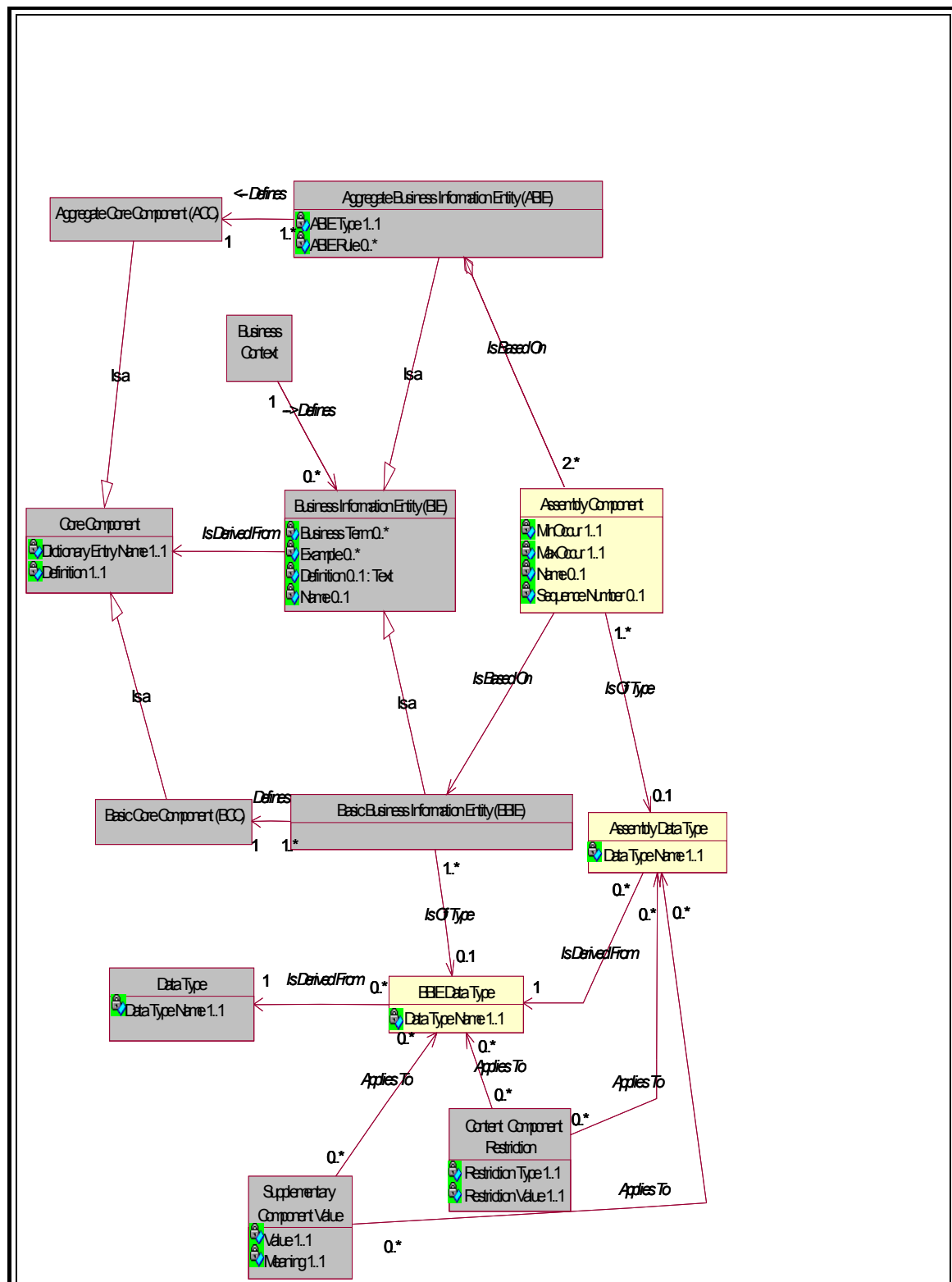
1788 All *Basic Business Information Entities* that relate to the same context-free concept form
 1789 the basis of the definition of a *Basic Core Component*.

1790 All *Aggregate Business Information Entities* that relate to the same context-free concept
 1791 form the basis of the definition of an *Aggregate Core Component*.

1792 An *Aggregate Business Information Entity* is either a sequence or a choice and will
 1793 consist of two or more *Assembly Components*, which are either *Basic Business*
 1794 *Information Entities* or *Aggregate Business Information Entities*. Each *Assembly*
 1795 *Component* has a certain cardinality (i.e. it is mandatory, optional and/or repetitive) and -
 1796 in case of a sequence - a sequence number. When used as an *Assembly Component*, it is
 1797 possible to change the name of the composing *Aggregate Business Information Entity* or
 1798 *Basic Business Information Entity* and to restrict the data type of a composing *Basic*
 1799 *Business Information Entity*.

1800

1800 **Figure 7-3. Business Information Entities – Full Definition**



1801 7.3.1 Stored Aggregate Business Information Entities

1802 [S32] *Stored Aggregate Business Information Entities* shall consist of a collection of
 1803 related pieces of business information and will convey a distinct business meaning
 1804 in a specified business context.

1805 [S33] *Stored Aggregate Business Information Entities* shall contain the following
 1806 attributes:

- 1807 • **ABIE Type 1..1:** *Aggregate Business Information Entity Type* indicates
 1808 whether the composing components of the *Aggregate Business Information*
 1809 *Entity* form a sequence (i.e. all composing components may occur when the
 1810 *Aggregate Business Information Entity* is used) or a choice (i.e. only one of
 1811 the composing components may occur when the *Aggregate Business*
 1812 *Information Entity* is used).
- 1813 • **ABIE Rule 0..*:** *Aggregate Business Information Entity Rule* describes a
 1814 restriction that relates to various *Assembly Components* of the *Aggregate*
 1815 *Business Information Entity*.

1816 7.3.2 Stored Assembly Component

1817 [S34] A stored *Assembly Component* shall be either an *Aggregate Business Information*
 1818 *Entity* or a *Basic Business Information Entity* that is a component of an *Aggregate*
 1819 *Business Information Entity*. It will specify the cardinality, and may specify the
 1820 alternative name and the sequence number to be used.

1821 [S35] *Stored Assembly Components* shall contain the following attributes:

- 1822 • **MinOccur 1..1:** Minimum number of occurrences that a composing *Business*
 1823 *Information Entity* must occur when used in an *Aggregate Business*
 1824 *Information Entity*. If the minimum is zero, the component is optional. If the
 1825 minimum is one or more, the component is mandatory.

1826 [Issue]

1827 Figure 7-3 shows the *Aggregate* and *Basic Business Information Entities* define the *Core*
 1828 *Components* they are derived from. Some in the team believe that whilst they may be
 1829 used to identify holes in the *Core Component* list, they are in general, more likely to be
 1830 derived from existing *Core Components* rather than the source of new ones. Note that
 1831 *Data Type* is marked as is derived from, which is inconsistent with the other *Core*
 1832 *Components*.

- 1833 • **MaxOccur 1..1:** Maximum number of occurrences that a composing *Business*
 1834 *Information Entity* may occur when used in an *Aggregate Business*
 1835 *Information Entity*. If the maximum is zero, the component is not allowed. If

1836 the maximum is more than one, the component is repetitive. The defined
1837 maximum must always be greater than or equal to the defined minimum.

1838 • **Name 0..1**: Optional alternative name to be used for a *Business Information*
1839 *Entity* when used in an *Aggregate Business Information Entity*.

1840 • **Sequence Number 0..1**: Position of the *Assembly Component* in an *Aggregate*
1841 *Business Information Entity* of type Sequence.

1842 **7.3.3 Stored Assembly Data Type**

1843 [S36] Stored *Assembly Data Types* shall define the set of valid values that can be used
1844 for a particular *Basic Business Information Entity* when used in a particular
1845 *Aggregate Business Information Entity*.

1846 [S37] Stored *Assembly Data Types* shall be defined by specifying restrictions on the
1847 *Content Component* and *Supplementary Components*.

1848 [S38] Stored *Assembly Data Types* shall contain the following attribute:

1849 • **Data Type Name 1..1**: Official name of the *Data Type*.

1850 **7.3.4 Basic Business Information Entity (BBIE)**

1851 [S39] Stored *Basic Business Information Entities* shall define a piece of business
1852 information with a unique concept having a single business semantic definition in
1853 a specified *Business Context*.

1854 **7.3.5 Basic Business Information Entity Data Type**

1855 [S40] Stored *Basic Business Information Entity Data Types* shall define the set of valid
1856 values that can be used for a particular Basic information Entity. It will be defined
1857 by specifying restrictions on the *Content Component* and *Supplementary*
1858 *Components*.

1859 [S41] Stored *Basic Business Information Entity Data Types* shall contain the following
1860 attributes:

1861 • **Data Type Name 1..1**: Official name of the *Data Type*.

1862 **7.3.6 Business Information Entity**

1863 [S42] Stored *Business Information Entities* shall have a unique business semantic
1864 definition in a specified business context.

1865 [S43] Stored *Business Information Entities* shall be categorized as either a *Basic*
1866 *Business Information Entity* or an *Aggregate Business Information Entity*.

1867 [S44] A stored *Business Information Entity* shall contain the following attributes:

1868 • **Business Term 0..***: A synonym term under which the *Core Component* is
1869 commonly known and used in the business. A *Core Component* may have
1870 several business terms or synonyms.

1871 • **Example 0..***: An example of a possible value of a *Core Component* in a
1872 given business context

1873 • **Definition 0..1**: Context dependent definition of a *Core Component*

1874 • **Name 0..1**: Context dependent name of a *Core Component*

1875 **7.3.7 Data Type**

1876 [S45] Stored *Data Types* shall contain the set of valid values that can be used for a
1877 particular *Basic Core Component*.

1878 [S46] Stored *Data Types* shall be defined by specifying restrictions on the *Core*
1879 *Component Type* that forms the basis of the *Representation Term* from which the
1880 *Data Type* is derived.

1881 **7.3.8 Supplementary Component Value**

1882 [S47] Stored *Supplementary Component Value(s)* shall define an enumerated list of
1883 possible values of a *Supplementary Component*. This will only exist if the values
1884 can be defined by an enumeration (e.g. list of quantity units).

1885 [S48] Stored *Supplementary Component Values* shall contain the following attributes:

1886 • **Value 1..1**: *Value* is a possible value of a *Supplementary Component*.

1887 • **Meaning 1..1**: Meaning describes the meaning of the *Supplementary*
1888 *Component* when it has a particular *Value*.

1889 [Note]

1890 The list of possible values can be further restricted when a *Core Component Type* is used
1891 for a particular *Basic Component*. Example: the *Core Component Type Quantity* has a
1892 supplementary component *Quantity Unit* with possible values like gram and second. A
1893 *Basic Core Component* like *Person. Weight. Quantity* will not accept *second* as quantity
1894 unit. The list can be further restricted when used in a particular context.

1895 **7.3.9 Stored Content Component Restriction**

1896 [S49] Stored *Content Component Restrictions* shall define a format restriction that
1897 applies to the possible values of a *Content Component*.

- 1898 [S50] *Stored Content Component Restrictions* shall only exist if the values can be
 1899 defined by a format restriction such as string pattern, minimum or maximum
 1900 length, or enumeration.
- 1901 [S51] *Stored Content Component Restrictions* shall contain the following attributes:
- 1902 • **Restriction Type 1..1:** *Restriction Type* defines the type of format restriction
 1903 that must be applied to the *Content Component*. Possible values include
 1904 pattern, length, minimum length, maximum length, enumeration, etc.
 - 1905 • **Restriction Value 1..1:** *Restriction Value* is the actual value of the *Restriction*
 1906 *Type* that applies to a *Content Component*. The possible values depend on the
 1907 restriction type (e.g. integer for a *Length Restriction Type*, list of possible
 1908 values for an *Enumeration Restriction Type*, ...).

1909 7.4 Core Component Storage Metadata

1910 *Core Components* and *Business Information Entities* are the linchpins for developing
 1911 standardized process models and business documents. Storing these artefacts so that they
 1912 are able to meet this role requires rich metadata storage as well.

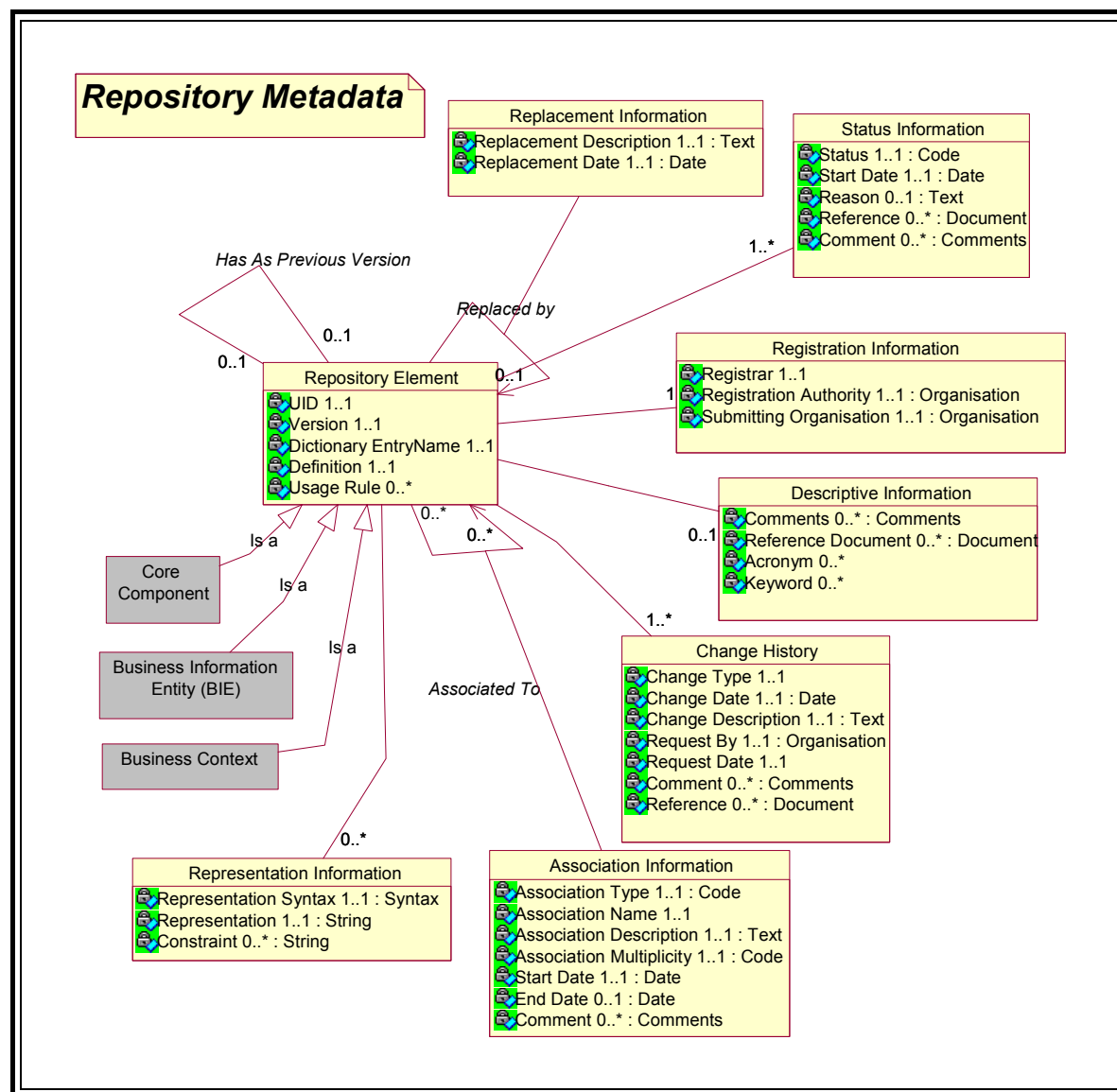
1913 Figure 7-4 focuses on the meta-information that needs to be defined for a *Repository*
 1914 *Element* (i.e. all information needed to store for *Core Components* and for *Business*
 1915 *Information Entities*). To simplify the diagram all information regarding the structure of a
 1916 *Core Component* and a *Business Information Entity* has been hidden.

1917 As shown in Figure 7-4, the following metadata categories will be required:

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

- 1934 • *Change History*: information about all changes that are made to a *Repository Element*.
- 1935
- 1936 • *Association Information*: a *Repository Element* may be associated to multiple
- 1937 other *Repository Elements*.
- 1938 • *Representation Information*: information about the physical representation of a
- 1939 *Repository Element* in a particular syntax (e.g. to document the XML-tag).

1940 **Figure 7-4. Repository Metadata**



1943 [Ed. Note]

1944 Figure 7-4 requires correction to reflect storage of context as part of the repository
1945 element stored data.

1946 **7.4.1 General Metadata Storage Rules**

- 1947 [S52] Stored *Repository Elements* shall include a universally unique identifier.
- 1948 [S53] Stored *Repository Elements* shall include a *Version Number* to keep track of the
1949 evolution over time of a *Repository Element*.
- 1950 [S54] Stored *Repository Elements* may include one or more *Usage Rules*, describing
1951 how and/or when to use the *Repository Element*.
- 1952 [S55] Except for the first version of a *Repository Element*, each stored version shall be
1953 linked to its previous version.
- 1954 [S56] Except for the last version of a *Repository Element*, each stored version shall be
1955 linked to its next version.
- 1956 [S57] Stored *Repository Elements* shall include the history of the status lifecycle of each
1957 version.

1958 **7.4.2 Management Information**

1959 **7.4.2.1 Administrative Information**

- 1960 [S58] Stored *Repository Elements* shall contain administrative information and will
1961 include the following attributes:
- 1962 • Registrar 1..1: Name of the responsible person who has created the *Repository*
1963 *Element* in the repository
 - 1964 • Registration Authority 1..1: Organisation authorised to register the *Repository*
1965 *Element*.
 - 1966 • Submitting Organisation 1..1: The organisation that has submitted / requested
1967 the *Repository Element*

1968 **7.4.2.2 Status Information**

- 1969 [S66] Stored *Repository Elements* shall contain status information to include the
1970 following attributes:
- 1971 • Status 1..1: Status of the *Repository Element* (i.e. draft, provisionally
1972 registered, registered, to be retired, retired, ...)
 - 1973 • Start Date 1..1: Date on which the status comes into effect
 - 1974 • Reason 0..1: Description of why the *Repository Element* status has been
1975 changed.

- 1976 • Reference 0..*: External Document(s) containing relevant information about
1977 the status change.
- 1978 • Comment 0..*: Remark about the *Repository Element* status.
- 1979 **7.4.2.3 Change History**
- 1980 [S60] Stored *Repository Elements* shall include the history of all modifications related to
1981 each version to include the following attributes:
- 1982 • Change Type 1..1: Purpose of the Change—such as *new element*, *new version*,
1983 *element modification*, *status modification*, *element replacement*.
- 1984 • Change Date 1..1: Date on which the modification has been made
- 1985 • Change Description 1..1: Description of why and how the *Repository Element*
1986 has been modified.
- 1987 • Request By 1..1: Name of the organisation that has requested the modification
1988 of the *Repository Element*
- 1989 • Request Date 1..1: Date on which the modification was requested.
- 1990 • Comment 0..*: Remark about the *Repository Element* modification.
- 1991 • Reference 0..*: External Document(s) containing relevant information about
1992 the modification.
- 1993 **7.4.2.4 Replacement Information**
- 1994 [S62] For each stored pair of *Repository Elements* where one *Repository Element*
1995 replaces the other, the stored information shall specify replacement information to
1996 include the following attributes:
- 1997 • Replacement Description 1..1: Reason for the *Repository Element* being
1998 replaced
- 1999 • Replacement Date 1..1: Date from which the replacement is effective.
- 2000 [S63] If another *Repository Element* has replaced a *Repository Element*, it shall be
2001 linked to the *Repository Element* by which it has been replaced.
- 2002 [S64] If a *Repository Element* replaces one or more other *Repository Element*, it shall be
2003 linked to the *Repository Element(s)* it replaces

2004 7.4.3 Content Information

2005 7.4.3.1 Descriptive Information

2006 [S61] Stored *Repository Elements* may optionally include additional descriptive
2007 information to include the following attributes:

- 2008 • **Comments 0..*:** Comments is additional information about a *Repository*
2009 *Element*, which is not part of the definition but that is considered relevant for
2010 clarification.
- 2011 • **Reference Document 0..*:** Reference Document is a reference (e.g. *Uniform*
2012 *Resource Locator*) to external documentation that contains relevant additional
2013 information about a *Repository Element*.
- 2014 • **Acronym 0..*:** Acronym is an abbreviation or code under which the *Semantic*
2015 *Information Component* is commonly known.
- 2016 • **Keyword 0..*:** Keyword is one or more significant words used for the search
2017 and retrieval of a *Semantic Information Component*.

2018 7.4.3.2 Representation Information

2019 [S65] Stored *Repository Elements* may optionally include information about the
2020 representation of the *Repository Element* in one or more syntaxes to include the
2021 following attributes.

- 2022 • Representation Syntax 1..1: Identification of the representation syntax
- 2023 • Representation 1..1: Physical representation of the *Repository Element* (e.g.
2024 *Extensible Markup Language* tag)
- 2025 • Constraint 0..*: Description of additional constraints that apply to the
2026 representation of the *Repository Element* in the given syntax (e.g. maximum
2027 length, ...)

2028 7.4.3.3 Association Information

2029 [S59] Stored *Repository Elements* shall include all associations they have with other
2030 stored *Repository Elements* and shall include the following attributes:

- 2031 • Association Name 1..1: Name of the association
- 2032 • Association Description 1..1: Descriptive text explaining the meaning of the
2033 association
- 2034 • Association Type 1..1: Type of association (e.g. aggregation, specialisation,
2035 generalisation, simple association ...)

- 2036 • Association Multiplicity 1..1: Cardinality of the association (i.e.
2037 optional/mandatory and repetition)
- 2038 • Start Date 1..1: Date at which the association becomes valid
- 2039 • End Date 0..1..1*: Date from which the association is no longer valid
- 2040 • Comment 0..*: Relevant information about the association (e.g. reason why it
2041 has been removed, ...)
- 2042

2043 8 Approved Core Component Type, Content, and 2044 Supplementary Components

2045 The Following subsections contain tables that convey the currently approved *Core*
2046 *Component Types* (Section 8.1) and *Core Component Type* Content and
2047 *Supplementary Components* (Section 8.2).

2048 8.1 Approved Core Component Types

2049 [Note]

2050 The UIDs in Table 6-1 are interim in nature and will be finalized prior to release of
2051 this document in specification status.]

2052 **Table 8-1 Core Component Types (CCT)**

2053

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)
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

UID	CCT Dictionary Entry Name	Definition	Remarks	Object Class	Property Term	CCT Components
000101	Identifier. Type	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme together with relevant supplementary information.		Identifier	Type	<ul style="list-style-type: none"> 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
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)

2054

[Issue]

2055

The Team has agreed on *Language.Code* as a valid *Core Component Type*

2056

Component for *Code. Type*. An issue was raised during the last review cycle in which

2057

it was requested to remove *Language. Code* with the argument that "this information

2058

will be maintained by the Agency for each identifier value. It is no supplementary

2059

information about the full list of codes." The team does not agree, but reserves

2060

judgement pending any additional comments received during the last eBTWG review

2061

cycle.

2062

2063 [Issue]

2064 The Team has agreed on the current definition of *Identifier. Type*. An issue was raised
 2065 during the last review cycle in which it was requested to expand this type to include
 2066 additional qualifiers. Specifically, the submitter stated: "change the names of things in
 2067 Table 6.2 of the *Core Component* specification which now have *Code* as the last term
 2068 in their names, but are not really *Code Type*.

2069 For example,

2070 *Amount Currency. Identification. Code* -- change to

2071 *Amount Currency. Code. Text* -- or maybe, just,

2072 *Amount. Currency Code* (data type string)

2073 *Language. Code* -- change to

2074 *Language. Language Code* (data type string) --or maybe change its usage,

2075 e.g.

2076 *Text. Content* and

2077 *Text. Language Code*.

2078 The team will make a final decision based on inputs received during the final eBTWG
 2079 review cycle.

2080 **8.2 Approved Core Component Type Content and** 2081 **Supplementary Components**

2082 Table 8-2 presents the currently approved set of *Core Component Type Content* and
 2083 *Supplementary Components*. The asterisk (*) in the property term column indicates
 2084 cases where the property term is the same as either the representation term or object
 2085 class, and is consequently dropped from the dictionary entry name.

2086

2086

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

2087

UID	Name	Data-type	Definition	Remarks
000106	Amount. Content	decimal	A number of monetary units specified in a currency where the unit of currency is explicit or implied	
000107	Amount Currency. Identification. Code	string	The currency of the amount	Reference ISO 4217.
000091	Code. Content	string	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute	
000093	Code List. Agency. Identifier	string	An agency that maintains one or more code lists	
000092	Code List. Identifier	string	The name of a list of codes	Can be used to identify the URL of a source that defines the set of currently approved permitted values
000099	Code List. Version. Identifier	string	The version of the code list	
000100	Code. Name	string	The textual equivalent of the code content	If no code content exists, the code name can be used on its own
000067	Date time. Content	string	The particular point in the progression of time	
000068	Date Time. Format. Text	string	The format of the date/time content	Reference ISO 8601
000202	Graphic. Content	binary	A diagram, graph, mathematical curves, or similar representation	
000203	Graphic. Format. Text	string	The format of the graphic content	
000104	Identification Scheme Agency. Name	string	The agency that maintains the identification scheme	
000103	Identification Scheme. Name	string	The name of the identification scheme	
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

UID	Name	Data-type	Definition	Remarks
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	

2088

2088 9 Definition of Terms

2089 **Aggregate Core Component - (ACC)** – A collection of *Core Components* that convey
2090 a distinct business meaning. An *Aggregate Core Component* will consist of two or
2091 more *Basic Core Components*, or at least one *Basic Core Component* plus one or
2092 more *Aggregate Core Components*.

2093 **Aggregate Business Information Entity (ABIE)**– A collection of related pieces of
2094 business information that together convey a distinct business meaning in a specified
2095 business context.

2096 **Aggregate Core Component - (ACC)** – A collection of pieces of business information
2097 that together form a single business concept (e.g. postal address). Each *Aggregate*
2098 *Core Component* has its own unique business semantic definition and can contain
2099 either:

- 2100 • two or more *Basic Core Components*, or
- 2101 • at least one *Basic Core Component* plus one or more *Aggregate Core*
2102 *Components*

2103 **Basic Business Information Entity (BBIE)** – A *Basic Business Information Entity* is
2104 derived from a *Basic Core Component*.

2105 **Basic Core Component (BCC)** – A *Core Component* with a unique concept having a
2106 single business semantic definition. It must be constructed by using a *Core*
2107 *Component Type*.

2108 **Business Context** – The formal description of a specific business circumstance as
2109 identified by the values of a set of context categories, allowing different business
2110 circumstances to be uniquely distinguished.

2111 **Business Information Entity (BIE)** – A *Business Information Entity* is a piece of
2112 business data or a group of pieces of business data with a unique business semantic
2113 definition. A BIE can be either a *Basic Business Information Entity* or an *Aggregate*
2114 *Business Information Entity*.

2115 **Business Term** – This is a synonym term under which the *Core Component* is
2116 commonly known and used in the business. A *Core Component* may have several
2117 business terms or synonyms.

2118 **Constraint Language** – A formal expression of actions occurring in specific contexts
2119 to assemble, structurally refine, and semantically qualify *Core Components*. The
2120 result of applying the constraint language to a set of *Core Components* in a specific
2121 context is a set of *Business Information Entities*.

- 2122 ***Content Component*** - Defines the primitive type used to express the content of a
2123 *Core Component Type*.
- 2124 ***Context Category*** – A group of one or more related values used to express one
2125 characteristic of a business circumstance.
- 2126 ***Context Information Entity*** – The influence of a particular context on the restriction
2127 on a reusable semantic building block for the exchange of business-related
2128 information.
- 2129 ***Controlled Vocabulary*** - A supplementary Vocabulary to define uniquely any words
2130 or business terms that are potentially ambiguous. This is to ensure that every word
2131 within any of the *Core Component* names and definitions is used in a consistent and
2132 unambiguous way and this will also aid accurate language translations.
- 2133 ***Core Component (CC)*** – A building block for the creation of a semantically correct
2134 and meaningful information exchange ‘parcel’. It contains only the information pieces
2135 necessary to describe a specific concept. A *Core Component* will always be defined as
2136 a *Basic Core Component*, a *Core Component Type*, or an *Aggregate Core Component*.
- 2137 ***Core Component Administrative Information*** – Administrative information
2138 regarding a *Core Component*
- 2139 ***Core Component Association Information*** – Information about the association
2140 between two *Core Components*.
- 2141 ***Core Component Change History*** – History of the modifications applied to a *Core*
2142 *Component* version.
- 2143 ***Core Component Replacement Information*** – Information about the replacement of a
2144 *Core Component* by another.
- 2145 ***Core Component Representation Information*** – Information about the physical
2146 representation of a *Core Component* in a particular syntax.
- 2147 ***Core Component Status Information*** – History of the lifecycle of a particular version
2148 of a *Core Component*.
- 2149 ***Core Component Type (CCT)***– A *Core Component* that consists of one and only one
2150 *Content Component* that carries the actual content plus one or more supplementary
2151 components giving an essential extra definition to the content component. *Core*
2152 *Component Types* do not have business meaning.
- 2153 ***Data Type*** – Defines the set of valid values that can be used for a particular BCC. It is
2154 defined by specifying restrictions on the *Core Component Type* that forms the basis of
2155 the *Representation Term* from which the *Data Type* is derived.

- 2156 **Definition** - This is the unique semantic business meaning of a *Core Component*
- 2157 **Dictionary Entry Name** – This is the unique official name of a *Core Component* in
2158 the dictionary.
- 2159 **Information Entity** – A reusable semantic building block for the exchange of
2160 business-related information.
- 2161 **Object Class** – The logical data grouping (in a logical data model) to which a data
2162 element belongs (ISO11179). The *Object Class* is the part of a *Core Component*'s
2163 *Dictionary Entry Name* that represents an activity or object in a specific context.
- 2164 **Primitive Type** – Primitive type used for the representation of the value of a
2165 *Supplementary Component*. Possible values are String, Decimal, Integer, Boolean,
2166 Date.
- 2167 **Property Term** – This identifies one of the characteristics belonging to the *Object*
2168 *Class*.
- 2169 **Representation Term** – The type of valid values for a *Basic Core Component*.
- 2170 **Supplementary Component** – Gives meaning to the *Content Component* in the *Core*
2171 *Component Type*.
- 2172 **User Community** – A user community is a group of practitioners, with a publicised
2173 contact address, who may define context profiles relevant to their area of business.
2174 Users within the community do not create, define or manage their individual context
2175 needs but conform to the community's standard. Such a community should liaise
2176 closely with other communities and with general standards-making bodies to avoid
2177 overlapping work and to avoid creating multiple *Towers of Babel*. A community may,
2178 of course, be as small as two consenting organisations!
- 2179

2180 **10 References**

- 2181 — ebXML Technical Architecture Specification v1.04ebXML Business Process
2182 Specification Schema v1.01
- 2183 — ebXML Registry Information Model v1.0
- 2184 — ebXML Registry Services Specification v1.0
- 2185 — ebXML Requirements Specification v1.06
- 2186 — ebXML Collaboration-Protocol Profile and Agreement Specification v1.0
- 2187 — ebXML Message Service Specification v1.0ebXML Technical Reports
- 2188 — Business Process and Business Information Analysis Overview v1.0Business
2189 Process Analysis Worksheets & Guidelines v1.0 -
- 2190 — E-Commerce Patterns v1.0
- 2191 — Catalog of Common Business Processes v1.0Core Component Overview v1.05
- 2192 — Core Component Discovery and Analysis v1.04
- 2193 — Context and Re-Usability of Core Components v1.04
- 2194 — Guide to the Core Components Dictionary v1.04
- 2195 — Naming Convention for Core Components v1.04
- 2196 — Document Assembly and Context Rules v1.04
- 2197 — Catalogue of Context Categories v1.04
- 2198 — Core Component Dictionary v1.04
- 2199 — Core Component Structure v1.04
- 2200 — Information Technology — Metadata registries: Framework for the
2201 Specification and Standardization of Data Elements, International
2202 Standardization Organization, ISO 11179-1
- 2203 — Information Technology — Metadata registries: Classification of Concepts for
2204 the Identification of Domains, International Standardization Organization, ISO
2205 11179-2
- 2206 — Information Technology — Metadata registries: Registry Metamodel,
2207 International Standardization Organization, ISO 11179-3
- 2208 — Information Technology — Metadata registries:Rules and Guidelines for the
2209 Formulation of Data Definitions, International Standardization Organization,
2210 ISO 11179-4
- 2211 — Information Technology — Metadata registries: Naming and Identification
2212 Principles for Data Elements, International Standardization Organization, ISO
2213 11179-5
- 2214 — Information Technology — Metadata registries: Framework for the
2215 Specification and Standardization of Data Elements, International
2216 Standardization Organization, ISO 11179-
- 2217 — Information Technologies – Open-edi Reference Model, ISO/IEC 14662

- 2218 — Information Technologies – Business Agreement Semantic Descriptive
- 2219 Techniques – Part 1: Operational Aspects of Open-edl for Implementation ,
- 2220 ISO/IEC 15944-1
- 2221 — UN/CEFACT Modeling Methodology, UN/CEFACT TMWG N090
- 2222 — Information Technologies – IT Enablement for Widely Used Coded Domains,
- 2223 ISO/IEC 18022
- 2224 — Information Technologies – Identification and Mapping of Various Categories
- 2225 of Jurisdictional Domains, ISO/IEC 18038

2226 11 Disclaimer

2227 The views and specification expressed in this document are those of the authors and
2228 are not necessarily those of their employers. The authors and their employers
2229 specifically disclaim responsibility for any problems arising from correct or incorrect
2230 implementation or use of this design.

2231 12 Contact Information

2232 Team Leader

2233 Name Hartmut Hermes

2234 Company Siemens AG

2235 Street Richard Strauss Strasse 76

2236 City, state, zip/other 81679 Munich

2237 Nation Germany

2238

2239 Phone: (089) 92 21-4564

2240 Email: hartmut.hermes@mch11.siemens.de

2241

2242 Editor

2243 Name Mark Crawford

2244 Company Logistics Management Institute

2245 Street 2000 Corporate Ridge

2246 City, state, zip/other McLean, Virginia 22102

2247 Nation USA

2248

2249 Phone: +01 703 917 7177

2250 Email: mcrawford@lmi.org

2251

Copyright Statement

Copyright © UN/CEFACT 2001. All Rights Reserved.

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

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

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