United Nations
Centre for Trade Facilitation and Electronic Business

CCTS Version 3.0 Implementation Verification Second Iteration

18 December 2008

Abstract

The Core Components Technical Specification defines a meta model and rules necessary for describing the structure and contents of conceptual and logical data models and information exchange models. The CCTS is described using the Unified Modelling Language (UML). It does not require UML in its implementation.

Table of Contents

1 Status of This Document

This UN/CEFACT Technical Specification is being developed in accordance with the UN/CEFACT/TRADE/R.650/Rev.4/Add.1/Rev.1 Open Development Process (ODP) for technical specifications. The CCTS Project Team has approved it for broad implementation verification.

This document contains information to guide in interpretation or implementation.

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

Distribution of this document is unlimited.

This version: UN/CEFACT Core Components Technical Specification Version 3.0 Implementation Verification Second Iteration 20081218

Previous version: UN/CEFACT Core Components Technical Specification, Version 3.0 Implementation Verification 20080207

This document may also be available in these non-normative formats: XML, XHTML with visible change markup. See also translations.

2 Core Components Technical Specification Project Team Participants

We would like to recognize the following for their significant participation to the development of this specification.

CCWG Chair	Mark Crawford, SAP Labs LLC (U.S.)
CCTS Chair	Jim Wilson, KCX Incorporated
Editor	Mark Crawford, SAP Labs LLC (U.S.)
Editing Team	Mary Kay Blantz, AIAG
	Anthony Coates, Miley Watts LLP, UK
Contributors	
	Oyvind Aassve, NorStella
	Ed Buchinski, Canada TBS
	Michael Dill, GEFEG, mbH
	Jostein Fromyr, EdiSys As
	Kenji Itoh, JASTPRO
	Garret Minakawa, Open Applications Group
	Sue Probert, SEPIAeb Ltd.
	Nada Reinprecht, Nemmco
	Jean-Luc Sanson, EDF
	Gunther Stuhec, SAP AG
	Fred Van Blommestein, Independent
	Sylvia Webb, GEFEG, US

2.1 Disclaimer

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

2.2 Contact Information

CCWG Chair	Mark Crawford, SAP Labs LLC (U.S.) mark.crawford@sap.com
CCTS Project Lead	Jim Wilson, KCX Incorporated jim.wilson@kcx.com

3 Introduction

This specification describes and specifies a semantic-based approach to the well-understood problem of the lack of information interoperability within and between applications and data bases in the e-business arena. Traditionally, data has been designed for specific applications and databases without regard to interoperability. Standards for the exchange of that business data between applications and databases have been focused on static message definitions that have not enabled a sufficient degree of interoperability or flexibility. A more flexible and interoperable way of standardizing business semantics has long been required.

The UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) core component solution described in this technical specification presents just such a methodology. This *Core Component Technical Specification* (CCTS) describes a revolutionary approach for developing a common set of semantic building blocks that represent the general types of business data in use today. This approach provides for the creation of new business vocabularies as well as restructuring of existing business vocabularies to achieve semantic interoperability of data.

3.1 Summary of Contents of Document

This specification consists of the following Sections.

- Abstract
- Table of Contents
- Section 1: Status
- Section 2: Project Team
- Section 3: Introduction
- Section 4: Objectives
- Section 5: Overview
- Section 6: Core Component Model
- Section 7: Business Information Entity Model
- Section 8: Data Types
- Section 9: Context
- Section 10: Definition of Terms

The Abstract, Table of Contents, and Sections 1, 2, 3, 4 and 5 are informative – with the exception of Section 4.2.1 Conformance which is normative. Sections 6, 7, 8 and 9 are normative, complementary and interdependent. Section 10 is normative.

In addition, the UN/CEFACT Forum will prepare supplemental documents that may be used in conjunction with this specification. These supplemental documents will include:

- Core Component Message Assembly (CCMA) – expands on the assembly principles contained in the CCTS and provides specific

methodology for assembling higher level business information entities (BIEs) for electronic messages.

- UN/CEFACT Context Methodology (UCM) –The context methodology provides a mechanism for business driven customization of BIEs.
- Data Type Catalogue – The collection of UN/CEFACT Permissible Representation Terms, Core Data Types, and Business Data Types.
- UML Profile for Core Components – Defines a UML profile for expressing core components in UML models.
- Core Components Library (CCL) – represents the work of various organizations working in a joint endeavour to develop and publish core component artefacts.

3.1.1 Notation

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.

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

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

[Note] – Explanatory information. Notes are informative.

[Rn] – Identification of a rule that requires conformance to ensure discovered core components (CCs) are properly defined, named and stored. The value R is a prefix to categorize the type of rule where R=A for Conformance rule, R=B for BIE rule, R=C for CC rule, R=X for Context rule, or R=D for Data Type (DT) rule. The value n (1..n) indicates the sequential number of the rule]. Rules are normative.

Italics – All words appearing in italics, when not titles or used for emphasis, are the first occurrences of special terms defined in Section 10.

`Courier` – All words appearing in bolded 10 point **`courier font`** are values or objects.

3.2 Audience

The CCTS can be employed wherever data is being defined, stored, used, shared or exchanged. It is especially well suited for defining data models and for creating data exchange standards for information flows amongst and between enterprises, governmental agencies, and/or other organizations in an open, global environment.

This specification forms the basis for international cross-industry standards development work of business analysts, business users and information technology specialists. The user community is drawn from both business and government, to include: data modellers, business document modellers, business process modellers,

---

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

and application developers of different organizations that require common understanding and interoperability of information.

3.3 Related Documents

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

- Information Technology – Specification and standardization of data elements – Part 1: Framework for the specification and standardization of data elements, International Standardization Organization, ISO 11179-1:1999
- Information Technology – Metadata registries (MDR) – Part 2: Classification, ISO 11179-2:Second Edition 2005-11-15
- Information Technology – Metadata registries (MDR) – Part 3: Registry Metamodel and Basic Attributes, ISO 11179-3(e):Second Edition 2003/Cor 1:2004
- Information Technology – Metadata registries (MDR) – Part 4: Formulation of Data Definitions, ISO 11179-4:Second Edition 2004-07-15
- Information Technology – Metadata registries (MDR) – Part 5: Naming and Identification Principles, ISO 11179-5:Second Edition 2005-09-01
- Information Technology - Metadata registries: Registration, ISO 11179-6: Second Edition 2005-01-15

4 Objectives

4.1 Goals of the Technical Specification

The CCTS has been developed to provide for standards based semantic modelling of business information. CCTS complements traditional data modelling techniques. The component models produced using CCTS may form the basis for syntax specific business information exchanges, but are independent of any specific technology platform or implementation language.

4.2 Requirements

Users of this specification should have an understanding of basic data modelling concepts and basic business information exchange concepts.

4.2.1 Conformance

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

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

4.3 Caveats and Assumptions

The components created as a result of employing this specification should be maintained in a universally freely accessible Core Component Library (CCL). UN/CEFACT will maintain their CCL in a UN/CEFACT specified registry and make its contents available to the entire core component community. It is recommended that all users of this specification submit their components for inclusion in the UN/CEFACT CCL.

5 Overview

This Core Components Technical Specification (CCTS) provides a way to identify, capture and maximize the re-use of business information to support and enhance information interoperability. The specification focuses both on human-readable and machine-processable representations of this information.

The CCTS approach is more flexible than current data and information exchange standards because the semantic standardization is done in a syntax-neutral fashion. This syntax-neutral semantic based methodology allows for the richness inherent in natural language to be used to create data and information exchange models that are devoid of computer-driven syntax limitations and requirements.

UN/CEFACT business process and core component solutions capture a wealth of information about the business reasons for variation in data model and message semantics and structure. In the past, these variations have led to incompatible models and a subsequent lack of interoperability. The core components mechanism will allow identification of similarities and differences between these models.

The CCTS key concepts are based on two levels of abstraction — core components and business information entities.

5.1 Core Components

The foundational concept of this specification is the core component. Core components are semantic building blocks that can be used for all aspects of data and information modelling and exchange. Core components are the linchpin for creating interoperable business process models and business documents. Core components are conceptual in nature, they are used for creating context specific BIEs as defined in Section 5.6.2. Figure 5-1 shows three different categories of CCs – aggregate core component (ACC), basic core component (BCC), and association core component (ASCC) that are discussed in the following subsections.

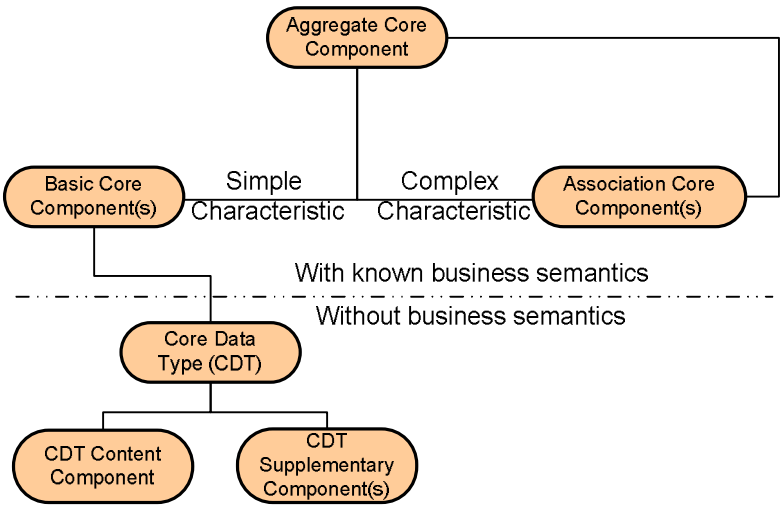


Figure 5-1. Core Component Overview

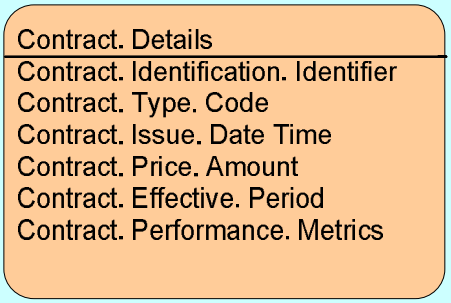
[Note] – Generic Terms

The term core component is used as a generic term that encompasses ACCs, BCCs, and ASCCs and their properties. Equally, the term business information entity is used as a generic term encompassing ABIEs, BBIEs, and ASBIEs and their properties.

5.1.1 Aggregate Core Component

An Aggregate Core Component (ACC) is a collection of related pieces of information that together convey a distinct meaning, independent of any business context. In data modelling terms, an ACC is the representation of an entity/object class, contains attributes/properties, and may participate in associations with other ACCs.

[Example] – Aggregate Core Component with Basic Core Component and Association Core Component Properties



Contract. Details ACC

A contract is an agreement between two or more parties, especially one that is written or spoken and enforceable by law.

Contract. Identification. Identifier BCC

A unique identification for this contract.

Contract. Type. Code BCC

A code specifying a type of contract such as a fixed price contract or a time and materials based contract.

Contract. Issue. Date Time BCC

A date or date time or other date time value of the issuance of this contract

Contract. Price. Amount BCC

Monetary value of a price of this contract

Contract. Effective. Period ASCC

A period within which the provisions of this contract are, or will be, in force or effective.

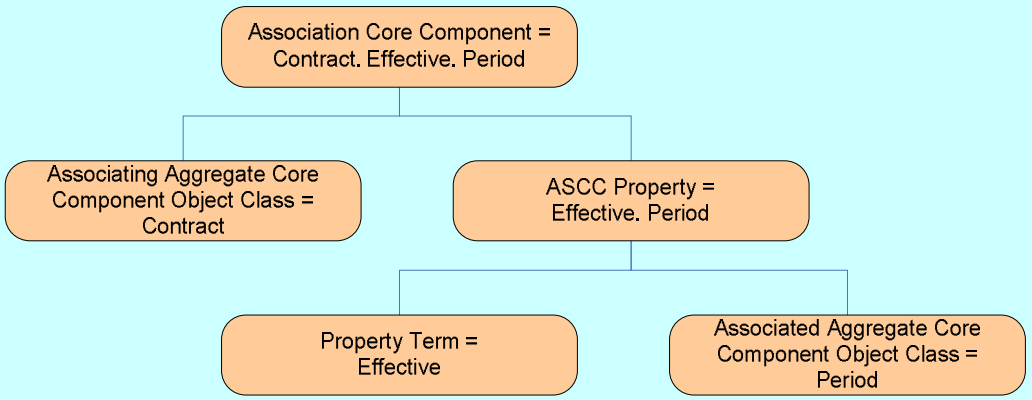
Contract. Performance. Metrics ASCC

Performance metrics for this contract.

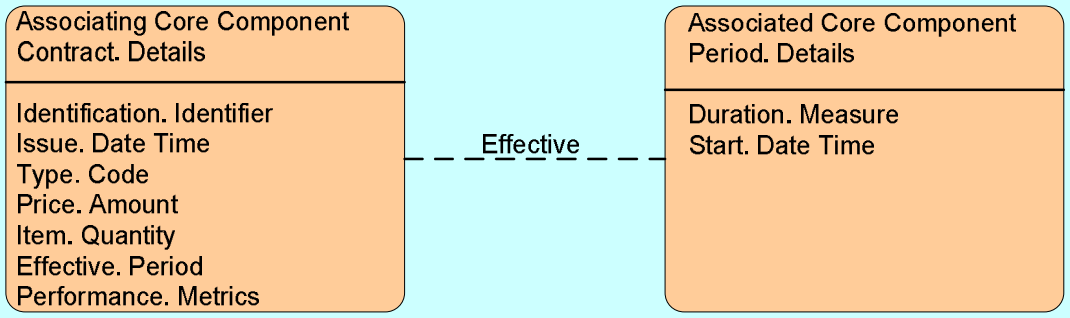
5.1.2 Association Core Component

An Association Core Component (ASCC) defines a role in an association between one ACC (known as the associating ACC) and another ACC (known as the associated ACC). An ASCC consists of an ASCC property plus the object class of the parent ACC. The ASCC property consists of a property term that expresses the nature of an association between object classes, and the name of the object class being associated. The associated object class is a complex data type that expresses the value domain of the ASCC. The ASCC property is reusable across object classes, but once it has been given the object class of a parent ACC, it creates an ASCC that is unique to the object class to which it is assigned. In CCTS constructs, ASCCs are equivalent to UML associations of **`AggregationKind=shared`**.[2,3]

[Example] – Association Core Component



[Example] – Association Core Component



The example shows two ACCs in a non-UML diagram:

- **`Contract. Details`**
- **`Period. Details`**

Each ACC has a number of properties.

[2] UML Association – A UML association defines a relationship between classes of objects. UML associations include AggregationKind=shared and AggregationKind=composite.

[3] UML Aggregation – An aggregation is a special form of UML association that specifies a whole-part relationship between the aggregate (whole) and a component part.

[Example] – Association Core Component (Continued)

The ACC **Period. Details** has:

- two simple BCC Properties:
  - **Duration. Measure**
  - **Start. Date Time**
- no complex ASCC Properties

The ACC **Contract. Details** has:

- five simple BCC properties:
  - **Identification. Identifier**
  - **Issue. Date Time**
  - **Type. Code**
  - **Price. Amount**
  - **Item. Quantity**
- two complex ASCC properties:
  - **Effective. Period**
  - **Performance. Metrics**

The simple properties are BCC properties. They represent a singular characteristic and their set of allowed values is defined by a CDT.

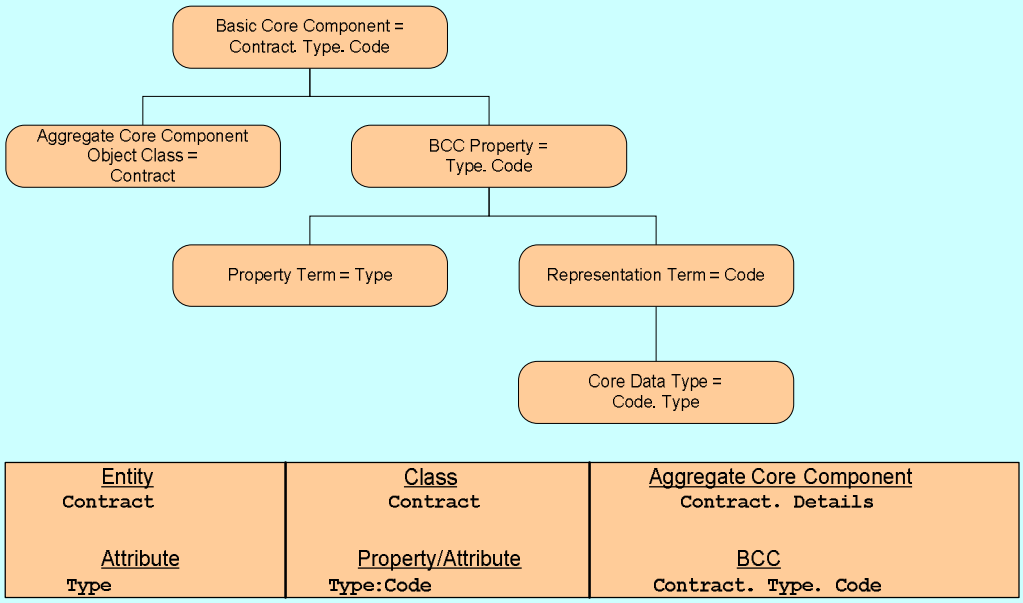
The complex properties are ASCC properties. Their structure is defined by another ACC. For example, the structure of **Contract. Effective. Period** is described by **Period. Details.**

Since CCTS is a semantic model, it is necessary to represent the associations as part of the content of the associating Contract class. Thus, the ASCC as represented by the ASCC property is actually contained in the content model of the associating Contract. Details ACC.

5.1.3 Basic Core Component

A Basic Core Component (BCC) represents a property of an ACC. A BCC consists of a BCC property plus the object class of the parent ACC. The BCC property is reusable across object classes, but once it has been given the object class of a parent ACC, it becomes a BCC that is unique to the object class to which it is assigned. In data modeling terms, a BCC is the equivalent of an entity attribute or class property (See section 5.6).

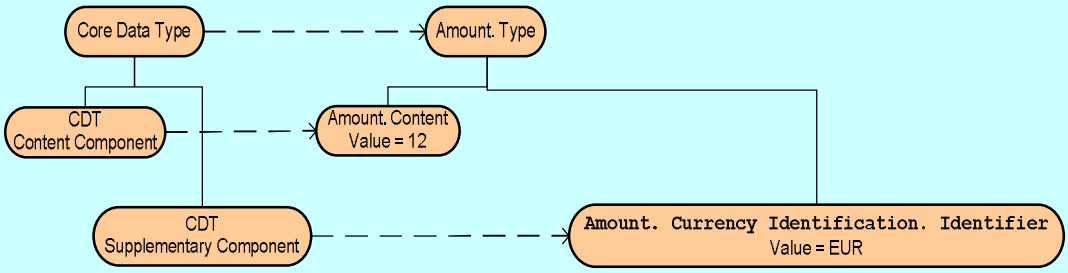
[Example] – Basic Core Component



5.2 Core Data Types

As identified in ISO 11179, a data type constitutes the value domain for the allowed values for a property. For CCs this data type is called a Core Data Type (CDT).

[Example] – Core Data Types



For the CDT **Amount. Type**, the primitive is decimal. A CDT has a content component and one or more supplementary components. The CDT content component **Amount. Content** carries the value of **12**. This value has no meaning on its own. But **12 EUR** has a meaning: **EUR** represents the Euro currency and is the value of the supplementary component **Amount. Currency Identification. Identifier** defined for the CDT **Amount. Type.** The supplementary component adds essential extra information about the content component and gives extra meaning to the CDT value domain.

A CDT represents the full range of values that can be used for representing an instance of a BCC. Every CDT has a content component and zero or more supplementary components. Every CDT content and supplementary component has a value domain. These value domains can be expressed by either a primitive type or a scheme or list. The value domain of the CDT is defined by the value domain (set of permissible values) for the CDT content component (the actual value of the data element) and the value domain of the additional constraints

expressed by the supplementary components. Supplementary components give meaning to the value domain by adding essential extra information about the content component. The number of defined supplementary components varies by CDT, and is determined by the number of attributes necessary to fully define the value domain of the CDT.

CDTs form the bedrock for interoperability of CC's. UN/CEFACT defines a formal set of CDTs as part of the overall CCTS standards stack. Other users are encouraged to adopt the UN/CEFACT DT catalogue to ensure maximum interoperability across implementations.[4]

5.3 Business Information Entities

Core components act as conceptual models that are used to define Business Information Entities (BIEs). BIEs are the expression of the conceptual core components as logical data model objects and information exchanges. BIEs are created through the application of context and may be qualified to guarantee unique business semantics. A BIE may specify a restricted form of its underlying CC. The structure of CCs and BIEs are complementary in many respects.

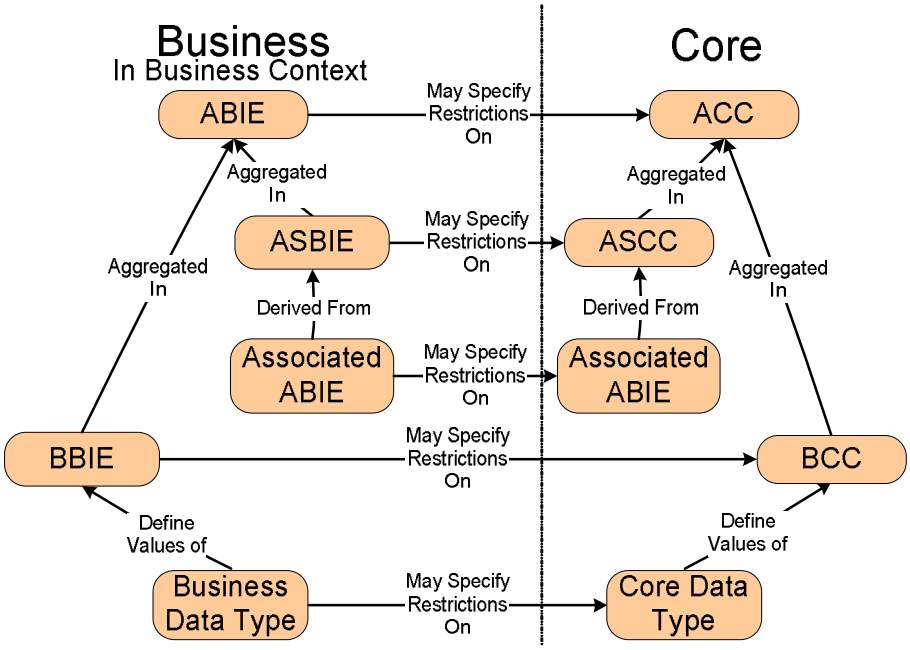


Figure 5-2. Relationships Between Core Components and Business Information Entities

The relationships between CCs and BIEs are described in Figure 5-2. The key differentiator between CCs and BIEs is the concept of business context. Business context is a mechanism for categorizing and refining CCs according to their use in a particular data model or business circumstance. In CCTS, business context is formally described for specific business circumstances for each BIE. This is

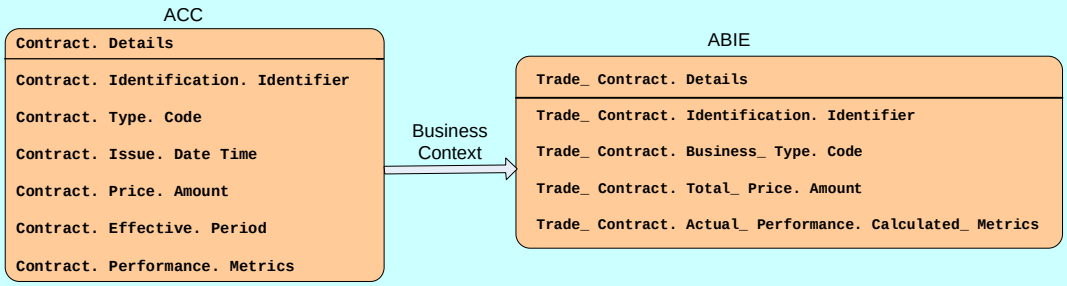
[4] Approved CDTs and their corresponding data type terms, representation terms, allowed restrictions, and supplementary components are published by the UN/CEFACT in the Data Type Catalogue.

accomplished by assigning values to a set of context categories (See Section 9). Once these business contexts are identified, BIEs can be defined to take into account any necessary qualification and refinement needed to support the use of the underlying CC in the given business context.[5]

5.3.1 Aggregate Business Information Entity

An Aggregate Business Information Entity (ABIE) is derived from an ACC and refines the business semantic definition for a specific business context. Just as an ACC is the representation of an object class, so too are its derived ABIEs. An ABIE may be qualified at the object class level, and its properties may be qualified at the property term level. The ABIE can reflect restrictions on the content model of the ACC through:

[Example] – Aggregate Business Information Entity with context driven restrictions and qualifications



For the ABIE **`Trade_ Contract. Details`**, business context has been applied to the ACC of **`Contract. Details`**. This context has resulted in qualification of the object class, qualification of selected property terms, and restriction on the content model.

- Restrictions on the cardinality of the BCCs and ASCCs as shown in Figure 5-3.
- Use and non-use of individual BCCs and ASCCs
- Qualification of individual ASCC and BCC properties
- Restrictions on the content model of an associated ACC for an ASCC
- Restrictions on the data type of the BCC
- Restrictions on the concept or conceptual domain of the ASCC or BCC property as reflected in the definition and usage rules.

BIE cardinality does not define how many BIEs may be derived from a source CC. Rather it describes the allowed occurrences of a specific BIE. For example, If a BCC or ASCC has the cardinality [0..1], the derived BBIE or ASBIE may have the same cardinality [0..1], or it may have a restricted cardinality of [1..1]. If a BCC or ASCC has the cardinality [0..*], the derived BBIE or ASBIE may have the same

[5] The *Core Components' Context* mechanism provides the more detailed linkage between specific business data and the exact circumstances of its business use.

cardinality [0..*], or it may have a restricted minimum occurrence or maximum occurrence.

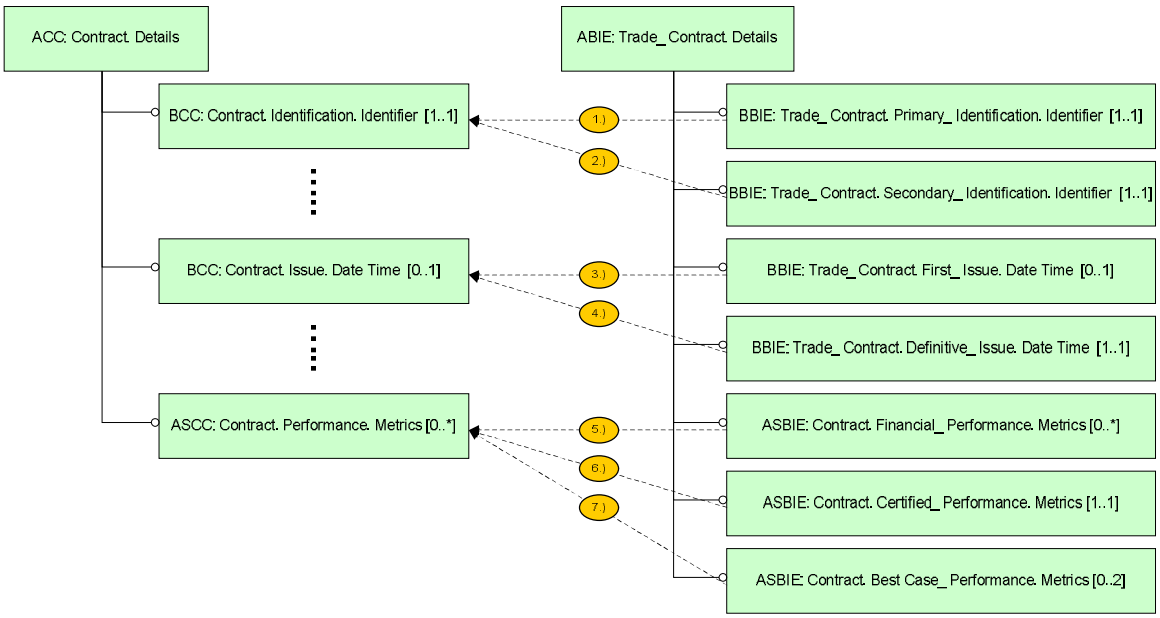


Figure 5-3. BIE Cardinality

ASCC and BCC properties may have different qualifiers applied. This may result in the ABIE having a greater number of qualified properties than its corresponding ACCs unqualified properties. This is still considered a restriction since each BIE property represents a restriction to its corresponding core component property. ASCC and BCC properties may also have multiple qualifiers applied. Multiple qualifiers create a qualifier hierarchy, with each additional qualifier reflecting a further restriction to its less qualified BIE property.

[Example] – Use of Qualifiers

The Multi-qualified ABIE

Electronic_ Trade_ Contract. Details

qualifies the qualified ABIE

Trade_ Contract. Details

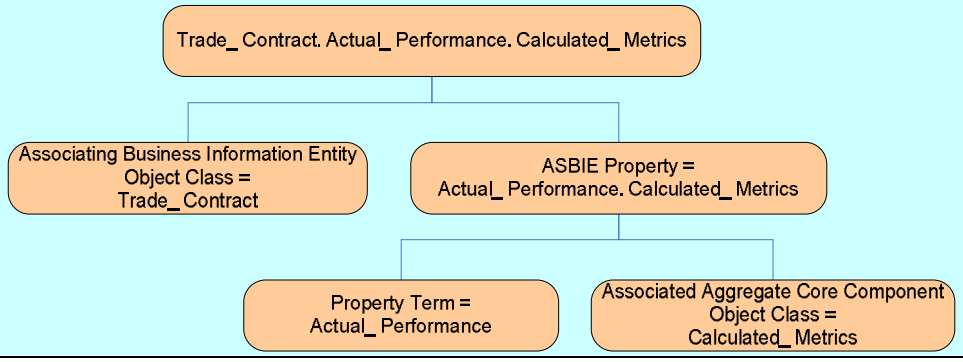
which qualifies the ACC

Contract. Details

5.3.2 Association Business Information Entity

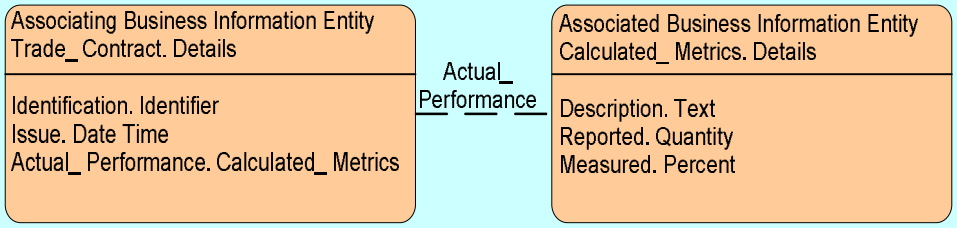
An Association Business Information Entity (ASBIE) is a BIE that represents a complex property of an ABIE. An ASBIE has the structure of, and represents another ABIE. An ASBIE is based on an ASCC, but exists in a business context. As its source ASCC, an ASBIE consists of an ASBIE property plus the object class of the parent ABIE. The ASBIE property is reusable across object classes, but once it has been given the object class of a parent ABIE, it creates an ASBIE that is unique to the object class to which it is assigned.

[Example] – Association Business Information Entity



ASBIEs can be either a UML association of **`AggregationKind=shared`** or **`AggregationKind=composite`**.[6]

[Example] – Aggregation Representation of an Association Business Information Entity



The example shows two ABIEs:

- **`Trade_ Contract. Details`**
- **`Calculated_ Metrics. Details`**.

Each ABIE has a number of properties (i.e. business characteristics).

The ABIE **`Calculated_ Metrics. Details`** has:

- three simple properties:
  - **`Description. Text`**
  - **`Reported. Quantity`**
  - **`Measured. Percent`**
- no complex properties

The ABIE **`Trade_ Contract. Details`** has:

- two simple properties:
  - **`Identification. Identifier`**
  - **`Issue. Date Time`**

[6] Composition – A composition is a strong form of aggregation association that requires that the component part only belongs to a single parent object, and only exists as long as that parent object exists.

[Example] – Aggregation Representation of an Association Business Information Entity (Cont'd)

- one complex property:
  - **Actual_ Performance. Calculated_ Metrics**

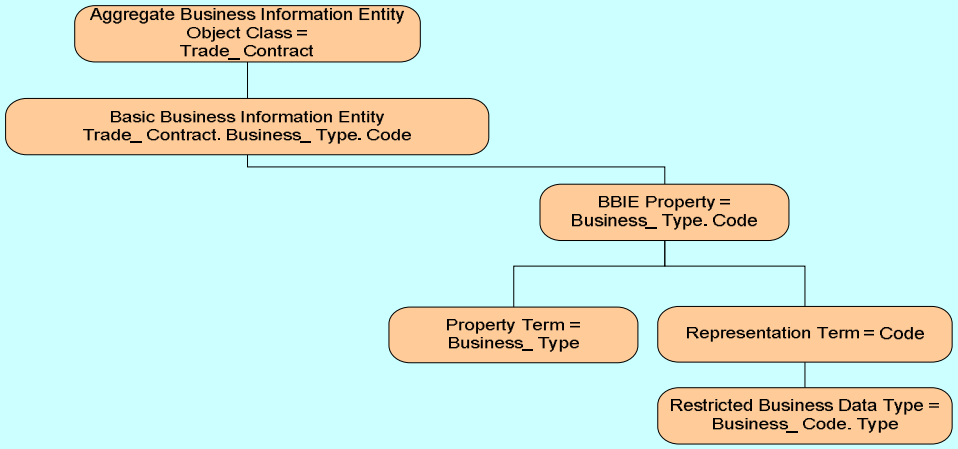
The simple properties are BBIEs. They represent a singular business characteristic and their set of allowed values is defined by a BDT.

The complex property is an ASBIE. It represents a complex business characteristic and its structure is therefore defined by another ABIE. The structure of **Actual_ Performance. Calculated_ Metrics** is described by **Calculated_ Metrics. Details.**

5.3.3 Basic Business Information Entity

A Basic Business Information Entity (BBIE) is a BCC used in a specific business context. Multiple BBIEs can be derived from a single BCC. A BBIE has a unique business semantic definition. A BBIE is created from a BBIE property and inherits the object class of the parent ABIE. The BBIE property is reusable across object classes. In data modelling terms, a BBIE is the equivalent of an entity attribute or class property (11179)/attribute (UML) (see section 5.7).

[Example] – Basic Business Information Entity



Basic Core Component

Contract. Type. Code

Basic Business Information Entity

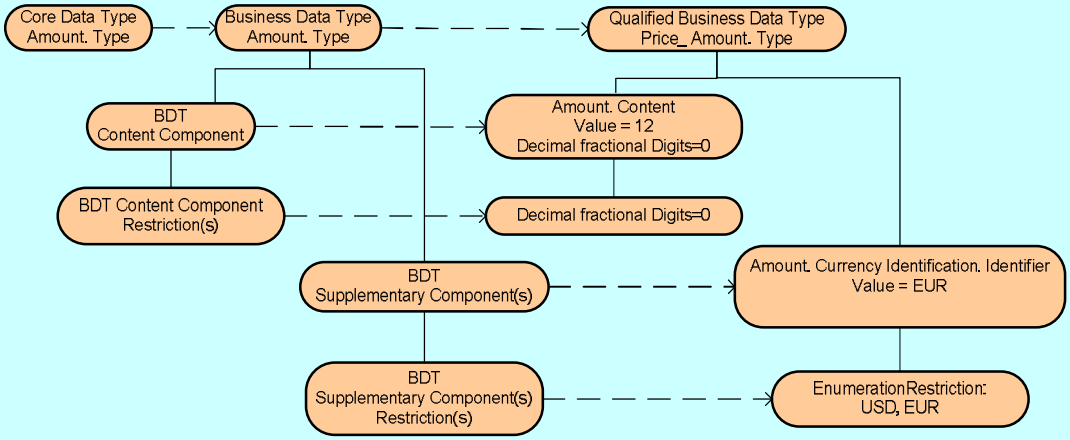
Trade_ Contract.Business_ Type. Code

Every BBIE property is derived from a BCC property. Like their BCC property counterparts, BBIE properties are reusable across object classes, but once it has been given the object class of a parent ABIE, it becomes a BBIE that is unique to the object class to which it is assigned. Each BBIE property has a Business Data Type (BDT) that describes its value domain. BBIE BDTs are derived from the CDT of the BCC.

5.4 Business Data Types

For every approved CDT, a corresponding unrestricted BDT will be created. This BDT will have no restrictions of the set of values of its source CDT's content component or supplementary components. Additional business data types may also be created that include restrictions of the set of values of its source CDT's content component and/or supplementary component(s). The restrictions represent a qualification of the BDT similar to the qualification of BIEs. Both the content component and supplementary component(s) have allowed component restrictions that provide all information necessary to understand the value domain for a specific BBIE.

Example – Business Data Type with BDT Content Component and BDT Supplementary Component Restrictions



The qualified BDT **`Price_ Amount. Type`** is derived from the **`BDT Amount. Type`** **`which is derived from the CDT Amount. Type.`**

For the BDT **`Price_ Amount. Type`**, the primitive is decimal. The BDT qualifier **`Price`** semantically conveys the data type value domain restrictions being applied to the source BDT **`Amount. Type`** for its specific use as the value domain for a type of payment.

The BDT has both BDT content and BDT supplementary components.

In the example, the BDT content component **`Amount. Content`** carries the value of 12. The allowed value range for the content component has been restricted using the BDT content component restriction **`Decimal Fractional Digits = 0`**.

In the example, the BDT **`Amount. Currency Identification. Identifier`** supplementary component carries the value **`EUR`**, where **`EUR`** represents the Euro currency. The BDT **`Amount. Currency Identification. Identifier`** supplementary component has been restricted using the enumeration component restriction to allowed values **`USD`** or **`EUR`**.

In addition to allowed component restrictions, BDTs may restrict the content model (only use a subset) of the allowed supplementary components from its source CDT. Restricted BDTs may be further restricted in hierarchical fashion through additional, more restrictive, content and/or supplementary component restrictions.

5.5 Relationship between ISO 11179 Data Element Concepts and Core Components Constructs

There is a direct relationship between the constructs of CCTS and those of ISO 11179. As shown in figure 5-3, the ISO 11179 data element concept consists of object class, property term, and representation term. The representation term, combined with a property term, constitutes a generic data element. This generic data element is the equivalent of BCC properties and BBIE properties. In ISO 11179, these generic data elements are reusable across object classes, and inherit the name of the object class in which they occur. Similarly, in CCTS, these properties are reusable to create BCCs or BBIEs in multiple ACCs and ABIEs. However, once a property is used to create a BCC or BBIE, to include refinements to the value domain at the BBIE level, the BCC or BBIE creates an unchangeable part of the ACC or ABIE to which it belongs. This concept has been extended in CCTS to include ASCC and ASBIE properties as well.

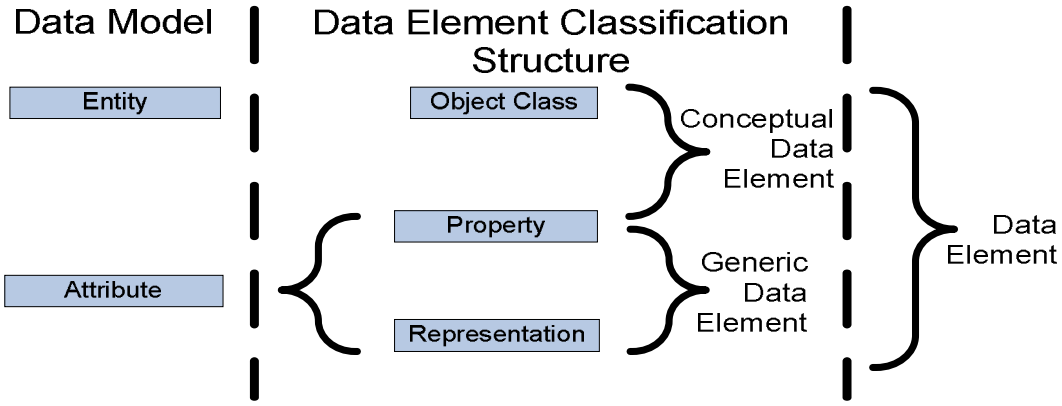


Figure 5-3. ISO 11179 Data Element Model

The ISO 11179 object class and property term constitute a conceptual data element. These conceptual data elements do not have a specific representation (value domain), and are reusable by applying different representations that create conceptually similar but distinct data elements. This concept is not currently included in the CCTS metamodel, but can be accommodated by implementers who choose to maintain such constructs in a registry.

The ISO 11179 object class, property term, and representation term together constitute a data element. These data elements are the equivalent of BCCs, ASCCs, BBIEs and ASBIEs. In ISO 11179 and UML, these data elements (classes) are unique in their occurrence, but can be associated with other object classes through UML association. When such UML associations of object classes occur, they are instantiated as ASCCs and ASBIEs in the CCTS model.

5.6 Relationship between UN/CEFACT Modelling Methodology and Core Components Constructs

UN/CEFACT has developed the *UN/CEFACT Modelling Methodology* (UMM).The UMM base and foundation modules define a UML profile for modeling choreographies of business collaborations and their business document

exchanges.[7] The UMM is the recommended business process methodology for developing CCTS artefacts.

[7] The UN/CEFACT Modelling Methodology (UMM) is a methodology for modelling collaborative *Business Processes* that is based on the Object Management Group's Unified Modelling Language.

6 Core Component Model

This section provides a detailed technical explanation of the core component metamodel as seen in the UML diagram figure 6-1.

> Note – Models
>
> Models are UML conformant figures and are normative to the level of detail at which they exist.

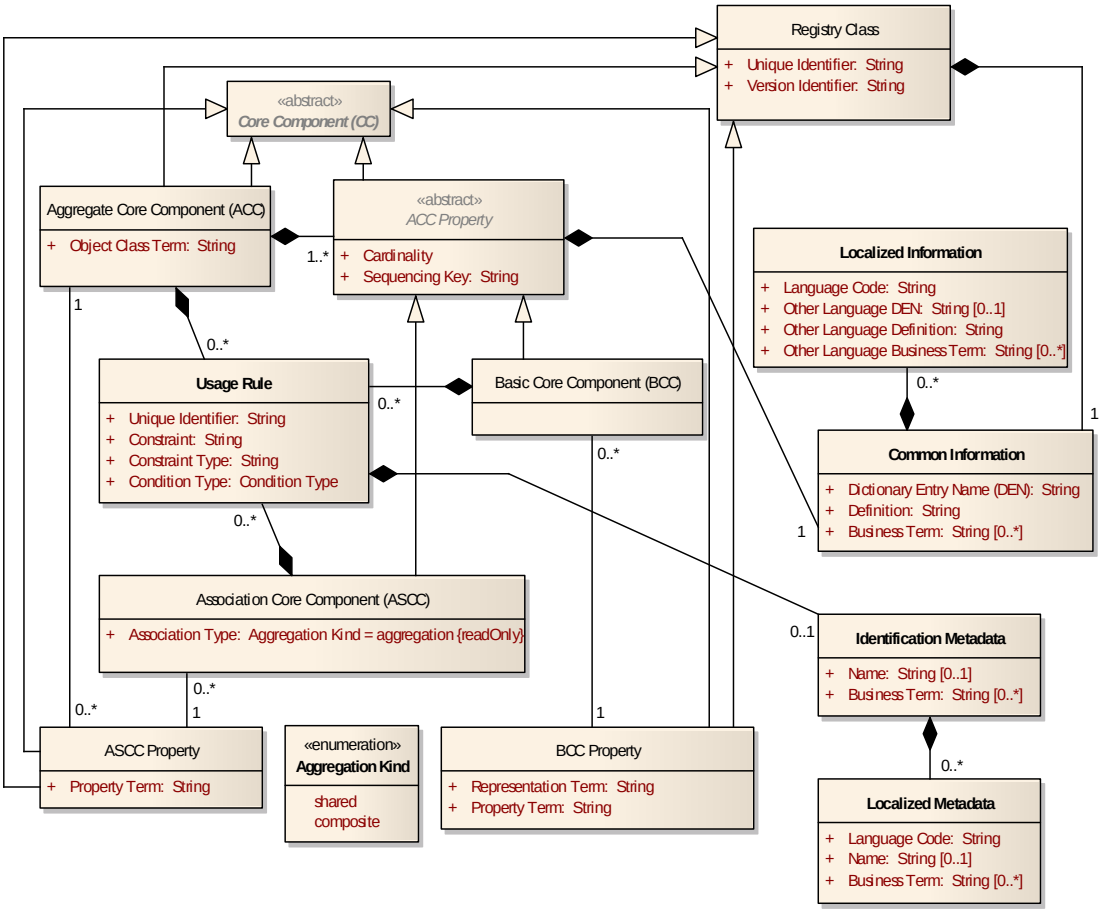


Figure 6-1. UML Diagram of Core Component Basic Definition Model

6.1 Overview

A core component is a building block for the development of a semantically correct and meaningful business information exchange 'parcel' containing the information pieces needed to describe a specific concept.

> [Definition] – Core Component (CC)
>
> A core component is a semantic building block for creating clear and meaningful data models, vocabularies, and information exchange packages. Core components are used as the basis for creating business information entities.

There are five categories of Core Components (CCs):

- Aggregate Core Component (ACC)
- Association Core Component (ASCC)
- Basic Core Component (BCC), and
- ASCC Property
- BCC Property

1 A CC shall be an ACC, ASCC, BCC, ASCC property, or BCC property.

[Note] – ACC Property

An ACC property is a generalization of a BCC or an ASCC, and not a property in its own right.

ACCs, ASCCs, BCCs, ASCC properties, and BCC properties are collectively called CCs and are typically stored in a registry, database, or other mechanism to maximize their reuse.

6.2 Core Component Naming and Definition Conventions

A naming convention is necessary to gain consistency in the naming and defining of all CCs. The resulting consistency facilitates comparison during the discovery and analysis process, and precludes ambiguity, such as the development of multiple CCs with different names that have the same semantic meaning.

The CC naming and definition conventions are derived from the guidelines and principles described in *ISO 11179 Part 4 – Definitions* and *ISO 11179 Part 5 – Naming and Identification Principles*. In certain instances, these guidelines have been adapted to the overall CC environment. In particular, the guidelines have been extended to cover the naming and defining of all CCs defined in this standard.

The required language for CCs is English. CC discovery work may very well occur in other languages; however official submissions for inclusion in a component library must be in English. In order to ensure absolute clarity and understanding of the names and definitions it is essential to use terms from the *Oxford English Dictionary*. A supplementary controlled vocabulary should be developed to identify the definition to be used for any terms that are potentially ambiguous. This controlled vocabulary shall also be used to identify the preferred term in cases where more than one term or spelling might be used to cover the same definition. The controlled vocabulary will also contain terms not found in the *Oxford English Dictionary*. This will ensure that each term within any of the names and definitions is used in a consistent and unambiguous way. The resultant semantic integrity will also mean that translation into other languages retains the precise original meaning.[8]

[Note] – Oxford English Dictionary

The *Oxford English Dictionary* and a controlled vocabulary, if used, will be the authoritative source for conflict resolution between competing spellings of component names or definitions.

[8] Implementers are encouraged to use the UN/CEFACT controlled vocabulary as the authoritative source for terms and definitions.

6.3 Core Component Registry Class

A core component registry class represents a cohesive set of information associated with a single CC.

[Definition] – Registry Class

A registry class is the formal definition of all the common information necessary to be recorded in the registry by a registry artefact – a core component, a business information entity, or a data type.

Each ACC, ASCC property, and BCC property, are registry classes.

[Note] – Registry Class

Although the term registry class is used, no normative requirement exists to actually use a registry. Other storage mechanisms such as data bases may also be used to uniquely store registry classes and their associated subordinate classes.

[C2] A registry class shall be created for each ACC, ASCC property, and BCC property.

Each CC registry class contains the following information:

- Unique Identifier
- Unique Version Identifier

[Note] – CC Identifier Structure

There are no specific rules for the structure of the CC identifiers; however the preferred identification scheme is the ITU-T Rec. X.667|ISO/IEC9834-8 Universally Unique Identifier (UUID) scheme. Implementers are free to use this scheme, or choose any other structure scheme, providing it guarantees uniqueness within the library to which it belongs.

6.4 Core Component Common Information

The CC common information class provides necessary component information that is applicable to components either directly or through inheritance. The CC common information class contains the following information:

- **DEN** – the unique official name of the CC in the dictionary.
- **Definition** – the unique business semantic meaning of the CC.
- **Business Term(s)** – a synonym term under which the component is commonly known and used in business. A CC may have several business terms.

6.4.1 Core Component Dictionary Entry Name

CC naming rules are based on the following concepts as defined in ISO 11179:

- **Object Class** – represents the logical data grouping or aggregation (in a logical data model) to which a property belongs. The object class is represented by an object class term. Thus, the object class is the part of a CCs DEN that represents an activity or object. Object classes

have explicit boundaries and meaning and their properties and behaviour follow the same rules.

- **Object Class Term** – a part of the dictionary entry name of a component which represents the object class to which it belongs
- **Property Term** – represents a distinguishing characteristic of the object class and shall occur naturally in the definition.
- **Representation Term** – an element of the component name which describes the form in which the component is represented.

[C3] All terms used in CC DENs shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[C4] A CC DEN shall be unique amongst all DENs within the library of which it is a part.

[C5] A CC DEN shall be extracted from the CC definition.

[C6] A CC DEN shall not include consecutive identical words or terms.

[C7] A CC DEN and all its components shall be in singular form unless the concept itself is plural.

[C8] A CC DEN shall only use alphabetic characters plus the dot and space characters.

[C9] A CC DEN shall only contain verbs, nouns, adverbs and adjectives unless a different part of speech is part of an official title, part of a term listed in the *Oxford English Dictionary*, or part of a controlled vocabulary.

[Note] – Parts of Speech

Articles, prepositions and related parts of speech that are not verbs, nouns, adverbs and adjectives normally add no semantic clarity and should not be used unless as part of an official title or in a controlled vocabulary as part of a common business term that can not otherwise be expressed.

[Example] – Exception use of Preposition

Goods Item. Free On Board Value. Amount

Where **`Free On Board`** is a globally recognized term and integral part of the property term for this BCC.

[C10] Abbreviations and acronyms that are part of the CC DEN shall be expanded or explained in the definition and should be added to a controlled vocabulary if it exists.

[C11] CC DEN object class terms, property terms, and representation terms shall be separated by a dot and a space character.

[C12] The space character shall separate words in multi-worded CC object class, property, and representation terms.

[C13] Each word in a CC DEN shall start with a capital letter.

[C14] The dots after CC object class and property terms shall be followed by a space character.

[C15] Each CC DEN shall remain unique when its separators are removed.

6.4.2 Core Component Definitions

CC definitions are based on the requirements for data element definitions defined in ISO 11179-4.

[C16] Each CC shall have its own unique semantic definition within the library of which it is a part.

[Note] – Order of Development of Definition and DEN

In the interest of quality, it is recommended that the CC definition be developed first and the DEN extracted from it.

[C17] The CC definition shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[C18] The CC definition shall be consistent with the requirements of ISO 11179-4.

[C19] The CC definition shall take into account the fact that the users of the CC library are not necessarily native English speakers. It shall therefore contain short sentences, using common terms. Wherever alternative terms exist for a word in the definition, the definition shall use the preferred term as identified in the *Oxford English Dictionary* or a controlled vocabulary if one exists.

[Example] – Use of Synonym Terms

ACC – Contract Line Item. Details

Definition – A **`contract line item`** is a distinct, separately defined line item specified in a contract.

Alternative terms for distinct include distinctive and distinguishable. The term **`distinct`** is a more easily understood common term for differentiating a contract line item amongst a group of line items.

[C20] Whenever both the definite (i.e. **`the`**) and indefinite article (i.e. **a**) are possible in a definition, preference shall be given to an indefinite article (i.e. **a**).

[Note] – Definition Quality

To verify the quality of the definition, place the DEN followed by is before the definition to ensure that it is not simply a repetition of the DEN.

6.4.3 Core Component Business Terms

CC business terms are those terms commonly used for day-to-day information exchanges within a given domain. As such, no specific rules apply to business term structures.

6.5 Core Component Localized Information Class

While the normative expressions of components are in the English language, implementers may choose to create alternative language expressions of definitions and business terms. The CC localized information class contains the relevant information necessary to associate the native language expressions to their normative English language counterparts. Other language CC DENs will only consist of alphabetic, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules. In addition to other language DEN, definition, and business term(s), a mandatory language code identifies the language in which the components are being expressed for storage in the registry. The localized information class contains:

- **Language Code –** A code which identifies the language being used. *Tags For the Identification of Languages*, Internet Engineering Task Force (IETF) RFC 3066 of January 2001 will be used as the authoritative source for code values.
- **DEN –** The official name of the component in a language other than English.
- **Definition –** the semantic meaning of the component in a language other than English.
- **Business Term –** A synonym term in another language under which the component is commonly known and used in a business expression in that language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language and locale code. The business terms must only be expressed in the language identified by the language and locale code, or a recognized dialect of the language.

6.6 Aggregate Core Components

Each ACC represents the logical data grouping or aggregation in a logical data model.

[Definition] – Aggregate Core Component (ACC)
An aggregate core component is a collection of related pieces of business information that together convey a distinct business meaning, independent of any specific business context. Expressed in modelling terms, it is the representation of an object class, independent of any specific business context.

6.6.1 Aggregate Core Component Object Class Term

The ACC object class is represented by an object class term that serves as the basis for the DEN of the ACC and for the DEN of all BCCs and ASCCs that are properties of the ACC.

[C21] The ACC object class term shall be unique amongst the set of object class terms in the library of which it is a part.

Object class terms may consist of more than one word.

[C22] Multi-worded ACC object class terms shall have a unique meaning separate from the individual words used independently or in any other combination.

[Example] – Single versus Multi-Worded Object Class Terms

`Currency Exchange. Details` does not represent the same thing as **`Currency. Details`**

`Currency Exchange. Details` does not represent the same thing as **`Exchange. Details`**

Thus the term **`Currency Exchange`** represents a different object than either **`Currency`** or **`Exchange`**.

6.6.2 Aggregate Core Component Usage Rule

ACCs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the ACC. ACC usage rules constrain the specific application of an ACC in its role as an object class. ACC usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[C23] An ACC shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[C24] ACC usage rules shall not replicate BCC, ASCC, or CDT usage rules.

[C25] An ACC usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The ACC usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[C26] An unstructured ACC usage rule constraint shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal constraint language such as the UML Object Constraint Language (OCL) or Object Management Group (OMG) Semantics of Business Vocabulary and Usage rules (SBVR).

[C27] A structured ACC usage rule constraint shall have a formal constraint language expression.

ACC usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[C28] Every ACC usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

ACC usage rules will also have a condition type that identifes when the constraint should be applied.

[C29] Every ACC usage rule shall have a condition type.

[C30] Every ACC usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

[Example] –Constraint and Condition Type

Constraint Type – OCL

Condition Type – pre-condition

6.6.2.1 Aggregate Core Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule from all other usage rules for a library, an ACC usage rule may also have a identification metadata class that provides additional information.

[C31] An ACC usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates the usage rule from all other named usage rules for the ACC.

[C32] An ACC usage rule shall have zero or one names that is unique within the group of usage rules of an ACC.

The ACC usage rule metadata class may contain several business terms. ACC usage rule business terms are synonym terms under which the ACC usage rule is commonly known and used in business.

[C33] Each ACC usage rule shall have zero or more business terms.

6.6.2.2 Aggregate Core Component Usage Rule Localized Metadata

ACC usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[C34] An ACC usage rule shall have zero or more localized metadata classes.

[C35] Each occurrence of an ACC usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[C36] ACC usage rule localized metadata shall be in the language identified by the language and locale code.

6.6.3 Aggregate Core Component Identifiers

Every ACC is a registry class. In order to ensure uniqueness, every ACC will have assigned a:

- **Unique Identifier:** The identifier that references an ACC in a unique and unambiguous way.
- **Version Identifier:** An indication of the evolution over time of an ACC.

[C37] Each ACC shall have a unique identifier within the library of which it is a part.

[C38] Each version of an ACC shall have a unique version identifier within the library of which it is a part.

6.6.4 Aggregate Core Component Common Information

[C39] Each ACC shall have a common information class.

[C40] The ACC common information class shall conform to all CC common information rules.

[C41] The ACC common information class shall consist of:

- **DEN (mandatory):** The official name of the ACC.
- **Definition (mandatory):** The semantic meaning of the ACC.
- **Business Term (optional, repetitive)**: A synonym term under which the ACC is commonly known and used in business.

[Example] – ACC Common Information

DEN – **`Contract. Details`**

Definition – A contract is an agreement between two or more parties, especially one that is written or spoken and enforceable by law.

Business Term – **`Purchasing Agreement`**

6.6.4.1 Aggregate Core Component Dictionary Entry Names

[C42] Each ACC DEN shall conform to all CC DEN rules.

[C43] The DEN of an ACC shall consist of a meaningful object class term followed by a dot, a space character, and the term **`Details`**.

The object class term may consist of more than one word.

[Example] – DEN for ACCs

`Contract. Details`

`Metrics. Details`

6.6.4.2 Aggregate Core Component Definitions

[C44] Each ACC definition shall conform to all CC definition rules.

[C45] Each ACC definition shall include the object class term.

[Example] – ACC Definition

`Contract. Details`

A **`contract`** is an agreement between two or more parties, especially one that is written or spoken and enforceable by law.

6.6.4.3 Aggregate Core Component Business Terms

An ACC may have several business terms. ACC business terms are synonym terms under which the ACC is commonly known and used in business.

[C46] Each ACC shall have zero or more business terms.

6.6.5 Aggregate Core Component Localized Information

The ACC localized information class contains the relevant information necessary to associate native language expressions of ACC attributes to the ACC.

[C47] Each ACC shall have zero or more localized information classes.

[C48] Each occurrence of an ACC localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the ACC in a language other than English.
- **Definition (mandatory):** The semantic meaning of the ACC in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ACC is commonly known and used in a business expression in that language.

ACC localized information DENs should follow, as much as possible, all ACC DEN rules.

[C49] Each ACC localized information DEN shall only consist of alphabetic, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[C50] Each ACC localized information definition shall adhere to all ACC definition rules other than the requirement to be in the English language.

[C51] ACC localized information DENs and definitions shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[C52] Each ACC localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

6.7 Aggregate Core Component Properties

An ACC consists of ACC properties. The ACC property is a generalization of either an ASCC or a BCC. Every ACC contains at least one ACC property.

[C53] An ACC shall contain at least one ACC property.

[C54] An ACC property shall be either a BCC or an ASCC.

[Definition] – ACC Property

An ACC property is a unique property of the aggregate core component that must be related to the concept of the aggregate core component. An ACC property is either an ASCC or a BCC.

[C55] Within an ACC, all embedded BCCs and ASCCs shall be related to the concept of the aggregate.

ACC properties must be unique within the ACC.

[C56] An ASCC and a BCC DEN shall never be identical when used in an ACC.

An ACC property that is an ASCC must be devoid of mandatory circular references.

[C57] An ACC shall never contain –directly or at any nested level – a mandatory ASCC whose associated ACC is the same as the top level ACC.

[Note] – Recursion

The objective of rule C48 is to prevent endless nesting in component models. The rule does not prevent an ACC containing an ASCC reference to itself. However, it must be optional, making it possible to stop the nesting at a finite number of levels.

6.8 Association Core Components

Association core components associate two ACCs, where the associated ACC is the property of the associating ACC. The property term represents the role of the associated ACC in the association. ASCCs have a defined minimum and maximum occurrence. ASCCs are always UML associations of **`AggregationKind=shared`**.

[Definition] – Association Core Component (ASCC)

An association core component is a core component which defines the role of a specific aggregate core component (known as the associated ACC) associated to another aggregate core component (known as the associating ACC). An association core component functions as an ACC property of the associating ACC.

6.8.1 Association Core Component Association Type

ASCCs represent an association between the associating (parent) ACC and the associated (child) ACC. The associated ACC will exist regardless of the state of the associating ACC, therefore the nature of the association of all ASCCs is as a UML association **`AggregationKind=shared`**. An association type indicator is required to reflect this association as a mechanism for transformation between alternative syntax storage expressions and UML representation.

[C58] An ASCC shall have an UML association **`AggregationKind`** value of **`shared`**.

6.8.2 Association Core Component Usage Rule

ASCCs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the ASCC. ASCC usage rules clarify (or constrain) the use of an ASCC as an ACC property. ASCC usage rules can be either

unstructured – expressed as free form text , or structured – expressed in a formal language.

[C59] An ASCC shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[C60] ASCC usage rules shall not replicate ACC, BCC, or CDT usage rules.

[C61] An ASCC usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The ASCC usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[C62] An unstructured ASCC usage rule constraint shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal constraint language such as the UML OCL or OMG SBVR.

[C63] A structured ASCC usage rule constraint shall have a formal constraint language expression.

ASCC usage rule constraint types must also be specified. The constraint type value is taken from a formal constraint type code list.

[C64] Every ASCC usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

ASCC usage rules expressed as formal constraints will also have a condition type that identifies when the constraint should be applied.

[C65] Every ASCC usage rule shall have a condition type

[C66] Every ASCC usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**..

6.8.2.1 Association Core Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given ASCC from all other usage rules for a library, an ASCC usage rule may also have an identification metadata class that provides additional information.

[C67] An ASCC usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates the usage rule from all other named usage rules for the ASCC.

[C68] An ASCC usage rule shall have zero or one names that is unique within the group of usage rules of an ACC.

The ASCC usage rule identification metadata class may contain several business terms. ASCC usage rule business terms are synonym terms under which the ASCC usage rule is commonly known and used in business.

[C69] Each ASCC usage rule shall have zero or more business terms.

6.8.2.2 Association Core Component Usage Rule Localized Metadata

ASCC usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[C70] An ASCC usage rule shall have zero or more localized metadata classes.

[C71] Each occurrence of an ASCC usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[C72] ASCC usage rule localized metadata shall be in the language identified by the language and locale code.

6.8.3 Association Core Component Cardinality

Each ASCC, in its role as an ACC property, will have its cardinality explicitly expressed.

[C73] Each ASCC shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[C74] ASCC cardinality values shall be non-negative integers of zero or greater, or – only in the case of maximum cardinality – the token **unbounded** if no limit applies.

6.8.4 Association Core Component Sequencing Key

Software and storage applications may have unique sequencing algorithms that change the normatively defined order of the ASCC within an ACC. To ensure the desired order is preserved, each ASCC within an ACC will be assigned a unique sequencing key only for the aspects of presentation.

[C75] Each ASCC shall be assigned a unique sequencing key within the ACC of which it is a part.

[Note] – Sequence Key Structure

There are no specific rules for the structure of the sequencing keys. Implementers are free to choose any structure providing it guarantees uniqueness within the ACC to which it belongs and the structuring scheme is readily available for anyone accessing or using the ACC.

6.8.5 Association Core Component Common Information

In its role as an ACC property, each ASCC has a common information class.

[C76] Each ASCC shall have a common information class.

[C77] The ASCC common information class shall conform to all CC common information rules.

[C78] The ASCC common information class shall consist of:

- **DEN (mandatory):** The official name of the ASCC.
- **Definition (mandatory):** The semantic meaning of the ASCC.
- **Business Term (optional, repetitive)**: A synonym term under which the ASCC is commonly known and used in business.

[Example] – ASCC Common Information

DEN – **`Contract. Effective. Period`**

Definition – A period within which the provisions of this contract are, or will be, in force or effective.

Business Term – **`Contract Duration`**

6.8.5.1 Association Core Component Dictionary Entry Names

[C79] Each ASCC DEN shall conform to all CC DEN rules.

[C80] The DEN of an ASCC shall consist of the following components in the specified order:

- The object class term of the associating ACC, followed by a dot and space character.
- The DEN of the included ASCC property.

[Example] - ASCCs

`Contract. Effective. Period` where the associated ACC **`Period. Details`** now becomes a property of the associating ACC of **`Contract. Details`** and the nature of that association is **`Effective.`**

6.8.5.2 Association Core Component Definitions

[C81] Each ASCC definition shall conform to all CC definition rules.

[C82] The definition of an ASCC shall include the object class term of the associating ACC, and the definition of the ASCC property the ASCC includes.

[Example] – ASCC Definition

`Contract. Effective. Period`
A **`period`** within which the provisions of this contract are, or will be, in force or **`effective`**. It constitutes a specific period of time such as the length of time between two known date/time points, from a start date onwards, or up to an end date that constitutes an effective period.

Where the ASCC Property

`Effective. Period` definition is:

A specific **`period`** of time such as the length of time between two known date/time points, from a start date onwards, or up to an end date that constitutes an **`effective period`**.

6.8.5.3 Association Core Component Business Terms

An ASCC may have several business terms. ASCC business terms are synonym terms under which the ASCC is commonly known and used in business.

[C83] – Each ASCC shall have zero or more business terms.

6.8.6 Association Core Component Localized Information

The ASCC localized information class contains the relevant information necessary to associate native language expressions of ASCC attributes to the ASCC.

[C84] An ASCC shall have zero or more localized information classes.

[C85] Each occurrence of an ASCC localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (Optional):** The official name of the ASCC in a language other than English.
- **Definition (mandatory):** The semantic meaning of the ASCC in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ASCC is commonly known and used in a business expression in that language.

ASCC localized information DENs should follow, as much as possible, all ASCC DEN rules.

[C86] Each ASCC localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[C87] Each ASCC localized information definition shall adhere to all ASCC definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[C88] Each ASCC localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[C89] Each ASCC localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

6.9 Association Core Component Properties

An ASCC property consists of a property term plus the object class term of the associated ACC.

[Definition] – Association Core Component Property (ASCC Property)

An association core component property is a core component property for which the permissible values are expressed as a complex structure, represented by an aggregate core component.

[C90] An ASCC property shall be defined for each ASCC.

ASCC properties are reusable across object classes.

[Example] – Reuse of ASCC Properties in Multiple Object Classes

For the ASCC property of **`Effective. Period – Contract. Effective. Period`** and **`Price. Effective. Period`** – may both exist.

6.9.1 Association Core Component Property – Property Term

Each ASCC property contains a property term. The property term of an ASCC property is a semantically meaningful name for the characteristic that represents the nature of the association to the associated ACC.

[C91] Each ASCC property shall have a property term.

[C92] The property term of an ASCC property may consist of more than one word.

[C93] A multi-worded property term of an ASCC property shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[Example] – Single versus Multiple Word Property Terms

`Trade Line Item. Additional Information. Note` is not the same as **`Trade Line Item. Additional. Note`**

`Trade Line Item. Additional Information. Note` is not the same as **`Trade Line Item. Information. Note`**

`Trade Line Item. Additional Information. Note` is not the same as **`Trade Line Item. Information Additional. Note`**

6.9.2 Association Core Component Property Identifiers

Every ASCC property is a registry class. In order to ensure uniqueness, every ASCC property will have assigned a:

- **Unique Identifier:** The identifier that references an ASCC property in a unique and unambiguous way.
- **Version Identifier:** An indication of the evolution over time of an ASCC property.

[C94] Each ASCC property shall have a unique identifier within the library of which it is a part.

[C95] Each version of an ASCC property shall have a unique version identifier within the library of which it is a part.

6.9.3 Association Core Component Property Common Information

[C96] Each ASCC property shall have a common information class.

[C97] The ASCC property common information class shall conform to all CC common information rules.

[C98] The ASCC property common information class shall consist of:

- **DEN (mandatory):** The official name of the ASCC property.
- **Definition (mandatory):** The semantic meaning of the ASCC property.
- **Business Term (optional, repetitive)**: A synonym term under which the ASCC property is commonly known and used in business.

[Example] – ASCC Property Common Information

DEN – **`Effective. Period`**

Definition – A specific period of time such as the length of time between two known date/time points, from a start date onwards, or up to an end date that constitutes an effective period.

Business Term – **`Effective Duration`**, **`In Force Period`**.

6.9.3.1 Association Core Component Property Dictionary Entry Names

[C99] Each ASCC property DEN shall conform to all CC DEN rules.

[C100] The DEN of an ASCC property shall consist of a property term that represents the nature of the association to the associated ACC, followed by a dot, a space character, and the object class term of the associated ACC.

[Example] – DEN for ASCC properties

`Performance. Metrics; Effective. Period; Effective. Contract`

6.9.3.2 Association Core Component Property Definitions

[C101] Each ASCC property definition shall conform to all CC definition rules.

[C102] The definition of an ASCC property shall include the object class term of the associated ACC and the property term that expresses the nature of the association.

[Example] – ASCC Property Definition

ASCC property: **`Performance. Metrics`**

Performance metrics are a system of quantitative parameters for performance assessment purposes.

6.9.3.3 Association Core Component Property Business Terms

An ASCC property may have several business terms. ASCC property business terms are synonym terms under which the ASCC property is commonly known and used in business.

[C103] Each ASCC property shall have zero or more business terms.

6.9.4 Association Core Component Property Localized Information

The ASCC property localized information class contains the relevant information necessary to associate native language expressions of ASCC property attributes to the ASCC property.

[C104] An ASCC property shall have zero or more localized information classes.

[C105] Each occurrence of an ASCC property localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the ASCC property in a language other than English
- **Definition (mandatory):** The semantic meaning of the ASCC property in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ASCC property is commonly known and used in a business expression in that language.

ASCC property localized information DENs should follow, as much as possible, all ASCC property DEN rules.

[C106] Each ASCC property localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[C107] Each ASCC property localized information definition shall adhere to all ASCC property definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[C108] Each ASCC property localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[C109] Each ASCC property localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

6.10 Basic Core Components

BCCs represent properties of an ACC. The BCC consists of a BCC property and the object class of the ACC to which it belongs.

[Definition] – Basic Core Component (BCC)

A basic core component is a core component which constitutes a singular business characteristic of a specific aggregate core component. It has a unique business semantic definition. A basic core component represents a basic core component property and is therefore of a core data type which defines its value domain. Basic core components function as properties of aggregate core components.

6.10.1 Basic Core Component Usage Rules

A BCC may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BCC. The BCC usage rules represent the specific application of a BCC as an ACC property. BCC usage rules can be either unstructured – expressed as free form text, or structured –expressed in a formal language.

[C110] A BCC shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[C111] BCC usage rules shall not replicate ACC, ASCC, or CDT usage rules.

[C112] A BCC usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BCC usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[C113] An unstructured BCC usage rule constraint shall have a free form text expression that fully details the usage rule.

A BCC structured constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[C114] A structured BCC usage rule shall have a formal constraint expressed in a formal constraint language.

BCC usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[C115] Every BCC usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BCC usage rules will also have a condition type that identifies when the formal constraint should be applied.

[C116] Every BCC usage rule shall have a condition type.

[C117] Every BCC usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**..

6.10.1.1 Basic Core Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given BCC from all other usage rules for a library, a BCC usage rule may also have an identification metadata class that provides additional information.

[C118] A BCC usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates it from all other named usage rules for the BCC.

[C119] A BCC usage rule shall have zero or one names that is unique within the group of usage rules of the ACC to which it belongs.

The BCC usage rule identification metadata class may contain several business terms. BCC usage rule business terms are synonym terms under which the BCC usage rule is commonly known and used in business.

[C120] Each BCC usage rule shall have zero or more business terms.

6.10.1.2 Basic Core Component Usage Rule Localized Metadata

BCC usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[C121] A VCC usage rule shall have zero or more localized metadata classes.

[C122] Each occurrence of a BCC usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[C123] BCC usage rule localized metadata shall be in the language identified by the language and locale code.

6.10.2 Basic Core Component Cardinality

Each BCC, in its role as an ACC property, will have its cardinality explicitly expressed.

[C124] Each BCC shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[C125] BCC cardinality values shall be non-negative integers of zero or greater, or – only in the case of maximum cardinality – the token **unbounded** if no limit applies.

6.10.3 Basic Core Component Sequencing Key

Business requirements may exist for BCCs to occur in a specific order within an ACC. Software and storage applications may have unique sequencing algorithms that change the normatively defined order of the BCC within an ACC. To ensure the desired order is preserved, each BCC within an ACC will be assigned a unique sequencing key.

[C126] Each BCC shall be assigned a unique sequencing key within the ACC of which it is a part.

Note – Sequencing Key Structure

There are no specific rules for the structure of the sequencing keys. Implementers are free to choose any structure providing it guarantees uniqueness within the ACC to which it belongs and the structuring scheme is readily available for anyone accessing or using the ACC.

6.10.4 Basic Core Component Common Information

In its role as an ACC property, each BCC has a common information class.

[C127] Each BCC shall have a common information class.

[C128] The BCC common information class shall conform to all component common information rules.

[C129] The BCC common information class shall consist of:

- **DEN (mandatory):** The official name of the BCC.
- **Definition (mandatory):** The semantic meaning of the BCC.
- **Business Term (optional, repetitive)**: A synonym term under which the BCC is commonly known and used in business.

[Example] – Common Information

DEN – **`Period. Start. Date Time`**

Definition – The date, time, date time or other date time value for the start of this period of time.

Business Term – **`Duration Start`**

6.10.4.1 Basic Core Component Dictionary Entry Names

[C130] Each BCC DEN shall conform to all CC DEN rules.

[C131] The DEN of a BCC shall consist of the following parts in the order specified:

- the object class term of the ACC owning the corresponding BCC, followed by a dot and space character.
- the DEN of the included BCC property.

Example – BCCs

`Period. Start. Date Time`; **`Contract. Price. Amount`**

6.10.4.2 Basic Core Component Definitions

[C132] Each BCC definition shall conform to all CC definition rules.

[C133] The definition of a BCC shall include the object class term of the ACC to which it belongs, and the definition of the included BCC property.

[Example] – BCC Definition

`Period. Start. Date Time`

The **`date`**, **`time`**, **`date time`** or other date time value for the **start** of this **period** of time.

6.10.4.3 Basic Core Component Business Terms

A BCC may have several business terms. BCC business terms are synonym terms under which the BCC is commonly known and used in business.

[C134] Each BCC shall have zero or more business terms.

6.10.5 Basic Core Component Localized Information

The BCC localized information class contains the relevant information necessary to associate native language expressions of BCC attributes to the BCC.

[C135] A BCC shall have zero or more localized information classes.

[C136] Each occurrence of a BCC localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BCC in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BCC in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BCC is commonly known and used in a business expression in that language.

BCC localized information DENs should follow, as much as possible, all BCC DEN rules.

[C137] Each BCC localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[C138] Each BCC localized information definition shall adhere to all BCC definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[C139] Each BCC localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[C140] Each BCC localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

6.11 Basic Core Component Properties

The BCC property represents a generic reusable data element independent of an object class. A BCC property consists of a property term plus a representation term.

[Definition] – Basic Core Component Property (BCC Property)

A basic core component property is a core component property for which the permissible values are expressed by simple values, represented by a data type.

[C141] A BCC property shall be defined for each BCC.

BCC properties are reusable across ACCs.

[Example] – Reuse of BCC Properties in Multiple BCCs

`Contact. Type. Code` and **`Event. Type. Code`** may both exist.

To ensure consistency in use, BCC properties are always based on an approved CDT in the *UN/CEFACT Data Type Catalogue*.

[C142] A BCC Property shall only use an approved CDT in the *UN/CEFACT Data Type Catalogue*.

6.11.1 Basic Core Component Property – Property Term

Each BCC property contains a property term. The property term of a BCC property is a semantically meaningful name for a unique characteristic that can be used in an ACC object class.

[C143] Each BCC property shall have a property term.

[C144] The property term of a BCC property may consist of more than one word.

[C145] A multi-worded property term of a BCC property shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[Example] – Single versus Multiple Word Property Terms

Longitude Direction. Indicator is not the same as **Longitude. Indicator**

Longitude Direction. Indicator is not the same as **Direction. Indicator**

Longitude Direction. Indicator is not the same as **Direction Longitude. Indicator**

6.11.2 Basic Core Component Property Representation Term

Each BCC property contains a representation term. The representation term is a semantically meaningful name that represents the value domain of the BCC property as defined by a CDT. UN/CEFACT defines the approved representation terms as part of the *UN/CEFACT Data Type Catalogue*.

[C146] A representation term shall be defined for each BCC property.

[C147] The name of the BCC property representation term may consist of more than one word.

[C148] A multi-worded BCC property representation term shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[C149] The name of the BCC property representation term shall be one of the approved representation terms in the *UN/CEFACT Data Type Catalogue*.

6.11.3 Basic Core Component Property Identifiers

Every BCC property is a registry class. In order to ensure uniqueness, every BCC property will have assigned a:

- **Unique Identifier (mandatory):** The identifier that references the BCC property in a unique and unambiguous way.
- **Version Identifier (mandatory):** An indication of the evolution over time of the BCC property.

[C150] Each BCC property shall have a unique identifier within the library of which it is a part.

[C151] Each version of a BCC property shall have a unique version identifier within the library of which it is a part.

6.11.4 Basic Core Component Property Common Information

[C152] Each BCC property shall have a common information class.

[C153] The BCC property common information class shall conform to all CC common information rules.

[C154] The BCC property common information class shall consist of:

- **DEN (mandatory):** The official name of the BCC property.
- **Definition (mandatory):** The semantic meaning of the BCC property.
- **Business Term (optional, repetitive)**: A synonym term under which the BCC property is commonly known and used in business.

[Example] – BCC Property Common Information

DEN – **`Start. Date Time`**

Definition – A date, time, **`date time`** or other date time value that marks the **`start`** or initiation of an event.

Business Term – Beginning Date Time

6.11.4.1 Basic Core Component Property Dictionary Entry Names

[C155] Each BCC property shall have a DEN.

[C156] Each BCC property DEN shall conform to all CC DEN rules.

[C157] The DEN of a BCC property shall consist of a property term, followed by a dot, a space character, and a representation term.

[C158] The DEN of a BCC property shall be unique within the context of an object class but may be reused across different object classes.

[Example] – Reuse of CC Properties in Multiple Object Classes

`Contract. Type. Code` and **`Metrics. Type. Code`** may both exist.

[Example] – DEN for BCC Property

`Start. Date Time`; **`Start. Measure Type. Code`**; **`Type. Text`**

6.11.4.2 Basic Core Component Property Definitions

[C159] BCC property definitions shall conform to all CC definition rules.

[C160] The definition of a BCC property shall include the property and representation term of the BCC property.

[Example] – BCC Property Definition

`Start. Date Time`

A date, time, **date time** or other date time value that marks the **start** or initiation of an event.

6.11.4.3 Basic Core Component Property Business Terms

A BCC property may have several business terms. BCC property business terms are synonym terms under which the BCC property is commonly known and used in business.

[C161] Each BCC property shall have zero or more business terms.

6.11.5 Basic Core Component Property Localized Information

The BCC property localized information class contains the relevant information necessary to associate native language expressions of BCC property attributes to the BCC property.

[C162] A BCC property shall have zero or more localized information classes.

[C163] Each occurrence of a BCC property localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BCC property in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BCC property in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BCC property is commonly known and used in a business expression in that language.

BCC property localized information DENs should follow, as much as possible, all BCC property DEN rules.

[C164] Each BCC property localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[C165] Each BCC property localized information definition shall adhere to all BCC definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[C166] Each BCC property localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[C167] Each BCC property localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

7 Business Information Entity Model

This section provides a detailed technical explanation of the Business Information Entity (BIE) metamodel as seen in the UML diagram figure 7-1.

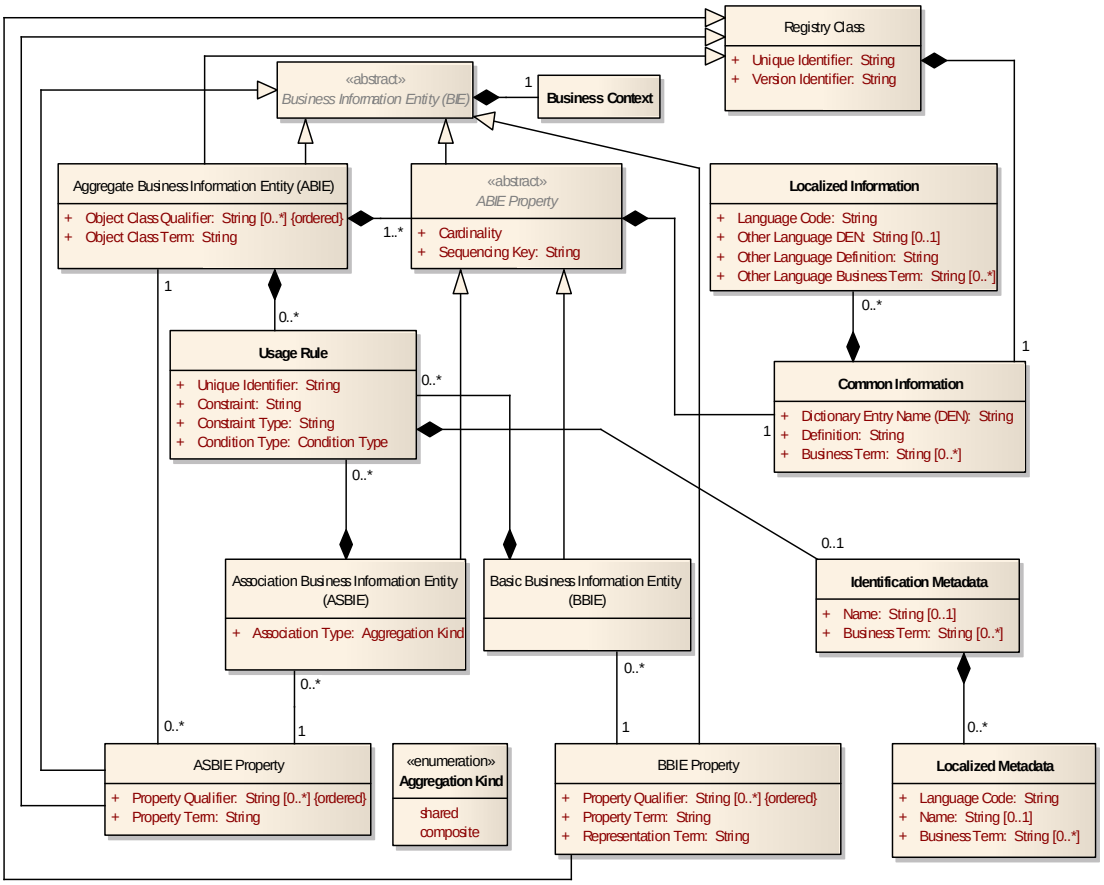


Figure 7-1. UML Diagram of Business Information Entity Basic Definition Model

7.1 Overview

Business information entities represent the real world application of core components for a specific context of use. BIEs are used to create logical models that may be implemented in a specific syntax as document models for business information exchanges. A BIE is a context specific instantiation of a conceptual core component. A BIE will be part of a package within a library. The package represents a set of BIEs being used in a specific context and tailored to meet the unique requirements for the package. BIEs are semantically unique within a package, but may be semantically similar in name and definition to, albeit with a different content model than, BIEs in other packages.

[Definition] – Business Information Entity (BIE)

A business information entity is a context specific instantiation of a core component that constitutes a piece of business data or a group of pieces of business data with a unique business semantic definition in a specific business context.

Figure 7-2 shows the relationships between BIEs and their CC counterparts.

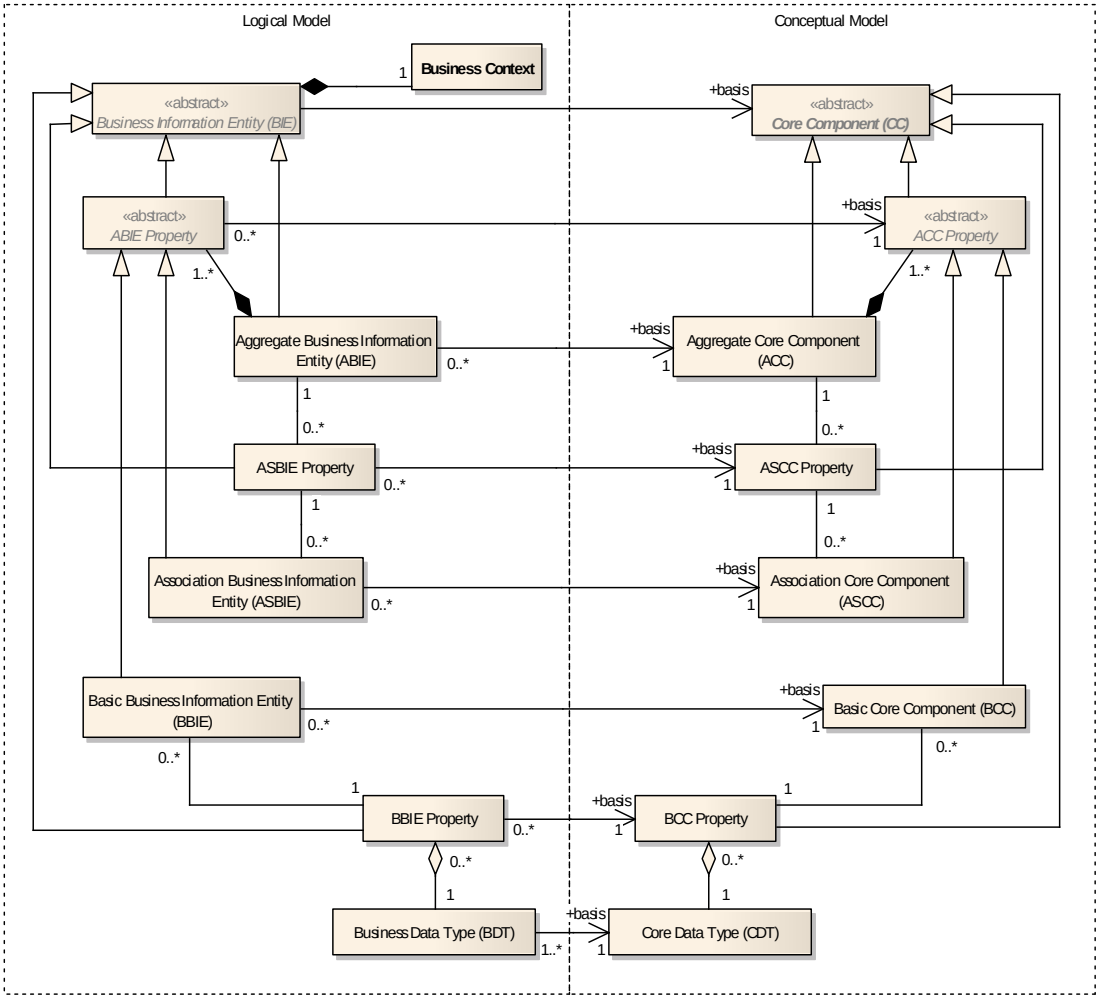


Figure 7-2. UML Diagram of Relationship Between Business Information Entities And Core Components

[Note] – Figure 7-2

For completeness, figure 7-2 includes CDTs and BDTs (See Section 8).

Just as with ACCs, there are five categories of BIEs:

- Aggregate Business Information Entity (ABIE). An ABIE is based on an (has one and only one source) ACC.
- Association Business Information Entity (ASBIE). An ASBIE is based on an (has one and only one source) ASCC.

- Basic Business Information Entity (BBIE). A BBIE is based on a (has one and only one source) BCC.
- Association Business Information Entity Property. An ASBIE property is based on an (has one and only one source) ASCC property.
- Basic Business Information Entity Property. A BBIE property is based on a (has only one source) BCC property.

[B1] A BIE shall be an ABIE, ASBIE, BBIE, ASBIE property or a BBIE property.

[B2] A BIE shall be defined by one or more individual business context category values that together constitute a unique business context.

[Definition] – Business Context

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

ABIEs, ASBIEs, BBIEs, ASBIE properties, and BBIE properties are collectively called BIEs and are typically stored in a registry, database, or other mechanism to maximize reuse.

7.2 Business Information Entity Naming and Definition Conventions

BIE naming and definition conventions are based on CC naming and definition conventions to ensure consistency in the naming and defining of BIEs with their source CCs. The BIE naming and definition conventions are derived from the guidelines and principles described in *ISO 11179 Part 4 – Definitions and ISO 11179 Part 5 – Naming and Identification Principles*.

The official language for UN/CEFACT BIEs is English. All official dictionary entries will be in English. BIE discovery work may very well occur in other languages; however official submissions for inclusion in the UN/CEFACT library must be in English. In order to ensure absolute clarity and understanding of the names and definitions it is essential to use words from the *Oxford English Dictionary*.

As with CCs, a controlled vocabulary will be developed to identify the definition to be used for any words that are potentially ambiguous.

[Note] – UN/CEFACT Controlled Vocabulary

Implementers are encouraged to use the UN/CEFACT controlled vocabulary as the authoritative source for BIE terms.

7.3 Business Information Entity Registry Class

A BIE registry class represents a cohesive set of information associated with a single BIE. Each ABIE, ASBIE property, and BBIE property are registry classes.

[B3] A registry class shall be created for each ABIE, ASBIE property, and BBIE property.

Each BIE registry class contains the following information:

- Unique Identifier

- Unique Version Identifier

[Note] – BIE Identifier Structure

There are no specific rules for the structure of the BIE identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the library to which it belongs.

7.4 Business Information Entity Common Information

The BIE common information class provides necessary component information that is applicable to business components either directly or through inheritance. The BIE common information class contains the following information:

- **DEN**. This is the unique official name of the BIE in the dictionary.
- **Definition**. This is the unique business semantic meaning of the BIE.
- **Business Term**. This is a synonym term under which the BIE is commonly known and used in business for a specific context. A BIE may have several business terms.

[Example] – BIE Common Information

DEN – **`Trade_ Contract. Business_ Type. Code`**

Definition – A **`code`** specifying a **`business type`** of the **`trade contract`**, such as a fixed price contract, or a time and materials based contract.

Business Term – **`Service Agreement Type`**

7.4.1 Business Information Entity Dictionary Entry Name

BIE naming rules are based on the following concepts as defined in ISO 11179:

- **Object Class** – represents the logical data grouping or aggregation (in a logical data model) to which a property belongs. The object class is represented by an object class term. Thus, the object class is the part of a BIE's DEN that represents an activity or object in a specific context. Object classes have explicit boundaries and meaning and their properties and behaviour follow the same rules.
- **Property Term** – represents a distinguishing characteristic of the object class and shall occur naturally in the definition.
- **Representation Term** – an element of the BIE name which describes the form in which the BIE is represented.
- **Qualifier Term** – a word or words which help define and differentiate a BIE from its associated CC and other BIEs. Qualifier terms are used to refine the semantic meaning of the DEN to reflect restriction to the BIE object class term and/or property terms as necessary to distinguish one BIE concept, conceptual domain, content model or data value domain from another.

[B4] BIE DENs shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[Note] – *Oxford English Dictionary*

Users may choose to utilize any version of the *Oxford English Dictionary* to create the spelling and definitions of BIEs; however the complete *Oxford English Dictionary* will be the authoritative source for conflict resolution between competing spellings of component names or definitions.

[B5] A BIE DEN shall be unique amongst all BIE DENs within the package of which it is a part.

[B6] A BIE DEN shall be extracted from the BIE definition.

[B7] A BIE DEN shall not include consecutive identical words.

[B8] A BIE DEN and all its components shall be in singular form unless the concept itself is plural.

[B9] A BIE DEN shall only use alphabetic characters plus the dot, the underscore and space characters.

[B10] A BIE DEN shall only contain verbs, nouns, adverbs and adjectives unless a different part of speech is part of an official title, part of a term listed in the *Oxford English Dictionary*, or part of a controlled vocabulary.

[Note] – Parts of Speech

Articles, prepositions and related parts of speech that are not verbs, nouns, adverbs and adjectives normally add no semantic clarity and should never be used unless as part of an official title or in a controlled vocabulary as part of a common business term that can not otherwise be expressed.

[Example] – Exception Use of Parts of Speech

`Office Of Surface Mining_ Goods Item. Free On Board Value. Amount`

Where the **`Office Of Surface Mining`** is a formal title that contains the preposition **`Of`**, and removal of the preposition would identify a different organization; and **`Free On Board Value`** where **`Free On Board`** is a recognized expression and removal of the preposition **`On`** would change the semantic meaning of the property term.

[B11] Abbreviations and acronyms that are part of the BIE DEN shall be expanded or explained in the definition.

[B12] BIE DEN object class terms, property terms, and representation terms shall be separated by dots.

[B13] The space character shall separate words in multi-worded BIE object class, property, and representation terms.

[B14] Each word in a BIE DEN shall start with a capital letter.

[B15] The dots after BIE object class and property terms shall be followed by a space character.

[B16] Each BIE DEN shall remain unique when its separators are removed.

[B17] Multi-worded object classes and property terms shall be used in lieu of qualifier terms when the concept the multi worded object class or property term represents exists in three or more dissimilar business domains.

[B18] The order of qualifier terms shall have semantic meaning.

[Example] – Qualifier Order

The BBIE **`Electronic_ Trade_ Contract. Issue. Date Time`** has a different semantic meaning than **`Trade_ Electronic_ Contract. Issue. Date Time`**.

[B19] Qualifier terms shall reflect the semantic restriction of the object class or property term that they are used with.

[Example] – Semantic Restrictions

`Trade_ Contract. Details` semantically restricts **`Contract. Details.`** The qualifier term of **`Trade`** is allowed even though it also may exist as a separate object class, property term, or representation term.

7.4.2 Business Information Entity Definitions

BIE definitions are based on the requirements for data element definitions defined in ISO 11179-4.

[B20] Each BIE shall have its own unique semantic definition within the library of which it is a part.

[Note] – Order of Development of Definition and DEN

In the interest of quality, it is recommended that the BIE definition be developed first and the DEN extracted from it.

[B21] The BIE definition shall be derived from the source CC definition.

[B22] The BIE definition shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[B23] The BIE definition shall be consistent with the requirements of *ISO 11179-4 Section 4* and shall provide an understandable meaning, which should also be translatable to other languages.

[B24] The BIE definition shall take into account the fact that the users of the BIE library are not necessarily native English speakers. It shall therefore contain short sentences, using normal words. Wherever synonym terms are possible, the definition shall use the preferred term as identified in the controlled vocabulary.

[B25] Whenever both the definite (i.e. **`the`**) and indefinite article (i.e. **a**) are possible in a BIE definition, preference shall be given to an indefinite article (i.e. **a**).

[Note] – Definition Quality

To verify the quality of the definition, place the DEN followed by the word **`is`** before the definition to ensure that it is not simply a repetition of the DEN.

7.4.3 Business Information Entity Business Terms

BIE business terms are those terms commonly used for day-to-day information exchanges within a given domain. As such, no specific naming rules apply to

business term structures. Interoperability of business terms will be given by linking them within the BIE common information class.

7.5 Business Information Entity Localized Information Class

As with CCs, the normative expressions of BIEs are in the English language. While the normative expressions of BIEs are in the English language, implementers may choose to create alternative language expressions of definitions and business terms. The BIE localized information class contains the relevant information necessary to associate the native language expressions to their normative English language counterparts. Other language BIE DENs will only consist of alphabetic, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules. In addition to other language DEN, definition, and business term(s), a mandatory language code identifies the language in which the components are being expressed for storage in the registry. The localized information class contains:

- **Language Code –** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* will be used as the authoritative source for code values.
- **DEN –** The official name of the component in a language other than English.
- **Definition –** the semantic meaning of the component in a language other than English.
- **Business Term –** A synonym term in another language under which the component is commonly known and used in a business expression in that language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language and locale code. The business terms must only be expressed in the language identified by the language and locale code, or a recognized dialect of the language.

7.6 Aggregate Business Information Entities

Each ABIE represents the logical data grouping or aggregation (in a logical data model) of the concept of the ABIE.

[Definition] – Aggregate Business Information Entity (ABIE)

An aggregate business information entity is a collection of related pieces of business information that together convey a distinct business meaning in a specific business context. Expressed in modelling terms, it is the representation of an object class, in a specific business context.

An ABIE represents an ACC with business context applied.

[B26] An ABIE shall be based on an ACC.

7.6.1 Aggregate Business Information Entity Object Class Term

The ABIE object class is expressed by an object class term. The ABIE object class term is the same as the ACC on which it is based. The object class term is a semantically meaningful name for the object class that is represented by the ABIE. It

serves as the basis for the DEN of the ABIE and for the DEN of all BBIEs and ASBIEs that are properties of the ABIE.

[B27] An ABIE object class term shall be identical to its basis ACC object class term.

7.6.2 Aggregate Business Information Entity Object Class Term Qualifier

The ABIE object class term qualifier is a word or words which help define and differentiate an ABIE from its associated CC and other BIEs. The ABIE object class term qualifier enhances the semantic meaning of the ABIE DEN to reflect a restriction to the BIE concept, conceptual domain, content model or data value. ABIE object class terms can have one or more qualifier terms.

[B28] A qualified ABIE shall be a restriction of its source ACC or its higher level ABIE in an ABIE hierarchy.

[Example] – Multi-qualified ABIEs

The Multi-qualified ABIE
Electronic_ Trade_ Contract. Details

qualifies the qualified ABIE
Trade_ Contract. Details

which qualifies the ACC
Contract. Details

Whereas the multi-word qualified
Electronic Trade_ Contract. Details

Qualifies the ACC
Contract. Details

and not the qualified ABIE
Trade_ Contract. Details.

[B29] ABIE object class qualifier terms shall precede the object class term.

[B30] Each ABIE object class qualifier term shall be followed by an underscore and a space character (_).

[B31] A multi-worded object class qualifier term shall have a unique semantic meaning compared to the words separately.

[B32] A qualifying ABIE hierarchy shall be established when multiple qualifiers are used.

Note – BIE Hierarchy

A BIE hierarchy is a tree like structure that reflects the order of the qualifiers for a set of qualified BIEs derived from the same unqualified BIE in a graph like form. The first level in a BIE hierarchy is the unqualified BIE construct, and each succeeding lower level is a more qualified BIE construct than its preceding BIE construct.

[Example] – BIE Hierarchy

Trade_ Contract. Details and **Electronic_ Trade_ Contract. Details** where **Electronic_ Trade_ Contract. Details** is a reuse and restriction of **Trade_ Contract. Details** and the hierarchy of **Trade_ Contract** is preserved in the reuse.

[B33] A qualified object class name shall be unique amongst the set of qualified object class names in the library of which it is a part.

[B34] A qualified object class name may be applied in its entirety as a qualifier for another object class to convey a semantic relationship between the objects providing the qualifier hierarchy is preserved.

7.6.3 Aggregate Business Information Entity Usage Rule

ABIEs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the ABIE. ABIE usage rules represent the specific application of an ABIE in its role as an object class. ABIE usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[B35] An ABIE shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[B36] ABIE usage rules shall not replicate BBIE, ASBIE, or BDT usage rules.

[B37] An ABIE usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The ABIE usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[B38] An unstructured ABIE usage rule constraint shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[B39] A structured ABIE usage rule shall have a formal constraint language expression.

ABIE usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[B40] Every ABIE usage rule shall have a formal constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

ABIE usage rules will also have a condition type that identifies when the constraint should be applied.

[B41] Every ABIE usage rule shall have a condition type.

[B42] Every ABIE usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

7.6.3.1 Aggregate Business Information Entity Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given ABIE from all other usage rules for a library, an ABIE usage rule may also have an identification metadata class that provides additional information.

[B43] An ABIE usage rule shall have zero or one identification metadata classes.

The ABIE usage rule identification metadata class contains a unique name that semantically differentiates the usage rule from all other named usage rules for the ABIE.

[B44] An ABIE usage rule shall have zero or one names that is unique within the group of usage rules of an ABIE.

The ABIE usage rule identification metadata class may contain business terms. ABIE usage rule business terms are synonym terms under which the ABIE usage rule is commonly known and used in business.

[B45] Each ABIE usage rule shall have zero or more business terms.

7.6.3.2 Aggregate Business Information Entity Usage Rule Localized Metadata

ABIE usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[B46] An ABIE usage rule shall have zero or more localized metadata classes.

[B47] Each occurrence of an ABIE usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[B48] ABIE usage rule localized metadata shall be in the language identified by the language and locale code.

7.6.4 Aggregate Business Information Entity Identifiers

Every ABIE is a registry class. In order to ensure uniqueness, every ABIE will have assigned a:

- **Unique Identifier:** The identifier that references an ABIE in a unique and unambiguous way.
- **Version Identifier:** An indication of the evolution over time of an ABIE.

[B49] Each ABIE shall have a unique identifier within the library of which it is a part.

[B50] Each version of an ABIE shall have a unique version identifier within the library of which it is a part.

7.6.5 Aggregate Business Information Entity Common Information

[B51] Each ABIE shall have a common information class.

[B52] The ABIE common information class shall conform to all BIE common information rules.

[B53] The ABIE common information class shall consist of:

- **DEN (mandatory):** The official name of the ABIE.
- **Definition (mandatory):** The semantic meaning of the ABIE.
- **Business Term (optional, repetitive)**: A synonym term under which the ABIE is commonly known and used in business.

[Example] – ABIE Common Information

DEN – **`Trade_ Contract. Details`**

Definition – A **`trade contract`** is a **`contract`**ual agreement between two or more parties for **`trade`** purposes.

Business Term – **`Service Agreement`**

7.6.5.1 Aggregate Business Information Entity Dictionary Entry Names

[B54] Each ABIE DEN shall conform to all BIE DEN rules.

[B55] The DEN of an ABIE shall consist of the object class term of the ACC it is based on, and possibly additional qualifier term(s) to represent its specific business context, followed by a dot, a space character, and the term **`Details`**.

[Example] – DEN for ABIEs

`Trade_ Contract. Details`; **`Currency Exchange. Details`**

7.6.5.2 Aggregate Business Information Entity Definitions

[B56] Each ABIE definition shall conform to all BIE definition rules.

[B57] The definition of an ABIE shall include the object class term and any qualifier terms.

[Example] – ABIE Definition

Trade_ Contract. Details

A **trade contract** is a **contract**ual agreement between two or more parties for **trade** purposes.

[B58] An ABIE with an unqualified object class shall have the same definition as the ACC the ABIE is based on.

[B59] An ABIE with a qualified object class term shall have a definition that semantically restricts the definition of the less qualified ABIE or ACC that the ABIE is based on.

7.6.5.3 Aggregate Business Information Entity Business Terms

An ABIE may have several business terms. ABIE business terms are synonym terms under which the ABIE is commonly known and used in business.

[B60] Each ABIE shall have zero or more business terms.

7.6.6 Aggregate Business Information Entity Localized Information

The ABIE localized information class contains the relevant information necessary to associate native language expressions of ABIE attributes to the ABIE.

[B61] Each ABIE shall have zero or more localized information classes.

[B62] Each occurrence of an ABIE localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the ABIE in a language other than English.
- **Definition (mandatory):** The semantic meaning of the ABIE in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ABIE is commonly known and used in a business expression in that language.

ABIE localized information DENs should follow, as much as possible, all ABIE DEN rules.

[B63] Each ABIE localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[B64] Each ABIE localized information definition shall adhere to all ABIE and definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[B65] Each ABIE localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[B66] Each ABIE localized information business term shall be in the language identified by the language and code.

7.7 Aggregate Business Information Entity Properties

An ABIE consists of ABIE properties. The ABIE property is a generalization of either an ASBIE or a BBIE. Every ABIE contains at least one ABIE property.

[Definition] – Aggregate Business Information Entity Property

An aggregate business information entity property is a business information entity property for which the permissible values are expressed as a complex structure, represented by an aggregate business information entity.

[B67] An ABIE shall contain at least one ABIE property.

[B68] An ABIE property shall be based on a CC property of the corresponding ACC.

[B69] An ABIE property shall either be a BBIE or an ASBIE.

Because an ABIE is an independent class, it is important that all listed properties are in fact conceptually related to the concept of the BIE, and not just added for convenience.

[B70] Within an ABIE, all embedded BBIEs and ASBIEs shall be related to the concept of the aggregate.

ABIE properties must be unique within the ABIE.

[B71] An ASBIE and a BBIE DEN shall never be identical when used in an ABIE.

An ABIE property that is an ASBIE must be devoid of mandatory circular references.

[B72] An ABIE shall never contain – directly or at any nested level – a mandatory ASBIE whose associated ABIE is the same as the top level ABIE.

[Note] – Recursion

The objective of the above rule is to avoid endless loops in the content model of an ABIE. The rule allows an ABIE to contain an ASBIE property that references itself. The fact that the ASBIE property is not mandatory makes it possible to stop the loop after a finite number of iterations.

7.8 Association Business Information Entities

An ASBIE is an ASCC with context. ASBIEs associate two ABIEs, where the associated ABIE is the property of the associating ABIE. The property term represents the role of the associated ABIE in the association. ASBIEs have a defined minimum and maximum occurrence. The associated ABIE in an ASBIE may be either a UML association **`AggregationKind=shared`** or **`AggregationKind=composite`** when used in an information model.

[Definition] – Association Business Information Entity (ASBIE)

An association business information entity is a business information entity which defines the role of a specific aggregate business information entity (known as the associated ABIE) associated to another aggregate business information entity (known as the associating ABIE). An association business information entity functions as an ABIE property of the associating ABIE.

[B73] An ASBIE shall be based on an ASCC.

7.8.1 Association Business Information Entity Association Type

ASBIEs represent an association between the associating (parent) ABIE and the associated (child) ABIE. The nature of the association of all ASBIEs may be either a UML association **`AggregationKind=shared`** or **`AggregationKind=composite`**. An association type indicator is required to reflect this association as a mechanism for transformation between alternative syntax storage expressions and UML representation.

[B74] An ASBIE shall have an UML association **`AggregationKind =shared`** or **`AggregationKind=composite`**.

7.8.2 Association Business Information Entity Usage Rule

ASBIEs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the ASBIE. ASBIE usage rules clarify (or constrain) the use of an ASBIE as an ABIE property. ASBIE usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[B75] An ASBIE shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[B76] ASBIE usage rules shall not replicate ABIE, BBIE, or BDT usage rules.

[B77] An ASBIE usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The ASBIE usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[B78] An ustructured ASBIE usage rule constraint shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[B79] A structured ASBIE usage rule constraint shall have a formal constraint language expression.

ASBIE usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[B80] Every ASBIE usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

ASBIE usage rules will also a have condition type that identifies when the constraint should be applied.

[B81] Every ASBIE usage rule shall have a condition type.

[B82] Every ASBIE usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

7.8.2.1 Association Business Information Entity Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given ASBIE from all other usage rules for a library, an ASBIE usage rule may also have an identification metadata class that provides additional information.

[B83] An ASBIE usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates it from all other named usage rules for the ASBIE.

[B84] An ASBIE usage rule shall have zero or one names that is unique within the group of usage rules of an ASBIE.

The ASBIE usage rule metadata class may contain business terms. ASBIE usage rule business terms are synonym terms under which the ASBIE usage rule is commonly known and used in business.

[B85] Each ASBIE usage rule shall have zero or more business terms.

7.8.2.2 Association Business Information Entity Usage Rule Localized Metadata

ASBIE usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[B86] An ASBIE usage rule shall have zero or more localized metadata classes.

[B87] Each occurrence of an ASBIE usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[B88] ASBIE usage rule localized metadata shall be in the language identified by the language and locale code.

7.8.3 Association Business Information Entity Cardinality

Each ASBIE, in its role as an ABIE property, will have its cardinality explicitly expressed.

[B89] Each ASBIE shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[B90] ASBIE cardinality values shall be non-negative integers of zero or greater, or – only in case of maximum occurrence – the token **unbounded** if no limit applies.

The ASBIE minimum cardinality will never be smaller than the ASCC minimum cardinality and the maximum cardinality will never be larger than the ASCC maximum cardinality.

[B91] ASBIE cardinality values shall never be an extension of its basis ASCC cardinality.

7.8.4 Association Business Information Entity Sequencing Key

Business requirements may exist for ASBIEs to occur in a specific order within an ABIE. Software and storage applications may have unique sequencing algorithms that change the normatively defined order of the ASBIE within an ABIE. To ensure the desired order is preserved, each ASBIE within an ABIE will be assigned a unique sequencing key.

[B92] Each ASBIE shall be assigned a unique sequencing key within the ABIE of which it is a part.

[Note] – Sequencing Key Structure

There are no specific rules for the structure of the sequencing keys. Implementers are free to choose any structure providing it guarantees uniqueness within the ABIE to which it belongs and the structuring scheme is readily available for anyone accessing or using the ABIE.

Since ASBIEs represent contextualized expressions of their source ASCCs, the sequencing requirements of an ASBIE in an ABIE might be different from the sequencing key of the corresponding ASCC in an ACC.

[B93] An ASBIE sequencing key may be different from its corresponding ASCC sequencing key.

7.8.5 Association Business Information Entity Common Information

In its role as an ABIE property, each ASBIE has a common information class.

[B94] Each ASBIE shall have a common information class.

[B95] The ASBIE common information class shall conform to all BIE common information rules.

[B96] The ASBIE common information class shall consist of:

- **DEN (mandatory):** The official name of the ASBIE.

- **Definition (mandatory):** The semantic meaning of the ASBIE.
- **Business Term (optional, repetitive)**: A synonym term under which the ASBIE is commonly known and used in business.

[Example] – ASBIE Common Information

DEN – **`Trade_ Contract. Effective. Measurement_ Period`**

Definition – A **`period`** within which the **`measurement`** of provisions of this **`trade contract`** are, or will be **`effective`**.

Business Term – **`Service Agreement Duration`**

7.8.5.1 Association Business Information Entity Dictionary Entry Names

[B97] Each ASBIE DEN shall conform to all BIE DEN rules.

[B98] The DEN of an ASBIE shall consist of the following components in the specified order:

- the object class term and qualifiers, if any, of the associating BIE,
- the DEN of the included ASBIE property.

[Example] – Association Business Information Entity DEN

`Trade_ Contract. Effective. Measurement_ Period`

Where the associated ABIE **`Measurement_ Period. Details`** now becomes part of a property in the associating ABIE of **`Trade_ Contract. Details`** and the property term (nature of that association) is **`Effective`**.

7.8.5.2 Association Business Information Entity Definitions

[B99] Each ASBIE definition shall conform to all BIE definition rules.

[B100] The definition of an ASBIE shall include the object class term and object class qualifier terms, if any, of the associating ABIE, and the definition of the ASBIE property the ASBIE includes.

7.8.5.3 Association Business Information Entity Business Terms

An ASBIE may have several business terms. ASBIE business terms are synonym terms under which the ASBIE is commonly known and used in business.

[B101] Each ASBIE shall have zero or more business terms.

7.8.6 Association Business Information Entity Localized Information

The ASBIE localized information class contains the relevant information necessary to associate native language expressions of ASBIE attributes to the ASBIE.

[B102] An ASBIE shall have zero or more localized information classes.

[B103] Each occurrence of an ASBIE localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.

- **DEN (optional):** The official name of the ASBIE in a language other than English.
- **Definition (mandatory):** The semantic meaning of the ASBIE in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ASBIE is commonly known and used in a business expression in that language.

ASBIE localized information DENs should follow, as much as possible, all ASBIE DEN rules.

[B104] Each ASBIE localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[B105] Each ASBIE localized information definition shall adhere to all ASBIE definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[B106] Each localized information ASBIE DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[B107] Each ASBIE localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

7.9 Association Business Information Entity Properties

An ASBIE property consists of a property term and qualifiers if any, plus the object class term and qualifiers if any, of the associated ABIE.

[Definition] – Association Business Information Entity Property

An association business information entity property is a business information entity property for which the permissible values are expressed as a complex structure, represented by an aggregate business information entity.

[B108] An ASBIE property shall be defined for each ASBIE.

ASBIE properties are reusable across object classes and packages.

[Example] – Reuse of ASBIE Properties in Multiple Object Classes

`Trade_ Contract. Effective. Measurement_ Period` and **`Lodging House. Effective. Measurement_ Period`** may both exist.

[B109] An ASBIE property shall be based on an ASCC property.

[B110] The associated ABIE of an ASBIE property shall be based on the associated ACC of the corresponding ASCC property.

7.9.1 Association Business Information Entity Property – Property Term

Each ASBIE property contains a property term. The property term of an ASBIE property is a semantically meaningful name for the characteristic that represents the nature of the association to the associated ABIE.

[B111] Each ASBIE property shall have a property term.

[B112] The property term of an ASBIE property may consist of more than one word.

[B113] A multi-worded property term of an ASBIE property shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[Example] – Single versus Multiple Word Property Terms

For the ASBIE **`Bid Bond_ Guarantee. Credit Charge. Guarantee Creditor_ Organization`**:

`Credit Charge. Guarantee Creditor_ Organization` is different than **`Credit. Guarantee Creditor_ Organization`**

`Credit Charge. Guarantee Creditor_ Organization` is different than **`Charge. Guarantee Creditor_ Organization`**

`Credit Charge. Guarantee Creditor_ Organization` is different than **`Charge Credit. Guarantee Creditor_ Organization`**

7.9.2 Association Business Information Entity Property Qualifier Term

The ASBIE property qualifier term is a word or words which help define and differentiate an ASBIE property from its associated ASCC property and other ASBIE properties. The ASBIE property qualifier enhances the semantic meaning of the ASBIE property DEN to reflect a restriction to the ASBIE property concept, conceptual domain, content model or data value. ASBIE properties can have one or more qualifier terms.

[B114] A qualified ASBIE property shall be a restriction of its source ASCC property or its higher level ASBIE properties in an ASBIE property hierarchy.

[Example] – Multi-qualified ASBIE Properties

The Multi-qualified ASBIE property

Total_ Actual_ Quantity. Work Item_ Dimension

qualifies the qualified ASBIE property

Actual_ Quantity. Work Item_ Dimension

which qualifies the ASCC property

Quantity. Dimension

Whereas the multi-word qualified

Initial Credit_ Charge. Creditor_ Organization

Qualifies the ASCC property

Charge. Organization

and not the qualified ASBIE property

Credit_ Charge. Creditor_ Organization

[B115] ASBIE property qualifier terms shall precede the property term.

[B116] Each ASBIE property qualifier term shall be followed by an underscore and a space character ( ).

[B117] A multi-worded ASBIE property qualifier term shall have a unique semantic meaning compared to the words separately.

[B118] A qualifying ASBIE property hierarchy shall be established when multiple qualifiers are used.

[B119] A qualified property term of an ASBIE property DEN may be applied in its entirety as a qualifier for another property term to convey a semantic relationship between the objects providing the qualifier hierarchy is preserved.

7.9.3 Association Business Information Entity Property Identifiers

In order to ensure uniqueness, every ASBIE property will have assigned a:

- **Unique Identifier:** The identifier that references an ASBIE property in a unique and unambiguous way.
- **Version Identifier:** An indication of the evolution over time of an ASBIE property.

[B120] Each ASBIE property shall have a unique identifier within the library of which it is a part.

[B121] Each version of an ASBIE property shall have a unique version identifier within the library of which it is a part.

7.9.4 Association Business Information Entity Property Common Information

[B122] Each ASBIE property shall have a common information class.

[B123] The ASBIE property common information class shall conform to all BIE common information rules.

[B124] The ASBIE property common information class shall consist of:

- **DEN (mandatory):** The official name of the ASBIE property.
- **Definition (mandatory):** The semantic meaning of the ASBIE property.
- **Business Term (optional, repetitive)**: A synonym term under which the ASBIE property is commonly known and used in business.

[Example] – ASBIE Property Common Information

DEN – **`Effective. Measurement_ Period`**

Definition – A **`period`** within which the **`measurement`** of provisions are, or will be **`effective`**.

Business Term – **`Valid Measurement Period`**

7.9.4.1 Association Business Information Entity Property Dictionary Entry Names

[B125] Each ASBIE property shall have a DEN.

[B126] Each ASBIE property DEN shall conform to all BIE DEN rules.

[B127] The DEN of an ASBIE property shall consist of a property term and property term qualifiers, if any, plus the object class term and qualifiers, if any, of the associated ABIE.

7.9.4.2 Association Business Information Entity Property Definitions

[B128] Each ASBIE property definition shall conform to all BIE definition rules.

[B129] The definition of an ASBIE property shall include the object class term and qualifiers, if any, of the associated ABIE and the property term and qualifiers, if any, that express the nature of the association.

[Example] – ASBIE Property Definition

`Effective. Measurement_ Period`

Definition – A **`period`** within which the **`measurement`** of provisions are, or will be **`effective`**.

Where the associated object class term **`period`**, and its qualifier **`Measurement`**, and property term **`Effective`** are included in the definition.

[B130] An ASBIE property with a qualified property term shall have a definition that semantically restricts the definition of the less qualified ASBIE property or the ASCC property that the ASBIE property is based on.

7.9.4.3 Association Business Information Entity Property Business Terms

An ASBIE property may have several business terms. ASBIE property business terms are synonym terms under which the ASBIE property is commonly known and used in business.

[B131] Each ASBIE property shall have zero or more business terms.

7.9.5 Association Business Information Entity Property Localized Information

The ASBIE property localized information class contains the relevant information necessary to associate native language expressions of ASBIE property attributes to the ASBIE property.

[B132] An ASBIE property shall have zero or more localized information classes.

[B133] Each occurrence of an ASBIE property localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the ASBIE property in a language other than English.
- **Definition (mandatory):** The semantic meaning of the ASBIE property in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the ASBIE property is commonly known and used in a business expression in that language.

ASBIE localized information DENs should follow, as much as possible, all ASBIE property DEN rules.

[B134] Each ASBIE property localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[B135] Each ASBIE localized information property definition shall adhere to all ASBIE property definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[B136] Each ASBIE property localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[B137] Each ASBIE property localized information business term shall be expressed in the language identified by the language and locale code, or a recognized dialect of the language.

7.10 Basic Business Information Entities

BBIEs represent simple ABIE properties. The BBIE consists of a BBIE property and the object class of the ABIE to which it belongs.

[Definition] – Basic Business Information Entity (BBIE)

A basic business information entity is a business information entity that represents a singular business characteristic of a specific aggregate business information entity in a given business context. A basic business information entity is based on a basic core component and has a basic business information entity property that is based on a business data type which defines its value domain.

[B138] A BBIE shall be based on a BCC.

7.10.1 Basic Business Information Entity Usage Rules

BBIEs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BBIE. The BBIE usage rules represent the specific application of a BBIE as an ABIE property. BBIE usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[B139] A BBIE shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[B140] BBIE usage rules shall not replicate ABIE, ASBIE, or BDT usage rules.

[B141] A BBIE usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BBIE usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[B142] An unstructured BBIE usage rule constraint shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal constraint language such as the UML OCL or OMG SBVR.

[B143] A structured BBIE usage rule constraint shall have a formal constraint language expression.

BBIE usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[B144] Every BBIE usage rule shall have a constraint type taken from a constraint type code list.

[Note] – Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BBIE usage rules will also have a condition type that identifies when the formal constraint should be applied.

[B145] Every BBIE usage rule shall have a condition type.

[B146] Every BBIE usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

7.10.1.1 Basic Business Information Entity Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given BBIE from all other usage rules for a library, a BBIE usage rule may also have an identification metadata class that provides additional information.

[B147] A BBIE usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates it from all other named usage rules for the BBIE.

[B148] A BBIE usage rule shall have zero or one names that is unique within the group of usage rules of an ABIE.

The BBIE usage rule identification metadata may contain several business terms. BBIE usage rule business terms are synonym terms under which the BBIE usage rule is commonly known and used in business.

[B149] Each BBIE usage rule shall have zero or more business terms.

7.10.1.2 Basic Business Information Entity Usage Rule Localized Metadata

BBIE usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[B150] A BBIE usage rule shall have zero or more localized metadata classes.

[B151] Each occurrence of a BBIE usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[B152] BBIE usage rule localized metadata shall be in the language identified by the language and locale code.

7.10.2 Basic Business Information Entity Cardinality

Each BBIE, in its role as an ABIE property, will have its cardinality explicitly expressed.

[B153] Each BBIE shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[B154] BBIE cardinality values shall be non-negative integers of zero or greater, or – only in the case of maximum cardinality – the token **unbounded** if no limit applies.

[B155] BBIE cardinality values shall be a restriction of its basis BCC.

The BBIE minimum cardinality will never be smaller than the BCC minimum cardinality and the maximum cardinality will never be larger than the BCC maximum cardinality.

7.10.3 Basic Business Information Entity Sequencing Key

Business requirements may exist for BBIEs to occur in a specific order within an ABIE. Software and storage applications may have unique sequencing algorithms that change the normatively defined order of the BBIE within an ABIE. To ensure the desired order is preserved, each BBIE within an ABIE will be assigned a unique sequencing key.

[B156] Each BBIE shall be assigned a unique sequencing key within the ABIE of which it is a part.

[Note] – Sequencing Key Structure

There are no specific rules for the structure of the sequencing keys. Implementers are free to choose any structure providing it guarantees uniqueness within the ACC to which it belongs and the structuring scheme is readily available for anyone accessing or using the ACC.

Since BBIEs represent contextualized expressions of their source BCCs, the sequencing requirements of a BBIE in an ABIE might be different than the sequencing key of the corresponding BCC in an ACC.

[B157] A BBIE sequencing key may be different than its corresponding BCC sequencing key.

7.10.4 Basic Business Information Entity Common Information

In its role as an ABIE property, each BBIE has a common information class.

[B158] Each BBIE shall have a common information class.

[B159] The BBIE common information class shall conform to all BIE common information rules.

[Example] – Common Information

DEN – **`Trade_ Contract. Total_ Price. Amount`**

Definition – The monetary **`amount`** of the **`total price`** of this **`trade contract`**.

Business Term – **`Service Agreement Total Price`**; **`Amount Owed`**

[B160] The BBIE common information class shall consist of:

- **DEN (mandatory):** The official name of the BBIE.
- **Definition (mandatory):** The semantic meaning of the BBIE.
- **Business Term (optional, repetitive)**: A synonym term under which the BBIE is commonly known and used in business.

7.10.4.1 Basic Business Information Entity Dictionary Entry Names

[B161] Each BBIE DEN shall conform to all BIE DEN rules.

[B162] The DEN of a BBIE shall consist of the following components in the specified order:

- The object class term and qualifiers, if any, of the ABIE owning the corresponding BBIE, followed by a dot and space character.
- The DEN of the included BBIE property.

[Example] – BBIE DENs

Trade_ Contract. Total_ Price. Amount; Calculated_ Metrics. Description. Text

7.10.4.2 Basic Business Information Entity Definitions

[B163] Each BBIE definition shall conform to all BIE definition rules.

[B164] The definition of a BBIE shall include the object class term and qualifiers, if any of the ABIE to which it belongs, and the definition of the included BBIE property.

[B165] A BBIE with an unqualified property term shall have the same definition as the BCC the BBIE is based on.

[B166] A BBIE with a qualified property term shall have a definition that semantically restricts the definition of the less qualified BBIE or BCC that the BBIE is based on.

[Example] – BBIE Definition

Trade_ Contract. Total_ Price. Amount

Definition – The monetary **amount** of the **total price** of this **trade contract**.

Where the object class term and qualifier **Trade_ Contract**, property term and qualifier **Total_ Price**, and representation term **Amount** are in the definition.

7.10.4.3 Basic Business Information Entity Business Terms

A BBIE may have several business terms. BBIE business terms are synonym terms under which the BBIE is commonly known and used in business.

[B167] Each BBIE shall have zero or more business terms.

7.10.5 Basic Business Information Entity Localized Information

The BBIE localized information class contains the relevant information necessary to associate native language expressions of BBIE attributes to the BBIE.

[B168] A BBIE shall have zero or more localized information classes.

[B169] Each occurrence of a BBIE localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BBIE in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BBIE in a language other than English.

- **Business Term (optional, repetitive)**: A synonym term in another language under which the BBIE is commonly known and used in a business expression in that language.

BBIE localized information DENs should follow, as much as possible, all BBIE DEN rules.

[B170] Each BBIE localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[B171] Each BBIE localized information definition shall adhere to all BBIE definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[B172] Each BBIE localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[B173] Each BBIE localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

7.11 Basic Business Information Entity Properties

A BBIE property represents a generic reusable data element independent of an object class. BBIE property consists of a property term plus a representation term.

[Definition} – Basic Business Information Entity Property

A BBIE is a business information entity property for which the permissible values are expressed by simple values, represented by a data type.

[B174] A BBIE property shall be defined for each BBIE.

BBIE properties are reusable across all object classes and packages.

[Example] – Reuse of BBIE Properties in Multiple BBIEs

`Trade_ Contact. Type. Code` and **`Delivery_ Event. Type. Code`** may both exist.

To ensure consistency in use, BBIE properties are always based on an approved BDT in the *UN/CEFACT Data Type Catalogue*.

[B175] A BBIE property shall only use a BDT based on an approved CDT in the *UN/CEFACT Data Type Catalogue*.

7.11.1.1 Basic Business Information Entity Property – Property Term

Each BBIE property contains a property term. The property term of a BBIE property is a semantically meaningful name for a unique characteristic that can be used in an ABIE object class.

[B176] Each BBIE property shall have a property term.

[B177] The property term of a BBIE property may consist of more than one word.

[B178] A multi-worded property term of a BBIE property shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[Example] – Single versus Multiple Word Property Terms

Legal Classification. Code is not the same as **Legal. Code**

Legal Classification. Code is not the same as **Classification. Code**

Classification Legal. Code is not the same as **Legal Classification. Code**

7.11.1.2 Basic Business Information Entity Property – Property Term Qualifiers

The BBIE property qualifier term is a word or words which help define and differentiate a BBIE property from its associated BCC property and other BBIE properties. The BBIE property qualifier enhances the semantic meaning of the BBIE property DEN to reflect a restriction to the BBIE property concept, conceptual domain, content model or data value. BBIE properties can have one or more qualifier terms.

[B179] BBIE property terms may be qualified to reflect semantic meaning.

[B180] A qualified BBIE property shall be a restriction of its source BCC property or its higher level BBIE properties in a BBIE property hierarchy.

[B181] BBIE property qualifier terms shall precede the property term.

[Example] – Multi-qualified BBIE Properties

The Multi-qualified BBIE property

Applied_ Actual_ Conversion Rate. Date Time

qualifies the qualified BBIE property

Actual_ Conversion Rate. Date Time

which qualifies the BCC property

Conversion Rate. Date Time

Whereas the multi-word qualified

Transport Tax Basis_ Information. Amount

Qualifies the BCC property

Information. Amount

and not the qualified BBIE property

Basis_ Information. Amount

[B182] Each BBIE property qualifier term shall be followed by an underscore and a space character ( ).

[B183] A multi-worded BBIE property qualifier term shall have a unique semantic meaning compared to the words separately.

[B184] A BBIE property hierarchy shall be established when multiple qualifiers are used.

[B185] A qualified property term of a BBIE property DEN may be applied in its entirety as a qualifier for another property term to convey a semantic relationship between the objects providing the qualifier hierarchy is preserved.

7.11.1.3 Basic Business Information Entity Property Representation Term

Each BBIE property contains a representation term. The representation term is a semantically meaningful name that represents the value domain of the BBIE property and its associated BDT. UN/CEFACT defines the approved representation terms as part of the *UN/CEFACT Data Type Catalogue*.

If a BBIEs BDT is qualified, the data type qualifier should be used as part of the BBIE object class, object class qualifier term(s), property term, and/or property term qualifier term(s).

[B186] A representation term shall be defined for each BBIE property.

[B187] The name of the BBIE property representation term may consist of more than one word.

[B188] A multi-worded BBIE property representation term shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[B189] The name of the BBIE property representation term shall be one of the approved representation terms in the *UN/CEFACT Data Type Catalogue*.

The BDT or qualified BDT will be of the same CDT as the basis BCC property.

[B190] A BBIE property shall have a BDT that is based on the CDT of the corresponding BCC property.

7.11.2 Basic Business Information Entity Property Identifiers

In order to ensure uniqueness, every BBIE property will have assigned a:

- **Unique Identifier (mandatory):** The identifier that references the BBIE property in a unique and unambiguous way.
- **Version Identifier (mandatory):** An indication of the evolution over time of the BBIE property.

[B191] Each BBIE property shall have a unique identifier within the library of which it is a part.

[B192] Each version of a BBIE property shall have a unique version identifier within the library of which it is a part.

7.11.3 Basic Business Information Entity Property Common Information

[B193] Each BBIE property shall have a common information class.

[B194] The BBIE property common information class shall conform to all BIE common information rules.

[B195] The BBIE property common information class shall consist of:

- **DEN (mandatory):** The official name of the BBIE property.
- **Definition (mandatory):** The semantic meaning of the BBIE property.

- **Business Term (optional, repetitive)**: A synonym term under which the BBIE property is commonly known and used in business.

[Example] – BBIE Property Common Information

DEN – **`Total_ Price. Amount`**

Definition – A monetary **`amount`** of a **`total price`**

Business Term – **`Price`**, **`Amount Owed`**

7.11.3.1 Basic Business Information Entity Property Dictionary Entry Names

[B196] Each BBIE property shall have a DEN.

[B197] Each BBIE property DEN shall conform to all BIE DEN rules.

[B198] The name of a BBIE property shall consist of a property term and property term qualifiers, if any, followed by a dot, a space character, and a representation term.

[B199] The name of a BBIE property shall be unique within the context of an object class but may be reused across different object classes.

[Example] – Reuse of BBIE Properties in Multiple Object Classes

`Trade_ Contact. Type. Code` and **`Delivery_ Event. Type. Code`** may both exist.

7.11.3.2 Basic Business Information Entity Property Definitions

[B200] BBIE property definitions shall conform to all BIE definition rules.

[B201] The definition of a BBIE property shall include the property and representation term of the BBIE property.

[Example] – DEN for BBIE Properties

`Total_ Price. Amount`

Definition – A monetary **`amount`** of a **`total price`**

Where the property term **`Price`** and optional qualifier term **`Total`** and the representation term **`Amount`** appear in the definition.

7.11.3.3 Basic Business Information Entity Property Business Terms

A BBIE property may have several business terms. BBIE property business terms are synonym terms under which the BBIE property is commonly known and used in business.

[B202] Each BBIE property shall have zero or more business terms.

7.11.4 Basic Business Information Entity Property Localized Information

The BBIE property localized information class contains the relevant information necessary to associate native language expressions of BBIE property attributes to the BBIE property.

[B203] A BBIE property shall have zero or more localized information classes.

[B204] Each occurrence of a BBIE property localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BBIE property in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BBIE property in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BBIE property is commonly known and used in a business expression in that language.

BBIE property localized information DENs should follow, as much as possible, all BBIE property DEN rules.

[B205] Each BBIE property localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[B206] Each BBIE property localized information definition shall adhere to all BBIE definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[B207] Each BBIE property localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[B208] Each BBIE property localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8 Data Types

This section provides a detailed technical explanation of CCTS data types. The abstract UML diagram shown in Figure 8-1 represents a generic data type (DT) metamodel that is used by both the CDT and BDT metamodels.

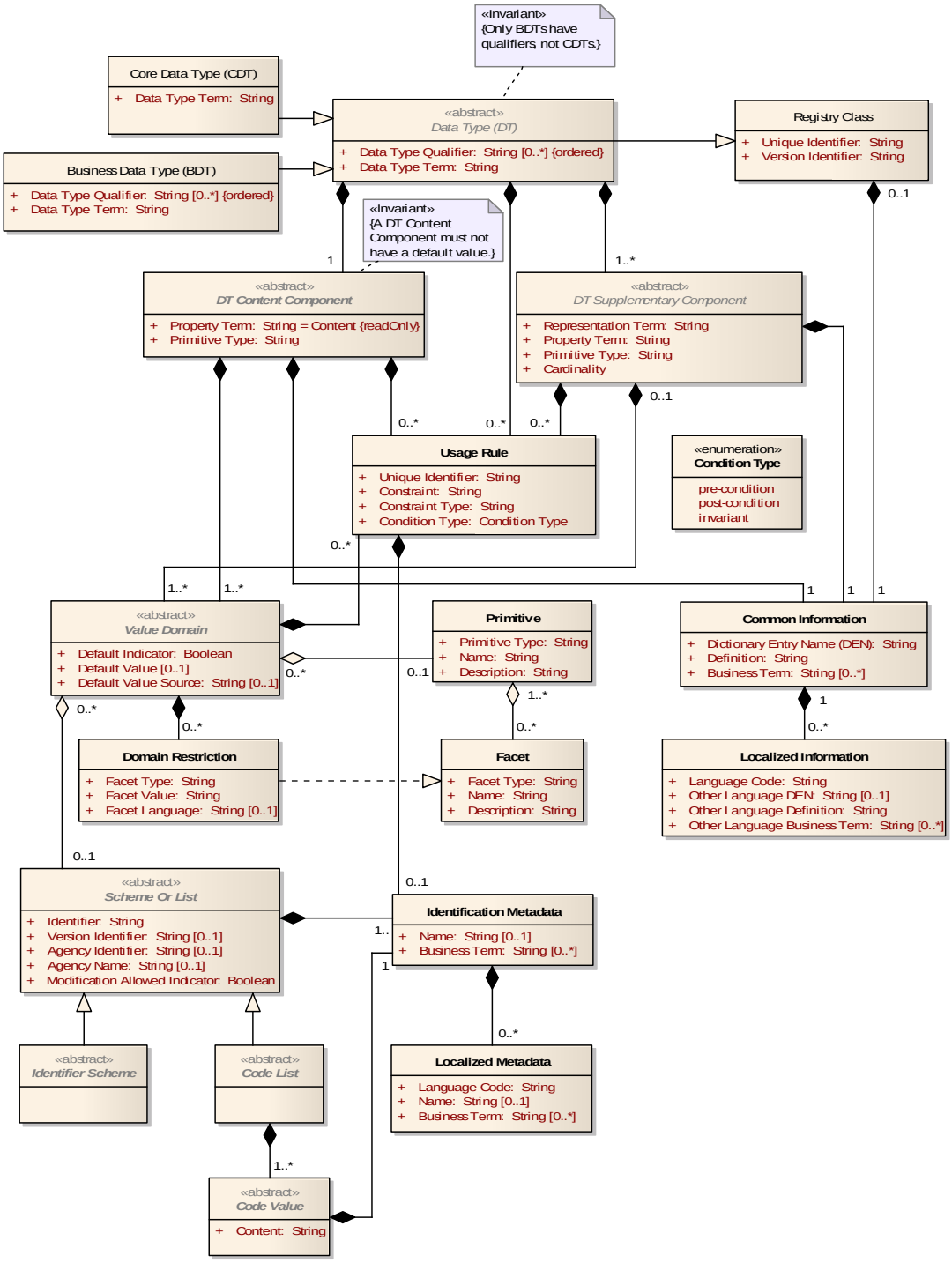


Figure 8-1. Abstract Data Type Metamodel

[Note] – Data Types

All references to data types in sections 8.1 through 8.5 refer to the generic concept that is the basis for the instantiated core and business data types. All rules in these sections apply to both core and business data types.]

Data types form the basis for defining the value domains of BCC and BBIE properties.

8.1 Overview

A data type defines the value domain – set of valid values – that can be used for a particular BCC property or BBIE property.

There are two categories of data Types (DTs)

- Core Data Type (CDT)
- Business Data Type (BDT)

[D1] A data type shall be a CDT or BDT.

8.2 Data Type Naming and Definition Conventions

The data type naming convention is based on CC and BIE naming and definition conventions to ensure consistency in the naming and defining of DT in their use with BCCs and BBIEs. The DT naming and definition conventions are derived from the guidelines and principles described in *ISO 11179 Part 4 – Definitions and ISO 11179 Part 5 – Naming and Identification Principles*.

The official language for UN/CEFACT DTs is English. All official dictionary entries will be in English. DT discovery work may very well occur in other languages; however official submissions for inclusion in the UN/CEFACT library must be in English. In order to ensure absolute clarity and understanding of the names and definitions it is essential to use words from the *Oxford English Dictionary*.

As with CCs and BIEs, a controlled vocabulary will be developed to identify the definition to be used for any DT terms that are potentially ambiguous.

[Note] – UN/CEFACT Controlled Vocabulary for Data Type Terms

Implementers are encouraged to use the UN/CEFACT controlled vocabulary as the authoritative source for DT terms.

8.3 Data Type -Registry Class

Data types are registry classes. Each DT registry class contains the following information:

- Unique Identifier
- Unique Version Identifier

[Note] – Structure of DT Identifiers

As with CCs and BIEs, there are no specific rules for the structures of the DT identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the library to which it belongs.

[D2] A registry class shall be created for each Data Type.

8.4 Data Type Common Information

The DT common information class provides necessary component information that is applicable to DTs either directly or through inheritance. The DT common information class contains the following information:

- **DEN** – this is the unique official name of the DT in the dictionary.
- **Definition –** this is the unique business semantic meaning of the DT.
- **Business Term(s) –** this is a synonym term under which the DT is commonly known and used in business. A DT may have several business terms.

[D3] Data Type common information content shall be in the English language following the *Oxford English Dictionary.* Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

8.4.1 Data Type Dictionary Entry Name

Data type naming rules are based on the following concepts as defined in ISO 11179:

- **Data Type Term –** defines the form of the set of valid values for a data element or value domain. It is the equivalent of the representation term of the BCC and BBIE, and their subordinate BCC and BBIE properties.

[D4] DT DENs shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[Note] – *Oxford English Dictionary*

The complete *Oxford English Dictionary* will be the authoritative source for conflict resolution between competing spellings of data type names or definitions.

[D5] A DT DEN shall be unique amongst all DENs within the library of which it is a part.

[D6] A DT DEN shall be extracted from the DT definition.

[D7] A DT DEN shall not include consecutive identical words.

[D8] A DT DEN and all its components shall be in singular form unless the concept itself is plural.

[D9] A DT DEN shall only use alphabetic characters plus the dot underscore and space characters.

[D10] A DT DEN shall only contain verbs, nouns, adverbs and adjectives unless a different part of speech is part of an official title, part of a term listed in the *Oxford English Dictionary*, or part of a controlled vocabulary.

[Note] – Parts of Speech

Articles, prepositions and related parts of speech that are not verbs, nouns, adverbs and adjectives normally add no semantic clarity and should not be used unless as part of an official title or in a controlled vocabulary as part of a common business term that can not otherwise be expressed.

[D11] Abbreviations and acronyms that are part of the DEN shall be expanded or explained in the definition.

[D12] The space character shall separate words in multi-worded DT data type qualifier and data type terms.

[D13] Each word in a DT DEN shall start with a capital letter.

[D14] The dots after DT terms shall be followed by a space character.

8.4.2 Data Type Definitions

Data Type definitions are based on the requirements for data element definitions defined in ISO 11179-4.

[D15] Each DT shall have its own unique semantic definition within the library of which it is a part.

[Note] – Order of Development of Definition and DEN

In the interest of quality, it is recommended that the definition be developed first and the DEN extracted from it.

[D16] The definition shall be in the English language following the latest version of the *Oxford English Dictionary*. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

[D17] The definition shall be consistent with the requirements of ISO 11179-4 and will provide an understandable meaning, which should also be translatable to other languages.

[D18] The definition shall take into account the fact that the users of the DT library are not necessarily native English speakers. It shall therefore contain short sentences, using normal words. Wherever synonym terms are possible, the definition shall use the preferred term as identified in the controlled vocabulary.

[D19] Whenever both the definite (i.e. **the**) and indefinite article (i.e. **a**) are possible in a definition, preference shall be given to an indefinite article (i.e. **a**).

[Note] – Definition Quality

To verify the quality of the definition, place the DEN followed by is before the definition to ensure that it is not simply a repetition of the DEN.

8.4.3 Business Terms

DT business terms are those terms commonly used for day-to-day information exchanges within a given domain. As such, no specific rules apply to business term structures. Interoperability of business terms will be given by linking them within the component common information class.

8.5 Data Type Localized Information Class

While the normative expressions of components are in the English language, implementers may choose to create alternative language expressions of definitions and business terms. While the normative expressions of DTs are in the English language, non-native English speakers may choose to create native language variations of the DEN, definition, and business term. The DT localized information class contains the relevant information necessary to associate the native language expressions to their normative English language counterparts. Other language DT DENs will only consist of alphabetic, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules. In addition to other language DEN, definition, and business term(s), a mandatory language code identifies the language in which the components are being expressed for storage in the registry. The localized information class contains:

- **Language Code –** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* will be used as the authoritative source for code values.
- **DEN –** The official name of the DT in a language other than English.
- **Definition –** the semantic meaning of the DT in a language other than English.
- **Business Term –** A synonym term in another language under which the DT is commonly known and used in a business expression in that language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language and locale codes. The business terms must only be expressed in the language identified by the language and locale code, or a recognized dialect of the language.

8.6 Core Data Types

A CDT defines the value domain for a BCC property. Figure 8-2 describes the CDT and relationships between the CDT and its subordinate parts.

[Definition] – Core Data Type (CDT)

A core data type is a data type consisting of one and only one core data type content component that carries the actual content, plus zero or more core data type supplementary components giving essential extra definition to the core data type content component. Core data types do not have business semantics.

UN/CEFACT publishes the approved CDTs in the *UN/CEFACT Data Type Catalogue*.

[D20] A CDT shall be one of the approved CDTs published in the *UN/CEFACT Data Type Catalogue*.

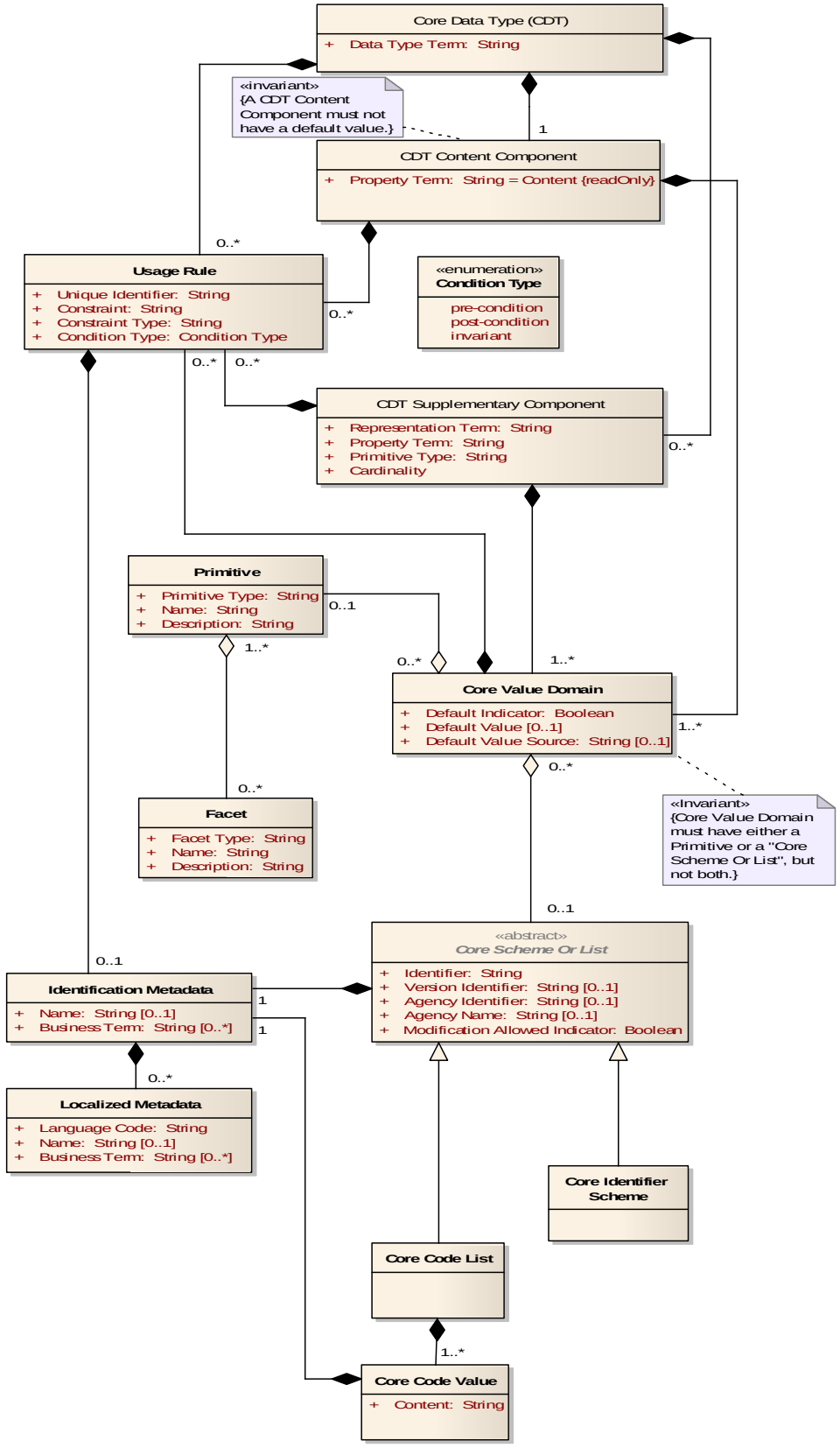


Figure 8-2. UML Diagram of Core Data Type Metamodel

8.6.1 Core Data Type – Data Type Term

The CDT is expressed by a data type term. The CDT data type term is a semantically meaningful name that serves as the basis for the DEN of the CDT and all BDTs derived from it. The CDT data type term defines the form of the set of valid values for a BCC property data element or value domain.

[D21] Each CDT shall have a unique data type term within the library of which it is a part.

[D22] Each CDT data type term shall semantically represent a value domain.

[D23] A data type term may have more than one word.

[D24] A multi-worded data type term must have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[D25] The CDT data type term shall be one of the terms specified in the list of permissible data type terms published in the *UN/CEFACT Data Type Catalogue*.

8.6.2 Core Data Type Usage Rules

CDTs may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the CDT. CDT usage rules represent the specific application of a CDT in its role of expressing the value domain of BCCs and BCC properties. CDT usage rules can be either unstructured – expressed as free form text, or structured –expressed in a formal language.

[D26] A CDT shall have zero or more usage rules.

CDT usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D27] CDT usage rules shall not replicate CDT content component, CDT supplementary component, or CDT core value domain usage rules.

[D28] A CDT usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The CDT usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D29] An unstructured CDT usage rule constraint shall have a free form text expression that fully details the usage rule.

If a CDT usage rule is expressed in a formal language, it will have a structured constraint. A structured constraint is a constraint that is expressed in a formal constraint language such as the UML OCL or OMG SBVR.

[D30] A structured CDT usage rule constraint shall have a formal constraint language expression.

CDT usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D31] Every CDT usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

CDT usage rules will also have a condition type that identifies when the constraint should be applied.

[D32] Every CDT usage rule shall have a condition type.

[D33] Every CDT usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.6.2.1 Core Data Type Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given CDT from all other usage rules for a library, a CDT usage rule may also have an identification metadata class that provides additional information.

[D34] A CDT usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates the usage rule from all other named usage rules for the CDT.

[D35] A CDT usage rule shall have zero or one names that is unique within the group of usage rules of a CDT.

The CDT usage rule identification metadata class may contain business terms. CDT usage rule business terms are synonym terms under which the CDT usage rule is commonly known and used in business.

[C36] Each ACC usage rule shall have zero or more business terms.

8.6.2.2 Core Data Type Usage Rule Localized Metadata

CDT usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D37] A CDT usage rule shall have zero or more localized information classes

[D38] Each occurrence of a CDT usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[C39] CDT usage rule localized metadata shall be in the language identified by the language and locale code.

8.6.3 Core Data Type Identifiers

In order to ensure uniqueness, every CDT will have assigned a:

- **Unique Identifier (mandatory):** The identifier that references the CDT in a unique and unambiguous way.
- **Version Identifier (mandatory):** An indication of the evolution over time of the CDT.

[D40] Each CDT shall have a unique identifier within the library of which it is a part.

[D41] Each version of a CDT shall have a unique version identifier within the library of which it is a part.

8.6.4 Core Data Type Common Information

[D42] Each CDT shall have a common information class.

[D43] The CDT common information class shall consist of:

- **DEN (mandatory):** The official name of the CDT.
- **Definition (mandatory):** The semantic meaning of the CDT.
- **Business Term (optional, repetitive)**: A synonym term under which the CDT is commonly known and used in business.

[Example] – CDT Common Information

DEN – **`Amount. Type`**

Definition – An **`amount`** is a number of monetary units specified in a currency

Business Term – **`Total Money`**, **`Sum of Money`**, **`Price`**, **`Monetary Value`**

8.6.4.1 Core Data Type Dictionary Entry Names

The CDT DEN is based on the ISO 11179 data type term.

[D44] Each CDT DEN shall conform to all DT DEN rules.

[D45] The CDT DEN shall consist of the data type term, plus a dot, a space character, and the term **`type`**.

[Example] – Core Data Type DENs

`Amount. Type`; **`Date Time. Type`**; **`Identifier. Type`**

8.6.4.2 Core Data Type Definitions

[D46] CDT definitions shall conform to all rules for DT definitions.

[D47] The CDT definition shall include the CDT data type term.

8.6.4.3 Core Data Type Business Terms

A CDT may have several business terms. CDT business terms are those terms commonly used for day-to-day information exchanges within a given domain.

[D48] A CDT shall have zero or more business terms.

8.6.5 Core Data Type Localized Information

The CDT localized information class contains the relevant information necessary to associate native language expressions of CDT attributes to the CDT.

[D49] A CDT shall have zero or more localized information classes.

[D50] Each occurrence of a CDT localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of a CDT in a language other than English.
- **Definition (mandatory):** The semantic meaning of the CDT in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the CDT is commonly known and used in a business expression in that language. Business terms in the localized information class shall be in the language identified by the language and locale code.

CDT localized information DENs should follow, as much as possible, all CDT DEN rules.

[D51] Each CDT localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[D52] Each CDT localized information definition shall adhere to all CDT definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D53] Each CDT localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D54] Each CDT localized information language business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.6 Core Data Type Content Component

CDT content components are defined in the *UN/CEFACT Data Type Catalogue* and are unique to the CDT to which they are assigned.

[D55] A CDT shall have one and only one CDT Content Component.

[D56] A CDT Content Component shall be the specified CDT content component as defined in the *UN/CEFACT Data Type Catalogue*.

8.6.6.1 Core Data Type Content Component Property Term

The CDT content component property term represents the actual content of a data element. The CDT Content Component property term has a fixed value of **`content`**.

[D57] Each CDT content component shall have a property term.

[D58] The CDT content component property term shall have a fixed value of **`Content.`**

8.6.6.2 Core Data Type Content Component Usage Rules

CDT content components may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the CDT content component. The CDT content component usage rules represent the specific application of a CDT content component in its role of expressing the value domain of its CDT. CDT usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D59] A CDT content component shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D60] CDT content component usage rules shall not replicate CDT, CDT supplementary component, or CDT core value domain usage rules.

[D61] A CDT content component usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The CDT content component usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D62] An unstructured CDT content component usage constraint rule shall have a free form text expression that fully details the usage rule.

A CDT content component formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D63] A structured CDT content component usage rule constraint shall have a formal constraint language expression.

CDT content component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D64] Every CDT content component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

CDT content component usage rules will also have a condition type that identifies when the formal constraint should be applied.

[D65] Every CDT content component usage rule shall have a condition type.

[D66] Every CDT content component usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.6.6.2.1 Core Data Type Content Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given CDT content component from all other usage rules for a library, a CDT content component usage rule may also have an identification metadata class that provides additional information.

[D67] A CDT content component usage rule shall have zero or one identification metadata classes.

The CDT content component usage rule identification metadata class contains a unique name that semantically differentiates it from all other named usage rules for the CDT.

[D68] A CDT content component usage rule shall have zero or one names that is unique within the group of usage rules of a CDT content component.

The CDT content component usage rule metadata class may contain business terms. CDT content component usage rule business terms are synonym terms under which the CDT content component usage rule is commonly known and used in business.

[D69] Each CDT usage rule shall have zero or more business terms.

8.6.6.2.2 Core Data Type Content Component Usage Rule Identification Metadata Localized Information

CDT Content component usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D70] A CDT content component usage rule shall have zero or more localized information classes.

[D71] Each occurrence of a CDT content component usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[D72] CDT content component usage rule localized metadata shall be in the language identified by the language and locale code.

8.6.6.3 Core Data Type Content Component Common Information

Each CDT content component has a common information class.

[D73] Each CDT content component shall have a common information class.

[D74] The CDT content component common information class shall consist of:

- **DEN (mandatory):** The official name of a CDT content component.
- **Definition (mandatory):** The semantic meaning of a CDT content component.
- **Business Term (optional, repetitive)**: A synonym term under which the CDT content component is commonly known and used in business.

[Example] – CDT Content Component Common Information

DEN – **`Amount. Content`**

Definition – An **`amount`** is a number of monetary units

Business Term – **`Money`**

8.6.6.3.1 Core Data Type Content Component Dictionary Entry Names

The CDT content component DENs are based on ISO 11179 defined data type and property terms.

[D75] Each CDT content component DEN shall conform to all DT DEN rules.

[D76] The DEN of a CDT content component shall consist of the data type term of the CDT to which it is assigned, plus a dot, space character, and the property term **`content`**.

[Example] – Core Data Type Content Components

`Amount. Content`; **`Date Time. Content`**

8.6.6.3.2 Core Data Type Content Component Definition

[D77] Each CDT content component definition shall conform to all DT definition rules.

[D78] The CDT content component definition shall include the primitive type term and the definition of the source representation term.

8.6.6.3.3 Core Data Type Content Component Business Terms

A CDT content component may have several business terms. CDT content component business terms are synonym terms under which the CDT content component is commonly known and used in business.

[D79] A CDT content component shall have zero or more business terms.

8.6.6.4 Core Data Type Content Component Localized Information

The CDT content component localized information class contains the relevant information necessary to associate native language expressions of CDT content components to the CDT content component.

[D80] Each CDT content component shall have zero or more localized information classes.

[D81] Each occurrence of a CDT content component localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the CDT content component in a language other than English
- **Definition (mandatory):** The semantic meaning of the CDT content component in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the CDT content component is commonly known and used in a business expression in that language.

CDT content component localized information DENs should follow, as much as possible, all CDT content component DEN rules.

[D82] Each CDT content component localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[D83] Each CDT content component localized information definition shall adhere to all CDT content component definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D84] Each CDT content component localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D85] Each CDT content component localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.6.5 Core Data Type Content Component Core Value Domain

CDT content components can have one or more value domains. A CDT content component core value domain is an abstract class that defines the set of allowed values through the presence of either a scheme or list, or a primitive with defined facets and their restrictions.

[D86] A CDT content component shall have one or more value domains.

Since each CDT content component may have multiple value domains, each defined value domain contains a default indicator that identifies it as the default value domain amongst the set of value domains for the CDT content component.

[D87] A CDT content component core value domain shall have a default indicator whose value = **`true`** if it is the default value domain.

[D88] A CDT content component core value domain shall have a default indicator whose value = **`false`** if it is not the default value domain.

Each CDT content component core value domain may also have a default value. This default value represents a CDT content component core value domain value that is to be automatically applied to the CDT content component in the absence of a choice made by the user.

[D89] A CDT content component core value domain shall have zero or one default values.

Default values will be conformant to the defined primitive or scheme or list of the CDT content component core value domain.

[D90] the CDT content component core value domain default value shall be conformant to its defined primitive or scheme or list.

If a CDT content component value domain has a default value, the source of that default value will also be identified.

[D91] A CDT core value domain with a default value shall have a default value source.

[Note] – Value Domain Default Sources

There are no specific rules for the structure of value domain default value sources. Implementers are free to choose any structure providing ti guarantees uniqueness within the group of value domain default value sources of a value domain.

CDT content component core value domains are defined by either a primitive or a scheme or list. Each primitive or scheme or list constitutes a separate core value domain for the CDT content component.

[D92] Each CDT content component core value domain shall consist of a primitive or a scheme or list.

8.6.6.5.1 Core Data Type Content Component Core Value Domain Primitive

Primitives represent basic building blocks for defining value domains of content and supplementary components. Each content component core value domain can have zero or one primitives defined for it. The CDT content component core value domain primitive defines the value domain. Primitives are referred to as primitive types.

Primitives include, but are not limited to:

- Binary
- Date
- Decimal
- Double

- Float
- Integer
- String
- Token

[D93] Each CDT content component core value domain shall have zero or one primitives.

[D94] A CDT content component primitive shall be one of the defined primitives in the *UN/CEFACT Data Type Catalogue*.

Each primitive shall have a primitive type. The primitive type code value is taken from a primitive type code list.

[D95] Every CDT content component core value domain primitive shall have a primitive type taken from a primitive type code list.

[Note] –Primitive Type Code List

UN/CEFACT will publish and make freely available a Primitive Type Code List for use in support of this rule.

Each primitive has a formal name. This name typically represents the nature of the value domain it represents.

[D96] Every CDT content component core value domain primitive shall have a primitive name.

[D97] Every CDT content component core value domain primitive name shall be unique within the set of primitives of CDTs.

A CDT content component core value domain primitive will also have a description that semantically defines its value domain.

[D98] Each CDT content component core value domain primitive shall have a description that semantically defines its value domain.

8.6.6.5.1.1 Core Data Type Content Component Core Value Domain Primitive Facet

The value domains expressed by primitives are quantified through their facets. A primitive may have one or more facets. Each facet defines or constrains an aspect of the value domain expressed by the primitive.

[D99] Each CDT content component core value domain primitive shall have zero or more facets.

Each facet shall have a facet type. The facet type code value is taken from a facet type code list.

[D100] Each CDT content component core value domain primitive facet shall have a facet type taken from a facet type code list.

[Note] –Facet Type Code List

UN/CEFACT will publish and make freely available a Facet Type Code List for use in support of this rule.

[D101] Each CDT content component core value domain primitive facet shall have a name that is unique amongst the set of facet names of a primitive.

[D102] Each CDT content component core value domain primitive facet shall have a description that semantically expresses the nature of the restrictions associated with it.

8.6.6.5.2 Core Data Type Content Component Core Value Domain Scheme or List

Schemes are the equivalent of a pattern facet. A scheme formally expresses the pattern and the allowed values for populating that pattern in the form of identifiers. Lists are the equivalent of enumerated lists and are typically published as formal code lists. The set of codes in a formal code list is used by core value domains as an enumerated set of allowed values.

Unambiguous identification of the scheme or list is necessary.

[D103] Every CDT content component core value domain scheme or list shall have an identifier.

A version identifier serves to differentiate one version of a scheme or list from all other versions of the scheme or list.

[D104] Every CDT content component core value domain scheme or list shall have zero or one version identifiers.

Every scheme or list will be owned by an organization. The organization may either identified by a unique identifier or a name.

[D105] Every CDT content component core value domain scheme or list shall have either an agency identifier or an agency name.

[D106] Every CDT content component core value domain scheme or list shall have zero or one agency identifiers.

[Note] – Agency Identifier

UN/CEFACT recommends using UN/CEFACT Agency Identifier Code List (Data Element 3055) in the latest version of the UN/CEFACT directory.

[D107] Every CDT content component core value domain scheme or list shall have zero or one agency names.

Business Data Types are able to place restrictions on schemes and lists. If such restrictions are undesireable, then this will be indicated through the use of a required modification allowed indicator.

[D108] Every CDT content component core value domain scheme or list shall have a modification allowed indicator whose value = `true` if modifications are allowed, or whose value = `false` if modifications are not allowed.

8.6.6.5.2.1 Core Data Type Content Component Core Scheme or List Identification Metadata

Although the identifier is sufficient to differentiate one core scheme or list for a given CDT content component core value domain from all other schemes or lists for that CDT content component, a CDT content component core scheme or list may also have an identification metadata class that provides additional information.

[D109] A CDT content component core scheme or list shall have zero or one identification metadata classes.

The CDT content component core scheme or list identification metadata class contains a unique name that semantically differentiates it.

[D110] A CDT content component core scheme or list shall have zero or one names that is unique within the group of core scheme or lists of a CDT.

The CDT content component core scheme or list identification metadata class may contain business terms. These business terms are synonym terms under which the scheme or list is commonly known and used in business.

[D111] Each CDT content component core value domain core scheme or list shall have zero or more business terms.

8.6.6.5.2.2 Core Data Type Content Component Core Value Domain Core Scheme or List Identification Metadata Localized Information

CDT content component core value domain core schemes or lists may have localizedsiness term or terms.

[D112] A CDT content component core value domain core scheme or list shall have zero or more localized information classes

[D113] Each occurrence of a CDT content component core value domain core scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D114] CDT content component core value domain core scheme or list localized metadata shall be in the language identified by the language and locale code.

8.6.6.5.2.3 Core Data Type Content Component Core Value Domain Core Identifier Scheme

Core identifier schemes are typically not enumerated. No additional rules are provided regarding the content of identifier schemes. However, at a minimum, an identifier scheme should define a specific pattern for the values of the identifiers to conform to.

8.6.6.5.2.4 Core Data Type Content Component Core Value Domain Core Code List

Core code lists contain lists of enumerated code values. However, the diversity of code list content is such that no additional rules are provided with the exception of the requirement for one or more code values.

[D115] Each CDT content component core value domain core scheme or list shall contain one or more core code values.

8.6.6.5.2.4.1 Core Data Type Content Component Core Value Domain Core Code Value Identification Metadata

A CDT content component core code value will contain an identification metadata class consisting of a name and optional business term or terms.

[D116] A CDT content component core value domain core code list core code value shall have zero or one identification metadata classes.

[D117] A CDT content component core value domain core code list core code value shall have zero or one names that is unique within the set of core code values for a core code list.

The CDT content component core code list identification metadata class may also contain several business terms. These business terms are synonym terms under which the core code value is commonly known and used in business.

[D118] Each CDT content component core code value shall have zero or more business terms.

8.6.6.5.2.4.2 Core Data Type Content Component Core Value Domain Core Code Value Localized Metadata

CDT content component core value domain core scheme or list may have localized metadata that is used to provide other language expressions of its name and business terms.

[D119] A CDT content component core value domain core scheme or list shall have zero or more localized metadata classes.

[D120] Each occurrence of a CDT content component core value domain core scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D121] CDT content component core value domain core scheme or list localized metadata shall be in the language identified by the language and locale code.

8.6.6.5.3 Core Data Type Content Component Core Value Domain Usage Rule

CDT content component core value domains may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the CDT content component value domain. CDT content component core value domain usage rules represent the specific application of a CDT content component value domain in its role of expressing the value domain of its CDT content component. CDT content component core value domain usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D122] A CDT content component core value domain shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D123] CDT content component core value domain usage rules shall not replicate CDT, CDT content component, CDT supplementary component or CDT supplementary component core value domain usage rules.

[D124] A CDT content component core value domain usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The CDT content component core value domain usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D125] An unstructured CDT content component core value domain usage rule constraint shall have a free form text expression that fully details the usage rule.

A CDT content component core value domain formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D126] A structured CDT content component core value domain usage rule constraint shall have a formal constraint language expression.

CDT content component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D127] Every CDT content component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

CDT content component usage rules will also have condition types that identify when the formal constraint should be applied.

[D128] Every CDT content component core value domain usage rule shall have a condition type.

[D129] Every CDT content component core value domain usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.6.6.5.3.1 Core Data Type Content Component Core Value Domain Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule in a library for all other usage rules, a CDT content component core value domain usage rule may also have an identification metadata class that provides additional information.

[D130] A CDT content component core value domain usage rule shall have zero or one identification metadata classes.

The CDT content component core value domain usage rule contains a unique name that semantically differentiates the usage rule from all other named usage rules for the CDT.

[D131] A CDT content component core value domain usage rule shall have zero or one names that is unique within the group of usage rules of a CDT.

The CDT content component core value domain usage rule identification metadata class may contain business terms. CDT content component core value domain Business terms are synonym terms under which the CDT content component core value domain usage rule is commonly known and used in business.

[D132] Each CDT content component core value domain usage rule shall have zero or more business terms.

8.6.6.5.3.2 Core Data Type Content Component Core Value Domain Usage Rule Identification Metadata Localized Information

CDT content component core value domain usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[B133] A CDT content component core value domain usage rule shall have zero or more localized metadata classes.

[B134] Each occurrence of a CDT content component core value domain usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[C135] CDT content component core value domain usage rule localized metadata shall be in the language identified by the language and locale code.

8.6.7 Core Data Type Supplementary Components

CDT supplementary components are defined and published in the UN/CEFACT Data Type Catalogue, and are unique to the CDT to which they are assigned. A CDT will have zero or more CDT supplementary components.

[D136] A CDT shall have zero or more CDT supplementary components.

[D137] A CDT supplementary component shall be one of the specified CDT supplementary components as defined in the UN/CEFACT Data Type Catalogue.

8.6.7.1 Core Data Type Supplementary Component Property Term

Each CDT supplementary component contains a property term. The CDT supplementary component property term is a semantically meaningful name for a unique characteristic that can be used in a CDT.

[D138] Each CDT supplementary component shall have a property term.

[D139] The CDT supplementary component property term may consist of more than one word.

[D140] A multi-worded CDT supplementary component property term must have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

8.6.7.2 Core Data Type Supplementary Component Representation Term

Each CDT supplementary component contains a representation term. The representation term is a semantically meaningful name that represents the value domain of the supplementary component. UN/CEFACT defines the approved representation terms as part of the UN/CEFACT Data Type Catalogue.

[D141] A representation term shall be defined for each CDT supplementary component.

[D142] The name of the CDT supplementary component representation term may consist of more than one word.

[D143] A multi-worded CDT supplementary component representation term shall have a unique semantic meaning compared to the words separately and compared to any other combination of these words.

[D144] The name of the CDT supplementary component representation term shall be one of the approved representation terms in the UN/CEFACT Data Type Catalogue.

8.6.7.3 Core Data Type Supplementary Component Cardinality

Each CDT supplementary component will have its cardinality explicitly expressed. The supplementary component cardinality defines the occurrence requirements of the supplementary component within its data type.

[D145] Each CDT supplementary component shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[D146] CDT supplementary component cardinality shall be equal to `[0..1]` if the CDT supplementary component is optional, or `[1..1]` if mandatory.

8.6.7.4 Core Data Type Supplementary Component Usage Rules

A CDT supplementary component may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the CDT supplementary component. The CDT supplementary component usage rules represent the specific application of a CDT supplementary component in its role of expressing the value domain of its CDT. CDT supplementary component usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D147] A CDT supplementary component shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D148] CDT supplementary component usage rules shall not replicate CDT, CDT content component, or CDT core value domain usage rules.

[D149] A CDT supplementary component usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The CDT supplementary component usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D150] An unstructured CDT supplementary component usage constraint rule shall have a free form text expression that fully details the usage rule.

A CDT supplementary component formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D151] A structured CDT supplementary component usage rule constraint shall have a formal constraint language expression.

CDT supplementary component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D152] Every CDT supplementary component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

CDT supplementary component usage rules will also have a condition type that identifies when the formal constraint should be applied.

[D153] Every CDT supplementary component usage rule shall have a condition type.

[D154] Every CDT supplementary component usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.6.7.4.1 Core Data Type Supplementary Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given CDT supplementary component from all other usage rules in a library, a CDT supplementary component usage rule may also have a unique name that semantically differentiates it from all other named usage rules for the CDT.

[D155] A CDT supplementary component usage rule shall have zero or one names that is unique within the group of usage rules of a CDT supplementary component.

A CDT supplementary component usage rule may have several business terms. CDT supplementary component usage rule business terms are synonym terms under which the CDT usage rule is commonly known and used in business.

[D156] Each CDT usage rule shall have zero or more business terms.

8.6.7.4.2 Core Data Type Supplementary Component Usage Rule Identification Metadata Localized Information

The CDT supplementary component usage rule localized information class contains the relevant information necessary to associate native language expressions of usage rules.

[D157] A CDT supplementary component usage rule shall have zero or more localized information classes

[D158] Each occurrence of a usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

Core Data Type supplementary component localized information names should follow, as much as possible, all CDT supplementary component DEN rules.

[D159] Each usage rule localized information name shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

The usage rule name in the localized information class must only be expressed in the language identified by the language code property of that class.

[D160] Each usage rule name shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D161] Each usage rule localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.7.5 Core Data Type Supplementary Component Common Information

Each CDT supplementary component has a common information class.

[D162] Each CDT supplementary component shall have a common information class.

[D163] The CDT supplementary component common information class shall consist of:

- **DEN (mandatory):** The official name of a CDT supplementary component.
- **Definition (mandatory):** The semantic meaning of the CDT supplementary component.
- **Business Term (optional, repetitive)**: A synonym term under which the CDT supplementary component is commonly known and used in business.

[Example] – CDT Supplementary Component Common Information

DEN – **`Amount. Currency Code List Agency. Identifier`**

Definition – The **`identifier`** of the **`agency`** that maintains the **`currency code list`** used for the **`amount`**.

Business Term – **`Currency Code Owner`**

8.6.7.5.1 Core Data Type Supplementary Component Dictionary Entry Names

The CDT supplementary component DENs are based on ISO 11179 defined data type and property terms.

[D164] Each CDT supplementary component DEN shall conform to all DT DEN rules.

[D165] The DEN of a CDT supplementary component shall consist of the following parts in the order specified:

- Data Type term of the CDT to which it belongs, followed by a dot and space character (.).
- Property term which expresses the unique characteristic of the CDT supplementary component, followed by a dot and space character (.).
- Representation term which represents the value domain of the content of the CDT supplementary component.

[Example] – Core Data Type Supplementary components

Amount. Currency Code List Version. Identifier; **Code. List Agency. Identifier**; **Quantity. Unit. Code**

[D166] The CDT supplementary component DEN shall be unique amongst all CDT supplementary component names within the library of which it is a part.

8.6.7.5.2 Core Data Type Supplementary Component Definition

A CDT supplementary component definition provides a clear, unambiguous and complete explanation of the meaning of a CDT supplementary component and its relevance for the related CDT.

[D167] Each CDT supplementary component definition shall conform to all DT definition rules.

[D168] The definition of a CDT supplementary component shall include the data type term of the CDT to which it belongs, the property term and the representation term.

8.6.7.5.3 Core Data Type Supplementary Component Business Terms

CDT supplementary components may have business terms. CDT supplementary component business terms are synonyms commonly used for day-to-day information exchanges within a given domain.

[D169] Each CDT supplementary component shall have zero or more business terms.

8.6.7.6 Core Data Type Supplementary Component Localized Information

The CDT supplementary component localized information class contains the relevant information necessary to associate native language expressions of CDT supplementary components to the CDT supplementary component.

[D170] Each CDT supplementary component shall have zero or more localized information classes.

[D171] Each occurrence of a CDT supplementary component localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the CDT supplementary component in a language other than English
- **Definition (mandatory):** The semantic meaning of the CDT supplementary component in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the CDT supplementary component is commonly known and used in a business expression in that language.

CDT supplementary component localized information DENs should follow, as much as possible, all CDT supplementary component DEN rules.

[D172] Each CDT supplementary component localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[D173] Each CDT supplementary component localized information definition shall adhere to all CDT supplementary component definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D174] Each CDT supplementary component localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D175] Each CDT supplementary component localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.7.7 Core Data Type Supplementary Component Core Value Domain

CDT supplementary components can have one or more value domains. A CDT supplementary component core value domain defines the set of allowed values through the presence of either a scheme or list, or a primitive with defined facets and their restrictions.

[D176] A CDT supplementary component shall have one or more value domains.

Since each CDT supplementary component may have multiple value domains, each defined value domain contains a default indicator that identifies it as the default value domain amongst the set of value domains for the CDT supplementary component.

[D177] A CDT supplementary component core value domain shall have a default indicator whose value = **`true`** if it is the default value domain.

[D178] A CDT supplementary component core value domain shall have a default indicator whose value = **`false`** if it is not the default value domain.

Each CDT supplementary component core value domain may also have a default value. This default value represents a CDT supplementary component value that is to be automatically applied to the CDT supplementary component in the absence of a choice made by the user. The default value should be used unless an alternate is required to meet specific business requirements.

[D179] A CDT supplementary component core value domain shall have zero or one default values.

Example – CDT Supplementary Component Default Value

Supplementary Component – **`Amount. Currency Code List. Identifier`**

Default Value – **`ISO 4217`**

Default values will be conformant to the defined primitive or scheme or list of the CDT supplementary component core value domain.

[D180] the CDT supplementary component core value domain default value shall be conformant to its defined primitive or scheme or list.

If a CDT supplementary component value domain has a default value, the source of that default value will also be identified.

[D181] A CDT value domain with a default value shall have a default value source.

[Note] – Value Domain Default Source

There are no specific rules for the structure of value domain default value sources. Implementers are free to choose any structure providing it guarantees uniqueness within the group of value domain default value sources of a value domain.

CDT supplementary component core value domains are defined by either a primitive or a scheme or list. Each primitive or scheme or list constitutes a separate core value domain for the CDT supplementary component.

[D182] Each CDT supplementary component core value domain shall consist of a primitive or a scheme or list.

8.6.7.7.1 Core Data Type Supplementary Component Core Value Domain Primitive

Primitives represent basic building blocks for defining value domains of content and supplementary components. Each CDT supplementary component core value domain can have zero or one primitives defined for it. The CDT supplementary component core value domain primitive defines the value domain. Primitives are referred to as primitive types.

Primitives include, but are not limited to:

- Binary
- Date
- Decimal
- Double
- Float
- Integer
- String
- Token

[D183] Each CDT supplementary component core value domain shall have zero or one primitives.

[D184] A CDT supplementary component primitive shall be one of the defined primitives in the *UN/CEFACT Data Type Catalogue*.

Each primitive shall have a primitive type. The primitive type code value is taken from a primitive type code list.

[D185] Every CDT supplementary component core value domain primitive shall have a primitive type taken from a primitive type code list.

[Note] –Primitive Type Code List

UN/CEFACT will publish and make freely available a Primitive Type Code List for use in support of this rule.

Each primitive has a formal name. This name typically represents the nature of the value domain it represents.

[D186] Every CDT supplementary component core value domain primitive shall have a primitive name.

[D187] Every CDT supplementary component core value domain primitive name shall be unique within the set of primitives of CDTs.

A CDT supplementary component core value domain primitive will also have a description that semantically defines its value domain.

[D188] Each CDT supplementary component core value domain primitive shall have a description that semantically defines its value domain.

8.6.7.7.1.1 Core Data Type Supplementary Component Core Value Domain Primitive Facet

The value domains expressed by primitives are quantified through their facets. A primitive may have one or more facets. Each facet defines or constrains an aspect of the value domain expressed by the primitive.

[D189] Each CDT supplementary component core value domain primitive shall have zero or more facets.

Each facet shall have a facet type. The facet type code value is taken from a facet type code list.

[D190] Each CDT supplementary component core value domain primitive facet shall have a facet type taken from a facet type code list.

[Note] –Facet Type Code List

UN/CEFACT will publish and make freely available a facet type code list for use in support of this rule.

[D191] Each CDT supplementary component core value domain primitive facet shall have a name that is unique amongst the set of facet names of a primitive.

[D192] Each CDT supplementary component core value domain primitive facet shall have a description that semantically expresses the nature of the restrictions associated with it.

8.6.7.7.2 Core Data Type Supplementary Component Core Value Domain Scheme or List

Schemes are the equivalent of a pattern facet. A scheme formally expresses the pattern and the allowed values for populating that pattern in the form of identifiers. Lists are the equivalent of enumerated lists and are typically published as formal code lists. The set of codes in a formal code list is used by core value domains as an enumerated set of allowed values.

Unambiguous identification of the scheme or list is necessary.

[D193] Every CDT supplementary component core value domain scheme or list shall have an identifier.

A version identifier serves to differentiate one version of a scheme or list from all other versions of the scheme or list.

[D194] Every CDT supplementary component core value domain scheme or list shall have zero or one version identifiers.

Every scheme or list will be owned by an organization. The organization may either identified by a unique identifier or a name.

[D195] Every CDT supplementary component core value domain scheme or list shall have either an agency identifier or an agency name.

[D196] Every CDT supplementary component core value domain scheme or list shall have zero or one agency identifiers.

[Note] – Agency Identifier

UN/CEFACT recommends using UN/CEFACT Code List Responsible Agency Code (Data Element 3055) in the latest version of the UN/CEFACT directory.

[D197] Every CDT supplementary component core value domain scheme or list shall have zero or one agency names.

Business Data Types are able to place restrictions on schemes and lists. If such restrictions are undesireable, then this will be indicated through the use of a required modification allowed indicator.

[D198] Every CDT supplementary component core value domain scheme or list shall have a modification allowed indicator whose value = `true` if modifications are allowed, or whose value = `false` if modifications are not allowed.

8.6.7.7.2.1 Core Data Type Supplementary Component Core Scheme or List Identification Metadata

Although the identifier is sufficient to differentiate one core scheme or list for a given CDT supplementary component core value domain from all other schemes or lists for that CDT supplementary component, a CDT supplementary component core scheme or list may also have a unique name that semantically differentiates it.

[D199] A CDT supplementary component core scheme or list shall have zero or one names that is unique within the group of core scheme or lists of a CDT supplementary component.

A CDT supplementary component core scheme or list may have several business terms. These business terms are synonym terms under which the scheme or list is commonly known and used in business.

[D200] Each CDT supplementary component core value domain core scheme or list shall have zero or more business terms.

8.6.7.7.2.2 Core Data Type Supplementary Component Core Value Domain Core Scheme or List Identification Metadata Localized Information

The CDT supplementary component core value domain core scheme or list localized information class contains the relevant information necessary to associate native language identification of schemes or lists.

[D201] A CDT supplementary component core value domain core scheme or list shall have zero or more localized information classes

[D202] Each occurrence of a CDT supplementary component core value domain core scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

CDT supplementary component core value domain core scheme or list names should follow, as much as possible, all CDT DEN rules.

[D203] Each CDT supplementary component core value domain core scheme or list name shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

The CDT supplementary component core value domain core scheme or list name in the localized information class must only be expressed in the language identified by the language code property of that class.

[D204] Each supplementary component core value domain core scheme or list name shall be in the language identified by the language and locale code.

The CDT supplementary component core value domain core scheme or list business term must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D205] Each CDT supplementary component core value domain core scheme or list business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.7.7.2.3 Core Data Type Supplementary Component Core Value Domain Core Identifier Scheme

Core identifier schemes are typically not enumerated. No additional rules are provided regarding the content of identifier schemes. However, at a minimum, an identifier scheme should define a specific pattern for the values of the identifiers to conform to.

8.6.7.7.2.4 Core Data Type Supplementary Component Core Value Domain Core Code List

Core code lists contain lists of enumerated code values. However, the diversity of code list content is such that no additional rules are provided with the exception of the requirement for one or more code values.

[D206] Each CDT supplementary component core value domain core scheme or list shall contain one or more core code values.

8.6.7.7.2.4.1 Core Data Type Supplementary Component Core Value Domain Core Code Value Identification Metadata

A CDT supplementary component core code value will contain identification metadata consisting of a name and optional business term or terms.

[D207] A CDT supplementary component core value domain core code list core code value shall have zero or one names that is unique within the set of core code values for a core code list.

A CDT supplementary component core code list may also have several business terms. These business terms are synonym terms under which the core code value is commonly known and used in business.

[D208] Each CDT supplementary component core code value shall have zero or more business terms.

8.6.7.7.2.4.2 Core Data Type Supplementary Component Core Value Domain Core Code Value Localized Metadata

The CDT supplementary component core value domain core scheme or list localized information class contains the relevant information necessary to associate native language identification of schemes or lists.

[D209] A CDT supplementary component core value domain core scheme or list shall have zero or more localized information classes

[D210] Each occurrence of a CDT supplementary component core value domain core scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

CDT supplementary component core value domain core scheme or list names should follow, as much as possible, all CDT DEN rules.

[D211] Each CDT supplementary component core value domain core scheme or list name shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

The CDT supplementary component core value domain core scheme or list name in the localized information class must only be expressed in the language identified by the language code property of that class.

[D212] Each supplementary component core value domain core scheme or list name shall be in the language identified by the language and locale code.

The CDT supplementary component core value domain core scheme or list business term must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D213] Each CDT supplementary component core value domain core scheme or list business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.6.7.7.3 Core Data Type Supplementary Component Core Value Domain Usage Rule

CDT supplementary component core value domains may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the CDT supplementary component value domain. The CDT supplementary component core value domain usage rules represent the specific application of a CDT supplementary component value domain in its role of expressing the value domain of its CDT supplementary component. CDT supplementary component core value domain usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D214] A CDT supplementary component core value domain shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D215] CDT supplementary component core value domain usage rules shall not replicate CDT, CDT content component, CDT content component core value domain, or CDT supplementary component usage rules.

[D216] A CDT supplementary component core value domain usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The CDT supplementary component core value domain usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D217] An unstructured CDT supplementary component core value domain usage rule constraint shall have a free form text expression that fully details the usage rule.

A CDT supplementary component core value domain formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D218] A structured CDT supplementary component core value domain usage rule constraint shall have a formal constraint language expression.

CDT supplementary component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D219] Every CDT supplementary component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

CDT supplementary component usage rules will also have condition types that identify when the formal constraint should be applied.

[D220] Every CDT supplementary component core value domain usage rule shall have a condition type.

[D221] Every CDT supplementary component core value domain usage rule condition type shall be one of `pre-condition`, `post-condition`, or `invariant`.

8.6.7.7.3.1 Core Data Type Supplementary Component Core Value Domain Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule in a library for all other usage rules, a CDT supplementary component core value domain usage rule may also have a unique name that semantically differentiates it from all other named usage rules.

[D222] A CDT supplementary component core value domain usage rule shall have zero or one names that is unique within the group of usage rules of library of which it is a part.

A CDT supplementary component core value domain usage rule may have several business terms. CDT supplementary component core value domain usage rule business terms are synonym terms under which the CDT supplementary component core value domain usage rule is commonly known and used in business.

[D223] Each CDT usage rule shall have zero or more business terms.

8.6.7.7.3.2 Core Data Type Supplementary Component Core Value Domain Usage Rule Identification Metadata Localized Information

The CDT supplementary component core value domain usage rule localized information class contains the relevant information necessary to associate native language expressions of usage rules.

[D224] A CDT supplementary component core value domain usage rule shall have zero or more localized information classes

[D225] Each occurrence of a CDT supplementary component core value domain usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

Core Data Type supplementary component core value domain localized information DENs should follow, as much as possible, all ASCC DEN rules.

[D226] Each CDT supplementary component core value domain usage rule localized information name shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

The CDT supplementary component core value domain usage rule name in the localized information class must only be expressed in the language identified by the language code property of that class.

[D227] Each usage rule name shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D228] Each CDT supplementary component core value domain usage rule localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.7 Business Data Types

A BDT defines the value domain for a particular BBIE property. Figure 8-3 describes the BDT and relationships between the BDT and its subordinate parts.

[Definition] – Business Data Type (BDT)

A business data type is a data type consisting of one and only one business data type content component that carries the actual content plus zero or more business data type supplementary components giving essential extra definition to the CDT content component. Business data types have business semantics.

BDTs can be qualified or unqualified. Unqualified BDTs are of type CDT without restrictions. Qualified BDTs are defined by specifying restrictions on the CDT from which it is derived. BDTs can only contain the CDT content and supplementary components contained in its underlying CDT. Qualified BDTs can only contain the supplementary components contained in its less qualified BDT.

8.7.1 Business Data Type – Data Type Term

The BDT is expressed by a data type term. The BDT data type term is a semantically meaningful name that serves as the basis for the DEN of the BDT and all qualified BDTs derived from it. The BDT data type term defines the form of the set of valid values for a BBIE data element or value domain. BDT data type terms semantically identify their source CDT by replicating the CDT data type term.

[D229] A BDT DEN data type term shall be the same as its source CDT data type term.

8.7.2 Business Data Type Qualifier Term

A BDT qualifier term is a word or words which help define and differentiate a qualified BDT from its higher level BDT. Qualifier terms are used to refine the

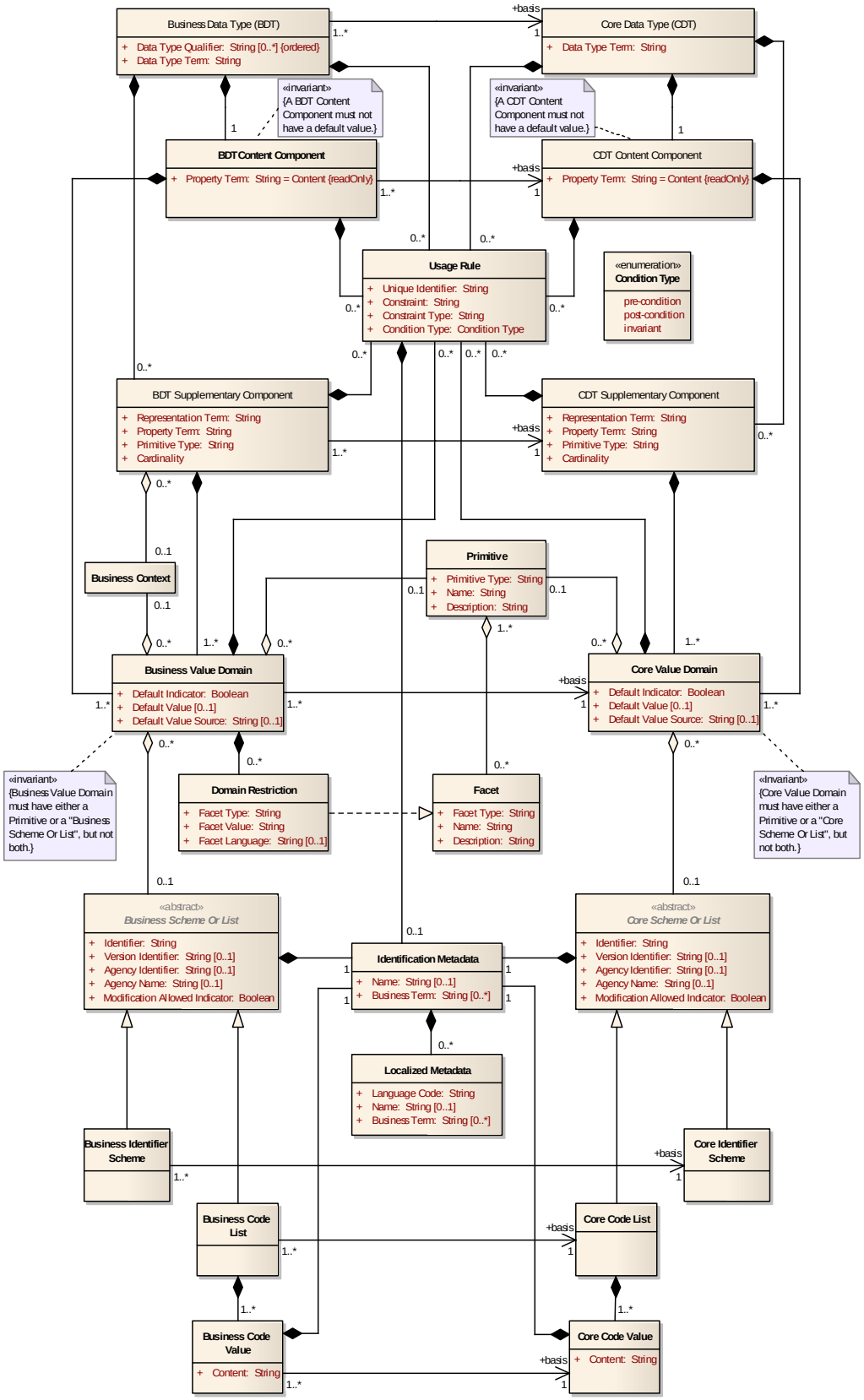


Figure 8-3. UML Diagram of Business Data Type Metamodel

semantic meaning of the DEN to reflect the nature of the restriction to the properties and representation of the data type as necessary to distinguish one BDT concept, conceptual domain, content model or data value domain from another.

[Note] BDT Qualifier Hierarchy

A BDT qualifier hierarchy is a tree like structure that reflects the order of the qualifiers for a set of qualified BDTs derived from the same unqualified BDT in a graph like form. The first level in a BDT hierarchy is the unqualified BDT construct, and each succeeding lower level is a more qualified BDT construct than its preceding BDT construct.

[D230] Where necessary, a BDT shall be qualified by restricting the set of valid values allowed by imposing restrictions on the BDT content component and/or the BDT supplementary component(s).

[D231] A qualified BDT shall be a restriction of its higher level BDT in a BDT hierarchy.

[D232] A qualified BDT shall be unique amongst the set of qualified BDTs in the library of which it is a part.

[D233] BDT qualifier terms shall precede the data type term.

[D234] Each BDT data type qualifier term shall be followed by an underscore and a space character (_).

[D235] Each word in a multi-worded BDT data type qualifier term shall be separated by a space character ( ).

BDT data type qualifier terms are derived from the semantic use of the restricted data type and not the restriction values themselves.

[D236] BDT qualifier terms shall be taken from the semantics of the supported BBIE(s).

[D237] BDT qualifier terms shall not describe the actual content or supplementary component restriction values.

[Example] – Allowed BDT Qualifiers

Allowed:

`Price_ Amount. Type`

Not Allowed:

`1 to 50 Euros_ Amount. Type` or **`One To Fifty Euros_ Amount. Type`**

[D238] A multi-worded BDT qualifier term shall have a unique semantic meaning compared to the words separately.

[D239] A qualifying BDT hierarchy shall be established when multiple qualifiers are used.

[Example] – BDT Qualifier Hierarchy

BBIE - **Trade_ Contract. Issue. Date Time**

May have any of the following data types:

Date Time. Type

Issue_ Date Time. Type

Contract_ Issue_ Date Time. Type

Trade_ Contract_ Issue_ Date Time. Type

8.7.3 Business Data Type Usage Rule

BDTs may have usage rules. Each BDT usage rule defines a constraint that describes specific conditions that are applicable to the BDT. BDT usage rules represent the specific application of a BDT in its role of expressing the value domain of BBIEs and BBIE Properties. BDT usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D240] A BDT shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D241] BDT usage rules shall not replicate CDT, BDT Content, BDT supplementary component, or BDT business value domain usage rules.

[D242] A BDT usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BDT usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free text.

[D243] An unstructured BDT usage rule constraint shall have a free form text expression that fully details the usage rule.

If a BDT usage rule is expressed in a formal language, it will have a structured constraint. A structured constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D244] A structured BDT usage rule constraint shall have a formal constraint language expression..

BDT usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D245] Every BDT usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BDT usage rules expressed will also have a condition type that identifies when the formal constraint should be applied.

[D246] Every BDT usage rule shall have a condition type.

[D247] Every BDT usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**..

8.7.3.1 Business Data Type Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given BDT from all other usage rules for a library, a BDT usage rule may also have an identification metadata class that provides additional information.

[D248] A BDT usage rule shall have zero or one identification metadata classes.

The usage rule identification metadata class contains a unique name that semantically differentiates it from all other named usage rules for the BDT.

[D249] A BDT usage rule shall have zero or one names that is unique within the group of usage rules of a BDT.

[Ed Note – we need to finalize how much uniqueness we want for the names – BDT or DT or library]

The BDT usage rule identification metadata class may contain business terms. BDT usage rule business terms are synonym terms under which the BDT usage rule is commonly known and used in business.

[D250] Each BDT usage rule shall have zero or more business terms.

8.7.3.2 Business Data Type Usage Rule Localized Metadata

BDT usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D251] A BDT usage rule shall have zero or more localized information classes

[D252] Each occurrence of a BDT usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[D253] BDT usage rule localized metadata shall be in the language identified by the language and locale code.

8.7.4 Business Data Type Identifiers

In order to ensure uniqueness, every BDT will have assigned a:

- **Unique Identifier (mandatory):** The identifier that references the BDT in a unique and unambiguous way.
- **Version Identifier (mandatory):** An indication of the evolution over time of the BDT.

[D254] Each BDT shall have a unique identifier within the library of which it is a part.

[D255] Each version of a BDT shall have a unique version identifier within the library of which it is a part.

8.7.5 Business Data Type Common Information

[D256] Each BDT shall have a common information class.

[D257] The BDT common information class shall consist of:

- **DEN (mandatory):** The official name of the BDT.
- **Definition (mandatory):** The semantic meaning of the BDT.
- **Business Term (optional, repetitive)**: A synonym term under which the BDT is commonly known and used in business.

[Example] – BDT Common Information

DEN – **`Start_ Date Time. Type`**

Definition – A **`start`** date, start time, start date time, or other start **`date time`** value used as a particular point in the progression of time.

Business Term – **`Begin`**

8.7.5.1 Business Data Type Dictionary Entry Names

The BDT DEN is based on the ISO 11179 data type term.

[D258] Each BDT DEN shall conform to all DT DEN rules.

[D139] The BDT DEN shall consist of the data type term and data type term qualifiers, if any, followed by a dot, a space character, and the term **`type`**.

[Example] – Business Data Type DEN

`Country_ Identifier. Type`

8.7.5.2 Business Data Type Definitions

[D259] BDT definitions shall conform to all rules for DT definitions.

[D260] The BDT definition shall include the BDT data type term and data type qualifier terms, if any.

8.7.5.3 Business Data Type Business Terms

A BDT may have several business terms. BDT business terms are those terms commonly used for day-to-day information exchanges within a given domain.

[D261] A BDT shall have zero or more business terms.

8.7.6 Business Data Type Localized Information

The BDT localized information class contains the relevant information necessary to associate native language expressions of BDT attributes to the BDT.

[D262] A BDT shall have zero or more localized information classes.

[D263] Each occurrence of a BDT localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of a BDT in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BDT in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BDT is commonly known and used in a business expression in that language. Business terms in the localized information class shall be in the language identified by the language and locale code.

BDT localized information DENs should follow, as much as possible, all BDT DEN rules.

[D264] Each BDT localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[D265] Each BDT localized information definition shall adhere to all BDT definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D266] Each BDT localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D267] Each BDT localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.7.7 Business Data Type Content Component

Each BDT will have a single BDT content component. BDT content components are defined in the UN/CEFACT Data Type Catalogue and are unique to the BDT to which they are assigned.

[D268] A BDT shall have one and only one BDT content component.

A BDT content component is the CDT content component of the source CDT.

[D269] A BDT content component shall inherit all of the properties and restrictions, including the name, of the CDT content component of the source CDT.

8.7.7.1 Business Data Type Content Component Property Term

Each BDT content component has a property term. The BDT content component property term represents the actual content of a data element. The BDT content component property term is the same as the CDT content component of the source CDT and has a fixed value of **`Content`**.

[D270] Each BDT content component shall have a property term.

[D271] The BDT content component property term shall have a fixed value of **`Content`**.

8.7.7.2 Business Data Type Content Component Usage Rules

BDT content components may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BDT content component. The BDT content component usage rules represent the specific application of a BDT content component in its role of expressing the value domain of its BDT. BDT usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D272] A BDT content component shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D273] BDT content component usage rules shall not replicate BDT, BDT supplementary component, or BDT business value domain usage rules.

[D274] BDT content component usage rules shall not replicate BDT business value domain restrictions.

[D275] A BDT content component usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BDT content component usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D276] An unstructured BDT content component usage rule shall have a free form text expression that fully details the usage rule.

A BDT content component formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D277] a structured BDT content component usage rule constraint shall have a formal constraint language expression.

BDT content component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D278] Every BDT content component usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BDT content component usage rules will also have a condition type that identifies when the constraint should be applied.

[D279] Every BDT content component usage rule shall have a condition type.

[D280] Every BDT content component usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.7.7.2.1 Business Data Type Content Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given BDT content component from all other usage rules for a library, a BDT content component usage rule may also have a unique name that semantically differentiates it from all other named usage rules for the BDT.

[D281] A BDT content component usage rule shall have zero or one names that is unique within the group of usage rules of a BDT content component.

The BDT content component usage rule metadata class may contain business terms. BDT content component usage rule business terms are synonym terms under which the BDT content component usage rule is commonly known and used in business.

[D282] Each BDT content component usage rule shall have zero or more business terms.

8.7.7.2.2 Business Data Type Content Component Usage Rule Identification Metadata Localized Information

BDT content component usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D283] A BDT content component usage rule shall have zero or more localized information classes.

[D284] Each occurrence of a BDT content component usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[D285] BDT content component usage rule localized metadata shall be in the language identified by the language and locale code.

8.7.7.3 Business Data Type Content Component Common Information

Each BDT content component has a common information class.

[D286] Each BDT content component shall have a common information class.

[D287] The BDT content component common information class shall consist of:

- **DEN (mandatory):** The official name of a BDT content component.
- **Definition (mandatory):** The semantic meaning of a BDT content component.
- **Business Term (optional, repetitive)**: A synonym term under which the BDT content component is commonly known and used in business.

[Example] – BDT Content Component Common Information

DEN – **`Amount. Content`**

Definition – An **`amount`** is a number of monetary units.

Business Term – **`Money`**

8.7.7.3.1 Business Data Type Content Component Dictionary Entry Names

The BDT content component DENs are based on ISO 11179 defined data type and property terms.

[D288] Each BDT content component DEN shall conform to all DT DEN rules.

[D289] The DEN of a BDT content component shall be the DEN of the CDT content component of the source CDT.

[Example] – Business Data Type Content Components

`Amount. Content`; **`Date Time. Content`**

8.7.7.3.2 Business Data Type Content Component Definition

[D290] Each BDT content component definition shall conform to all DT definition rules.

[D291] The BDT content component definition shall include the primitive type term and the definition of the source representation term.

8.7.7.3.3 Business Data Type Content Component Business Terms

A BDT content component may have several business terms. BDT content component business terms are synonym terms under which the BDT content component is commonly known and used in business.

[D292] A BDT content component shall have zero or more business terms.

8.7.7.4 Business Data Type Content Component Localized Information

The BDT content component localized information class contains the relevant information necessary to associate native language expressions of BDT content components to the BDT content component.

[D293] Each BDT content component shall have zero or more localized information classes.

[D294] Each occurrence of a BDT content component localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BDT content component in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BDT content component in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BDT content component is commonly known and used in a business expression in that language.

BDT content component localized information DENs should follow, as much as possible, all BDT content component DEN rules.

[D295] Each BDT content component localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

[D296] Each BDT content component localized information definition shall adhere to all BDT content component definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D297] Each BDT content component localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D298] Each BDT content component localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.7.7.5 Business Data Type Content Component Business Value Domain

BDT content components can have one or more value domains. A BDT content component business value domain defines the set of allowed values through the presence of either a scheme or list, or a primitive with defined facets and their restrictions.

[D299] A BDT content component shall have one or more value domains.

A BDT content component business value domain is based on its source CDT content component value domain.

[D300] A BDT content component value domain shall inherit all of the properties and restrictions, including the name of the source CDT content component value domain.

Since each BDT content component may have multiple value domains, each defined value domain contains a default indicator that identifies it as the default value domain amongst the set of value domains for the BDT content component.

[D301] A BDT content component business value domain shall have a default indicator whose value = **`true`** if it is the default value domain.

[D302] A BDT content component business value domain shall have a default indicator whose value = **`false`** if it is not the default value domain.

Each BDT content component business value domain may also have a default value. This default value represents a BDT content component business value domain value that is to be automatically applied to the BDT content component in the absence of a choice made by the user.

[D303] A BDT content component business value domain shall have zero or one default values.

Default values will be conformant to the defined primitive or scheme or list of the BDT content component business value domain.

[D304] the BDT content component business value domain default value shall be conformant to its defined primitive or scheme or list.

If a BDT content component value domain has a default value, the source of that default value will also be identified.

[D305] A BDT business value domain with a default value shall have a default value source.

[Note] – Default Value Sources

There are no specific rules for the structure of value domain default value sources. Implementers are free to choose any structure providing ti guarantees uniqueness within the group of value domain default value sources of a value domain.

BDT content component business value domains are defined by either a primitive or a scheme or list. Each primitive or scheme or list constitutes a separate business value domain for the BDT content component.

[D306] Each BDT content component business value domain shall consist of a primitive or a scheme or list.

8.7.7.5.1 Business Data Type Content Component Business Value Domain – Domain Restrictions

.Domain restrictions of a BDT content component business value domain restrict the set of allowed values of the source CDT content component or less qualified BDT content component. Domain restrictions only apply when the business value domain is defined by a primitive as domain restrictions take the form of facets of the primitive. Restrictions of the business value domain when the domain is defined by a business scheme or list take the form of restrictions to the scheme or list itself rather than domain restriction facets.

[D307] Domain restrictions of a BDT content component business value domain shall only be used to define restrictions on possible values.

[D308] A BDT content component business value domain shall have zero or more domain restrictions.

The allowed set of facets for a specific BDT content component business value domain is determined by its primitive type. The allowed restrictions for each primitive type are defined in the UN/CEFACT Data Type Catalogue.

[D309] BDT content component business value domain domain restrictions shall be limited to those allowed for the primitive of the BDT content component business value domain as specified in the UN/CEFACT Data Type Catalogue.

[Example] – Allowed Facet Restrictions for Primitive Type of Date

BDT Content Component – Date. Content

Primitive Type – Date

Allowed Restriction Facets for Date:

Facet Type: Minimum Inclusive – **`2005-06-25`**

Facet Type: Maximum Inclusive – **`2005-06-30`**

`or`

Facet Type: Minimum Exclusive – **`2007-01-01`**

Facet Type: Maximum Exclusive – **`2007-03-31`**

[D310] Each BDT content component business value domain domain restriction shall have zero or one component restrictions for each facet type.

[Example] – Multiple facet restrictions

The BDT of **`Code. Type`** has a content component of **`Code. Content`** whose business value domain primitive is **`string.`** The allowed facet types for the string primitive type include **`Expression`**, **`Length`**, **`Minimum Length`**, **`Maximum Length`**, and **`Enumeration`**. For a qualified data type of **`Currency_ Code. Type`**, each of the allowed facet restrictions may or may not be present. If an allowed facet restriction is present, there can only be one occurrence of that facet type.

Primitive type facet restrictions for BDT content component business value domains consist of the facet type, facet value, and optional facet language.

[D311] Each BDT content component business value domain – domain restriction shall contain the following attributes:

- **Facet Type (mandatory)**: Identifies the facet being defined.
- **Facet Value (mandatory)**: The actual facet restriction value.
- **Facet Language (mandatory for expression/not used otherwise)**: For a facet type of expression, defines the language of the regular expression of the facet value such as Perl, W3C XML Schema Definition Language, JAVA, or Microsoft .Net.

[Example] – Component Restriction

For a BDT content component business value domain whose primitive type is **`binary`**, an allowed facet would be **`length`**. The values for the **`length`** facet would be:

Facet Type (mandatory) - **`length`**

Facet Value (mandatory): **`10`**

Facet Language (optional): not used since the facet type is not **`expression.`**

8.7.7.5.2 Business Data Type Content Component Business Value Domain Primitive

Primitives represent basic building blocks for defining value domains of content and supplementary components. Each content component business value domain can have zero or one primitives defined for it. The BDT content component business value domain primitive defines the value domain. Primitives are referred to as primitive types.

Primitives include, but are not limited to:

- Binary
- Date
- Decimal
- Double
- Float
- Integer
- String
- Token

[D312] Each BDT content component business value domain shall have zero or one primitives.

[D313] A BDT content component primitive shall be one of the defined primitives in the *UN/CEFACT Data Type Catalogue*.

Each primitive shall have a primitive type. The primitive type code value is taken from a primitive type code list.

[D314] Every BDT content component business value domain primitive shall have a primitive type taken from a primitive type code list.

[Note] –Primitive Type Code List

UN/CEFACT will publish and make freely available a primitive type code list for use in support of this rule.

Each primitive has a formal name. This name typically represents the nature of the value domain it represents.

[D315] Every BDT content component business value domain primitive shall have a primitive name.

[D316] Every BDT content component business value domain primitive name shall be unique within the set of primitives of BDTs.

A BDT content component business value domain primitive will also have a description that semantically defines its value domain.

[D317] Each BDT content component business value domain primitive shall have a description that semantically defines its value domain.

8.7.7.5.2.1 Business Data Type Content Component Business Value Domain Primitive Facet

The value domains expressed by primitives are quantified through their facets. A primitive may have one or more facets. Each facet defines or constrains an aspect of the value domain expressed by the primitive.

[D318] Each BDT content component business value domain primitive shall have zero or more facets.

Each facet shall have a facet type. The facet type code value is taken from a facet type code list.

[D319] Each BDT content component business value domain primitive facet shall have a facet type taken from a facet type code list.

[Note] –Facet Type Code List

UN/CEFACT will publish and make freely available a Facet Type Code List for use in support of this rule.

[D320] Each BDT content component business value domain primitive facet shall have a name that is unique amongst the set of facet names of a primitive.

[D321] Each BDT content component business value domain primitive facet shall have a description that semantically expresses the nature of the restrictions associated with it.

8.7.7.5.3 Business Data Type Content Component Business Value Domain Scheme or List

Schemes are the equivalent of a pattern facet. A scheme formally expresses the pattern and the allowed values for populating that pattern in the form of identifiers. Lists are the equivalent of enumerated lists and are typically published as formal code lists. The set of codes in a formal code list is used by business value domains as an enumerated set of allowed values.

Unambiguous identification of the scheme or list is necessary.

[D322] Every BDT content component business value domain scheme or list shall have an identifier.

A version identifier serves to differentiate one version of a scheme or list from all other versions of the scheme or list.

[D323] Every BDT content component business value domain scheme or list shall have zero or one version identifiers.

Every scheme or list will be owned by an organization. The organization may either identified by a unique identifier or a name.

[D324] Every BDT content component business value domain scheme or list shall have either an agency identifier or an agency name.

[D325] Every BDT content component business value domain scheme or list shall have zero or one agency identifiers.

[Note] – Agency Identifier

UN/CEFACT recommends using UN/CEFACT Agency Identifier Code List (Data Element 3055) in the latest version of the UN/CEFACT directory.

[D326] Every BDT content component business value domain scheme or list shall have zero or one agency names.

Business Data Types are able to place restrictions on the content of schemes and lists. If such restrictions are undesireable, then this will be indicated through the use of a modification allowed indicator.

[D327] Every BDT content component business value domain scheme or list shall have a modification allowed indicator whose value = **`true`** if modifications are allowed, or whose value = **`false`** if modifications are not allowed.

8.7.7.5.3.1 Business Data Type Content Component Business Scheme or List Identification Metadata

Although the identifier is sufficient to differentiate one business scheme or list for a given BDT content component business value domain from all other schemes or lists for that BDT content component, a BDT content component business scheme or list may also have an identification metadata class that provides additional information.

[D328] A BDT content component business scheme or list shall have zero or one identification metadata classes.

The BDT content component business scheme or list identification metadata class contains a unique name that semantically differentiates it.

[D329] A BDT content component business scheme or list shall have zero or one names that is unique within the group of business scheme or lists of a BDT content component.

The BDT content component business scheme or list identification metadata class may contain business terms. These business terms are synonym terms under which the scheme or list is commonly known and used in business.

[D330] Each BDT content component business value domain business scheme or list shall have zero or more business terms.

8.7.7.5.3.2 Business Data Type Content Component Business Value Domain Business Scheme or List Identification Metadata Localized Information

BDT content component business value domain business schemes or lists may have localized information classes.

[D331] A BDT content component business value domain business scheme or list shall have zero or more localized information classes

[D332] Each occurrence of a BDT content component business value domain business scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D333] BDT content component business value domain business scheme or list localized metadata shall be in the language identified by the language and locale code.

8.7.7.5.3.3 Business Data Type Content Component Business Value Domain Business Identifier Scheme

Business Data Type content component business value domain business identifier schemes are typically not enumerated, rather the scheme defines a regular expression or pattern that is used to populate its set of values and also used to validate values. BDT content component business value domain business identifier schemes can be restricted from their source CDT content component core value domain core scheme, or less restricted BDT content component business value domain business scheme. These restrictions take the form of subsetting the set of allowed values.

[D334] Each BDT content component business value domain business identifier scheme set of allowed values shall be equal to or less than the set of allowed values of its source CDT content component core value domain core identifier scheme or less restricted BDT content component core value domain business identifier scheme.

8.7.7.5.3.4 Business Data Type Content Component Business Value Domain Business Code List

Business Data Type content component business value domain business code lists contain lists of enumerated code values.

[D335] Each BDT content component business value domain business code list shall contain one or more business code values.

BDT content component business value domain business code lists can be equal to or a restriction of, but never an extension of, their source CDT content component core value domain core code list, or less restricted BDT content component business value domain business code list.

[D336] Each BDT content component business value domain business code list set of allowed values shall be equal to or less than the set of allowed values of its source CDT content component core value domain core code list or less restricted BDT content component core value domain business code list.

8.7.7.5.3.4.1 Business Data Type Content Component Business Value Domain Business Code Value Identification Metadata

A BDT content component business code value will contain an identification metadata class consisting of a name and optional business term or terms.

[D337] A BDT content component business value domain business code list business code value shall have zero or one identification metadata classes.

[D338] A BDT content component business value domain business code list business code value shall have zero or one names that is unique within the set of business code values for a business code list.

The BDT content component business code list identification metadata class may also contain several business terms. These business terms are synonym terms under which the business code value is commonly known and used in business.

[D339] Each BDT content component business code value shall have zero or more business terms.

8.7.7.5.3.4.2 Business Data Type Content Component Business Value Domain Business Code Value Localized Metadata

BDT content component business value domain business scheme or list may have localized metadata that is used to provide other language expressions of its name and business terms.

[D340] A BDT content component business value domain business scheme or list shall have zero or more localized metadata classes.

[D341] Each occurrence of a BDT content component business value domain business scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D342] BDT content component business value domain business scheme or list localized metadata shall be in the language identified by the language and locale code.

8.7.7.5.4 Business Data Type Content Component Business Value Domain Usage Rule

BDT content component business value domains may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BDT content component value domain. BDT content component business value domain usage rules represent the specific application of a BDT content component value domain in its role of expressing the value domain of its BDT content component. BDT content component business value domain usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D343] A BDT content component business value domain shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D344] BDT content component business value domain usage rules shall not replicate BDT, BDT content component, BDT supplementary component or BDT supplementary component business value domain usage rules.

[D345] A BDT content component business value domain usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BDT content component business value domain usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D346] An unstructured BDT content component business value domain usage rule constraint shall have a free form text expression that fully details the usage rule.

A BDT content component business value domain formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D347] A structured BDT content component business value domain usage rule constraint shall have a formal constraint language expression.

BDT content component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D348] Every BDT content component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BDT content component usage rules will also have condition types that identify when the formal constraint should be applied.

[D349] Every BDT content component business value domain usage rule shall have a condition type.

[D350] Every BDT content component business value domain usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.7.7.5.4.1 Business Data Type Content Component Business Value Domain Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule in a library for all other usage rules, a BDT content component business value domain usage

rule may also have an identification metadata class that provides additional information.

[D351] A BDT content component business value domain usage rule shall have zero or one identification metadata classes.

The BDT content component business value domain usage rule contains a unique name that semantically differentiates the usage rule from all other named usage rules for the BDT.

[D352] A BDT content component business value domain usage rule shall have zero or one names that is unique within the group of usage rules of a BDT.

The BDT content component business value domain usage rule identification metadata class may contain business terms. BDT content component business value domain usage rule business terms are synonym terms under which the BDT content component business value domain usage rule is commonly known and used in business.

[D353] Each BDT content component business value domain usage rule shall have zero or more business terms.

8.7.7.5.4.2 Business Data Type Content Component Business Value Domain Usage Rule Identification Metadata Localized Information

BDT content component business value domain usage rules may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D354] A BDT content component business value domain usage rule shall have zero or more localized information classes.

[D355] Each occurrence of a BDT content component business value domain usage rule localized metadata class shall contain:

- **Language Code (mandatory):** A code which identifies the language being used. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[D356] BDT content component business value domain usage rule localized metadata shall be in the language identified by the language and locale code.

8.7.8 Business Data Type Supplementary Components

Business Data Type supplementary components are defined and published in the UN/CEFACT Data Type Catalogue, and are unique to the BDT to which they are assigned. A BDT will have zero or more CDT supplementary components.

[D357] A BDT shall have zero or more BDT supplementary components.

BDT supplementary components are based on the CDT supplementary component of the source CDT.

[D358] A BDT supplementary component shall inherit all of the properties and restrictions, including the name of the CDT supplementary component of the source CDT.

8.7.8.1 Business Data Type Supplementary Component Property Term

Each BDT supplementary component contains a property term. The BDT supplementary component property term is a semantically meaningful name for a unique characteristic that can be used in a BDT. The BDT supplementary component property term is the same as the CDT supplementary component of the source CDT.

[D359] Each BDT supplementary component shall have a property term.

[D360] Each BDT supplementary component property term shall be the same as the source CDT supplementary component of the source CDT.

8.7.8.2 Business Data Type Supplementary Component Representation Term

Each BDT supplementary component contains a representation term. The representation term is a semantically meaningful name that represents the value domain of the supplementary component. UN/CEFACT defines the approved representation terms as part of the UN/CEFACT Data Type Catalogue.

[D361] A representation term shall be defined for each BDT supplementary component.

[D362] Each BDT supplementary component representation term shall be the same as the CDT supplementary component of the source CDT.

8.7.8.3 Business Data Type Supplementary Component Cardinality

The restriction on the presence of BDT supplementary components will be accomplished through the use of the BDT supplementary component cardinality value. Each BDT supplementary component will have its cardinality explicitly expressed. The BDT supplementary component cardinality defines the occurrence requirements of the supplementary component within its data type,

[D363] Each BDT supplementary component shall have a cardinality that consists of a set of values consisting of a minimum occurrence and a maximum occurrence.

[D364] BDT supplementary component cardinality shall be equal to `[0..1]` if the BDT supplementary component is optional, or `[1..1]` if mandatory.

An unqualified BDT will always include the same supplementary components as its source CDT.

[D365] An unqualified BDT shall include the same supplementary components as its source CDT.

An unqualified BDT will never change the cardinality of the included supplementary components of its source CDT.

[D366] The cardinality of a supplementary component of an unqualified BDT shall be the same as its source CDT.

A qualified BDT will always include all mandatory supplementary components of its source BDT and retain its cardinality as being mandatory. A qualified BDT may also restrict to being mandatory, or not include, any supplementary components that are optional in its source BDT.

Once a BDT supplementary component is not included in a qualified BDT, it can never be added back to a more qualified BDT.

[D367] A BDT supplementary component occurrence shall only be restricted and never extended.

8.7.8.4 Business Data Type Supplementary Component Usage Rules

A BDT supplementary component may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BDT supplementary component. The BDT supplementary component usage rules represent the specific application of a BDT supplementary component in its role of expressing the value domain of its BDT. BDT supplementary component usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D368] A BDT supplementary component shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D369] BDT supplementary component usage rules shall not replicate BDT, BDT content component, or BDT core value domain usage rules.

[D370] A BDT supplementary component usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a BDT supplementary component.

The BDT supplementary component usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D371] An unstructured BDT supplementary component usage rule shall have a free form text expression that fully details the usage rule.

A structured constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D372] A structured BDT supplementary component usage rule shall have a formal constraint language expression.

BDT supplementary component usage rule formal constraints must also be specified. The constraint type value is taken from a constraint type code list.

[D373] Every BDT supplementary component usage rule shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BDT supplementary component usage rules will also have a condition type that identifiesy when the constraint should be applied.

[D374] Every BDT supplementary component usage rule shall have a condition type.

[D375] Every BDT supplementary component usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.7.8.4.1 Business Data Type Supplementary Component Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule for a given BDT supplementary component from all other usage rules in a library, a BDT supplementary component usage rule may also have a unique name that semantically differentiates it from all other named usage rules for the BDT supplementary component.

[D376] A BDT supplementary component usage rule shall have zero or one names that is unique within the group of usage rules of a BDT supplementary component.

An BDT supplementary component usage rule may have several business terms. BDT supplementary component usage rule business terms are synonym terms under which the BDT supplementary component usage rule is commonly known and used in business.

8.7.8.4.2 Business Data Type Supplementary Component Usage Rule Identification Metadata Localized Information

The BDT supplementary component usage rule localized information class contains the relevant information necessary to associate native language expressions of usage rules.

[D377] A BDT supplementary component usage rule shall have zero or more localized information classes.

[D378] Each occurrence of a usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

Business Data Type supplementary component localized information names should follow, as much as possible, all BDT supplementary component DEN rules.

[D379] Each usage rule localized information name shall only consist of alphabetic characters, ideographic characters, plus the dot, the underscore and the space characters unless required by language rules.

The usage rule name in the localized information class must only be expressed in the language identified by the language code property of that class.

[D380] Each usage rule name shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D381] Each usage rule localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.7.8.5 Business Data Type Supplementary Component Common Information

Each BDT supplementary component has a common information class.

[D382] Each BDT supplementary component shall have a common information class.

[D383] The BDT supplementary component common information class shall consist of:

- **DEN (mandatory):** The official name of the BDT supplementary component.
- **Definition (mandatory):** The semantic meaning of the BDT supplementary component.
- **Business Term (optional, repetitive)**: A synonym term under which the BDT supplementary component is commonly known and used in business.

Example] – BDT Supplementary Component Common Information

DEN – **`Amount. Currency Code List Agency. Identifier`**

Definition – The **`identifier`** of the **`agency`** that maintains the **`currency code list`** used for the **`amount`**.

Business Term – **`Currency Code Owner`**

8.7.8.5.1 Business Data Type Supplementary Component Dictionary Entry Names

The BDT supplementary component DENs are based on ISO 11179 concepts of data type term, property term, and representation term.

[D384] Each BDT supplementary component DEN shall conform to all DT DEN rules.

[D385] The DEN of a BDT supplementary component shall be the DEN of the CDT supplementary component of the source CDT.

[Example] – Business Data Type Supplementary Components

Amount. Currency Code List Version. Identifier, **Code. List Agency. Identifier**, **Quantity. Unit. Code**

[D386] The BDT supplementary component DEN shall be unique amongst all BDT supplementary component names within the library of which it is a part.

8.7.8.5.2 Business Data Type Supplementary Component Definitions

A BDT supplementary component definition provides a clear, unambiguous and complete explanation of the meaning of a BDT supplementary component and its relevance for the related BDT.

[D387] Each BDT supplementary component definition shall conform to all DT definition rules.

[D388] The definition of a BDT supplementary component shall include the data type term of the BDT to which it belongs, the property term and the representation term.

8.7.8.5.3 Business Data Type Supplementary Component Business Terms

BDT supplementary components may have business terms. BDT supplementary component business terms are synonyms commonly used for day-to-day information exchanges within a given domain.

[D389] Each BDT supplementary component shall have zero or more business terms.

8.7.8.6 Business Data Type Supplementary Component Localized Information

The BDT supplementary component localized information class contains the relevant information necessary to associate native language expressions of BDT supplementary components to the BDT supplementary component.

[D390] Each BDT supplementary component shall have zero or more localized information classes.

[D391] Each occurrence of a BDT supplementary component localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **DEN (optional):** The official name of the BDT supplementary component in a language other than English.
- **Definition (mandatory):** The semantic meaning of the BDT supplementary component in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the BDT supplementary component is commonly known and used in a business expression in that language. Business terms in the localized information class shall be in the language identified by the language and locale code.

BDT supplementary component localized information DENs should follow, as much as possible, all BDT supplementary component DEN rules.

[D392] Each BDT supplementary component localized information DEN shall only consist of alphabetic characters, ideographic characters, plus the dot, the undersbusiness and the space characters unless required by language rules.

[D393] Each BDT supplementary component localized information definition shall adhere to all BDT supplementary component definition rules other than the requirement to be in the English language.

The DEN and definition in the localized information class must only be expressed in the language identified by the language code property of that class.

[D394] Each BDT supplementary component localized information DEN and definition shall be in the language identified by the language and locale code.

The business terms must only be expressed in the language identified by the language code property of that class, or a recognized dialect of the language.

[D395] Each BDT supplementary component localized information business term shall be in the language identified by the language and locale code, or a recognized dialect of the language.

8.7.8.7 Business Data Type Supplementary Component Business Value Domain

BDT supplementary components can have one or more value domains. A BDT supplementary component business value domain defines the set of allowed values through the presence of either a scheme or list, or a primitive with defined facets and their restrictions.

[D396] A BDT supplementary component shall have one or more value domains.

Since each BDT supplementary component may have multiple value domains, each defined value domain contains a default indicator that identifies it as the default value domain amongst the set of value domains for the BDT supplementary component.

[D397] A BDT supplementary component business value domain shall have a default indicator whose value = **`true`** if it is the default value domain.

[D398] A BDT supplementary component business value domain shall have a default indicator whose value = **`false`** if it is not the default value domain.

Each BDT supplementary component business value domain may also have a default value. This default value represents a BDT supplementary component value that is to be automatically applied to the BDT supplementary component in the absence of a choice made by the user. The default value should be used unless an alternate is required to meet specific business requirements.

[D399] A BDT supplementary component business value domain shall have zero or one default values.

Example – BDT Supplementary Component Default Value

Supplementary Component – **`Amount. Currency Code List. Identifier`**

Default Value – **`ISO 4217`**

Default values will be conformant to the defined primitive or scheme or list of the BDT supplementary component business value domain.

[D400] the BDT supplementary component business value domain default value shall be conformant to its defined primitive or scheme or list.

If a BDT supplementary component value domain has a default value, the source of that default value will also be identified.

[D401] A BDT value domain with a default value shall have a default value source.

[Note] – Value Domain Default Value

There are no specific rules for the structure of value domain default value sources. Implementers are free to choose any structure providing it guarantees uniqueness within the group of value domain default value sources of a value domain.

BDT supplementary component business value domains are defined by either a primitive or a scheme or list. Each primitive or scheme or list constitutes a separate business value domain for the BDT supplementary component.

[D402] Each BDT supplementary component business value domain shall consist of a primitive or a scheme or list.

8.7.8.7.1 Business Data Type Supplementary Component Business Value Domain – Domain Restrictions

.Domain restrictions of a BDT supplementary component business value domain restrict the set of allowed values of the source CDT supplementary component or less qualified BDT supplementary component. Domain restrictions only apply when the business value domain is defined by a primitive as domain restrictions take the form of facets of the primitive. Restrictions of the business value domain when the domain is defined by a business scheme or list take the form of restrictions to the scheme or list itself rather than domain restriction facets.

[D403] Domain restrictions of a BDT supplementary component business value domain shall only be used to define restrictions on possible values.

[D404] A BDT supplementary component business value domain shall have zero or more domain restrictions.

The allowed set of facets for a specific BDT supplementary component business value domain is determined by its primitive type. The allowed restrictions for each primitive type are defined in the UN/CEFACT Data Type Catalogue.

[D405] BDT supplementry component business value domain domain restrictions shall be limited to those allowed for the primitive of the BDT supplementary component business value domain as specified in the UN/CEFACT Data Type Catalogue.

[Example] – Allowed Facet Restrictions for Primitive Type of String

BDT Supplementary Component – **`Measure. Unit. Code`**

Primitive Type – String

Allowed Restriction Facets for String:

Facet Type: Expression – **`[A-Z]{1,2}`**

Facet Type: Length – **not used**

Facet Type: Minimum Length – **`1`**

Facet Type: Maximum Length – **`2`**

Facet Type: Enumeration – **`FT, YD, MI, CM, M, CM`**

[D406] Each BDT supplementary component business value domain domain restriction shall have zero or one domain restrictions for each facet type.

[Example] – Multiple facet restrictions

The BDT of **`Code. Type`** has a supplementary component of **`Code List. Name. Text`** whose primitive type is **`string`**. The allowed facet types for the string primitive type include **`Expression`**, **`Length`**, **`Minimum Length`**, **`Maximum Length`**, and **`Enumeration`**. For a qualified data type of **`Business Type_ Code. Type`**, each of the allowed facet restrictions may or may not be present for the **`Code List. Name. Text`**. If present, there can only be one instance of each facet type.

Primitive type facet restrictions for BDT supplementary component business value domains consist of the facet type, facet value, and optional facet language.

[D407] Each BDT supplementary component business value domain – domain restriction shall contain the following attributes:

- **Facet Type (mandatory)**: Identifies the facet being defined.
- **Facet Value (mandatory)**: The actual facet restriction value.
- **Facet Language (mandatory for expression/not used otherwise)**: For a facet type of expression, defines the language of the regular expression of the facet value such as Perl, W3C XML Schema Definition Language, JAVA, or Microsoft .Net.

[Example] – Domain Restriction

For a BDT supplementary component whose primitive type is **`string`**, an allowed facet would be **`expression`**. The values for the **`Expression`** facet would be:

Facet Type (mandatory): **`Expression`**

Facet Value (mandatory): **`[A-Z]*`**

Facet Language (optional): **`Perl`**

8.7.8.7.2 Business Data Type Supplementary Component Business Value Domain Primitive

Primitives represent basic building blocks for defining value domains of content and supplementary components. Each BDT supplementary component business value

domain can have zero or one primitives defined for it. The BDT supplementary component business value domain primitive defines the value domain. Primitives are referred to as primitive types.

Primitives include, but are not limited to:

- Binary
- Date
- Decimal
- Double
- Float
- Integer
- String
- Token

[D408] Each BDT supplementary component business value domain shall have zero or one primitives.

[D409] A BDT supplementary component primitive shall be one of the defined primitives in the *UN/CEFACT Data Type Catalogue*.

Each primitive shall have a primitive type. The primitive type code value is taken from a primitive type code list.

[D410] Every BDT supplementary component business value domain primitive shall have a primitive type taken from a primitive type code list.

[Note] –Primitive Type Code List

UN/CEFACT will publish and make freely available a primitive type code list for use in support of this rule.

Each primitive has a formal name. This name typically represents the nature of the value domain it represents.

[D411] Every BDT supplementary component business value domain primitive shall have a primitive name.

[D412] Every BDT supplementary component business value domain primitive name shall be unique within the set of primitives of BDTs.

A BDT supplementary component business value domain primitive will also have a description that semantically defines its value domain.

[D413] Each BDT supplementary component business value domain primitive shall have a description that semantically defines its value domain.

8.7.8.7.2.1 Business Data Type Supplementary Component Business Value Domain Primitive Facet

The value domains expressed by primitives are quantified through their facets. A primitive may have one or more facets. Each facet defines or constrains an aspect of the value domain expressed by the primitive.

[D414] Each BDT supplementary component business value domain primitive shall have zero or more facets.

Each facet shall have a facet type. The facet type code value is taken from a facet type code list.

[D415] Each BDT supplementary component business value domain primitive facet shall have a facet type taken from a facet type code list.

[Note] –Facet Type Code List

UN/CEFACT will publish and make freely available a facet type code list for use in support of this rule.

[D416] Each BDT supplementary component business value domain primitive facet shall have a name that is unique amongst the set of facet names of a primitive.

[D417] Each BDT supplementary component business value domain primitive facet shall have a description that semantically expresses the nature of the restrictions associated with it.

8.7.8.7.3 Business Data Type Supplementary Component Business Value Domain Business Scheme or List

Schemes are the equivalent of a pattern facet. A scheme formally expresses the pattern and the allowed values for populating that pattern in the form of identifiers. Lists are the equivalent of enumerated lists and are typically published as formal code lists. The set of codes in a formal code list is used by business value domains as an enumerated set of allowed values.

Unambiguous identification of the scheme or list is necessary.

[D418] Every BDT supplementary component business value domain scheme or list shall have an identifier.

A version identifier serves to differentiate one version of a scheme or list from all other versions of the scheme or list.

[Dxx] Every BDT supplementary component business value domain scheme or list shall have zero or one version identifiers.

Every scheme or list will be owned by an organization. The organization may either identified by a unique identifier or a name.

[D419] Every BDT supplementary component business value domain scheme or list shall have either an agency identifier or an agency name.

[D420] Every BDT supplementary component business value domain scheme or list shall have zero or one agency identifiers.

[Note] – Agency Identifer

UN/CEFACT recommends using UN/CEFACT Code List Responsible Agency Code (Data Element 3055) in the latest version of the UN/CEFACT directory.

[D421] Every BDT supplementary component business value domain scheme or list shall have zero or one agency names.

Business Data Types are able to place restrictions on the content of schemes and lists. If such restrictions are undesireable, then this will be indicated through the use of a modification allowed indicator.

[D422] Every BDT supplementary component business value domain scheme or list shall have a modification allowed indicator whose value = **`true`** if modifications are allowed, or whose value = **`false`** if modifications are not allowed.

8.7.8.7.3.1 Business Data Type Supplementary Component Business Value Domain Business Scheme or List Identification Metadata

Although the identifier is sufficient to differentiate one business scheme or list for a given BDT supplementary component business value domain from all other schemes or lists for that BDT supplementary component, a BDT supplementary component business scheme or list may also have an identification metadata class that provides additional information.

[D423] A BDT supplementary component business scheme or list shall have zero or one identification metadata classes.

The BDT supplementary component business scheme or list identification metadata class contains a unique name that semantically differentiates it.

[D424] A BDT supplementary component business scheme or list shall have zero or one names that is unique within the group of business scheme or lists of a BDT supplementary component.

A BDT supplementary component business scheme or list may contain business terms. These business terms are synonym terms under which the scheme or list is commonly known and used in business.

[D425] Each BDT supplementary component business value domain business scheme or list shall have zero or more business terms.

8.7.8.7.3.2 Business Data Type Supplementary Component Business Value Domain Business Scheme or List Identification Metadata Localized Information

The BDT supplementary component business value domain business scheme or list localized information class contains the relevant information necessary to associate native language identification of schemes or lists.

[D426] A BDT supplementary component business value domain business scheme or list shall have zero or more localized information classes

[D427] Each occurrence of a BDT supplementary component business value domain business scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D428] BDT content component business value domain business scheme or list localized metadata shall be in the language identified by the language and locale code.

8.7.8.7.3.3 Business Data Type Supplementary Component Business Value Domain Business Identifier Scheme

Business Data Type supplementary component business value domain business identifier schemes are typically not enumerated, rather the scheme defines a regular expression or pattern that is used to populate its set of values and also used to validate values. BDT supplementry component business value domain business identifier schemes can be restricted from their source CDT supplementary component core value domain scheme, or less restricted BDT supplementary component business value domain scheme. These restrictions take the form of subsetting the set of allowed values.

[D429] Each BDT supplementary component business value domain business identifier scheme set of allowed values shall be equal to or less than the set of allowed values of its source CDT content component core value domain core identifier scheme or less restricted BDT content component core value domain business identifier scheme.

8.7.8.7.3.4 Business Data Type Supplementary Component Business Value Domain Business Code List

Business Data Type supplementary component business value domain business code lists contain lists of enumerated code values.

[D430] Each BDT supplementary component business value domain business scheme or list shall contain one or more business code values.

BDT supplementary component business value domain business code lists can be equal to or a restriction of, but never an extension of, their source CDT supplementary component core value domain core code list, or less restricted BDT supplementary component business value domain business code list.

[D431] Each BDT content component business value domain business code list set of allowed values shall be equal to or less than the set of allowed values of its source CDT content component core value domain core code list or less restricted BDT content component core value domain business code list.

8.7.8.7.3.4.1 Business Data Type Supplementary Component Business Value Domain Business Code Value Identification Metadata

A BDT supplementary component business code value will contain an identification metadata class consisting of a name and optional business term or terms.

[D432] A BDT supplementary component business value domain business code list business code value shall have zero or one identification metadata classes.

[D433] A BDT supplementary component business value domain business code list business code value shall have zero or one names that is unique within the set of business code values for a business code list.

A BDT supplementary component business code list may also have several business terms. These business terms are synonym terms under which the business code value is commonly known and used in business.

[D434] Each BDT supplementary component business code value shall have zero or more business terms.

8.7.8.7.3.4.2 Business Data Type Supplementary Component Business Value Domain Business Code Value Localized Metadata

The BDT supplementary component business value domain business scheme or list localized information class contains the relevant information necessary to associate native language identification of schemes or lists.

[D435] A BDT supplementary component business value domain business scheme or list shall have zero or more localized information classes

[D436] Each occurrence of a BDT supplementary component business value domain business scheme or list localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the scheme or list in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the scheme or list is commonly known and used in a business expression in that language.

[D437] BDT supplementary component business value domain business scheme or list localized metadata shall be in the language identified by the language and locale code.

8.7.8.7.4 Business Data Type Supplementary Component Business Value Domain Usage Rule

BDT supplementary component business value domains may have usage rules. Each usage rule defines a constraint that describes specific conditions that are applicable to the BDT supplementary component value domain. The BDT supplementary component business value domain usage rules represent the specific application of a BDT supplementary component value domain in its role of expressing the value domain of its BDT supplementary component. BDT supplementary component business value domain usage rules can be either unstructured – expressed as free form text, or structured – expressed in a formal language.

[D438] A BDT supplementary component business value domain shall have zero or more usage rules.

Usage rules will be defined at the lowest possible level of the hierarchical structure to which they apply.

[D439] BDT supplementary component business value domain usage rules shall not replicate BDT, BDT content component, BDT content component business value domain, or BDT supplementary component usage rules.

[D440] A BDT supplementary component business value domain usage rule shall have an identifier that is unique amongst all usage rules for the library of which it is a part.

[Note] – Usage Rule Identifier Structure

There are no specific rules for the structure of usage rule identifiers. Implementers are free to choose any structure providing it guarantees uniqueness within the group of usage rules of a library.

The BDT supplementary component business value domain usage rule constraint is the formal expression of the usage rule. The constraint can be structured or unstructured. An unstructured constraint will be expressed as free form text.

[D441] An unstructured BDT supplementary component business value domain usage rule constraint shall have a free form text expression that fully details the usage rule.

A BDT supplementary component business value domain formal constraint is a constraint that is expressed in a formal language such as the UML OCL or OMG SBVR.

[D442] A structured BDT supplementary component business value domain usage rule constraint shall have a formal constraint language expression.

BDT supplementary component usage rule constraint types must also be specified. The constraint type value is taken from a constraint type code list.

[D443] Every BDT supplementary component usage rule expressed as a formal constraint shall have a constraint type taken from a constraint type code list.

[Note] –Constraint Type Code List

UN/CEFACT will publish and make freely available a Formal Constraint Type Code List for use in support of this rule.

BDT supplementary component usage rules will also have condition types that identify when the formal constraint should be applied.

[D444] Every BDT supplementary component business value domain usage rule shall have a condition type.

[D445] Every BDT supplementary component business value domain usage rule condition type shall be one of **`pre-condition`**, **`post-condition`**, or **`invariant`**.

8.7.8.7.4.1 Business Data Type Supplementary Component Business Value Domain Usage Rule Identification Metadata

Although the unique identifier is sufficient to differentiate one usage rule in a library for all other usage rules, a BDT supplementary component business value domain usage rule may also have an identification metadata class that provides additional information.

[D446] A BDT supplementary component business value domain usage rule shall have zero or one identification metadata classes.

The BDT supplementary component business value domain usage rule contains a unique name that semantically differentiates it from all other named usage rules for the BDT.

[D447] A BDT supplementary component business value domain usage rule shall have zero or one names that is unique within the group of usage rules of a BDT.

A BDT supplementary component business value domain usage rule may contain business terms. BDT supplementary component business value domain usage rule business terms are synonym terms under which the BDT supplementary component business value domain usage rule is commonly known and used in business.

[D448] Each BDT supplementary component business value domain usage rule shall have zero or more business terms.

8.7.8.7.4.2 Business Data Type Supplementary Component Business Value Domain Usage Rule Identification Metadata Localized Information

The BDT supplementary component business value domain usage rule may have localized metadata that is used to provide other language expressions of its name and business term or terms.

[D449] A BDT supplementary component business value domain usage rule shall have zero or more localized information classes

[D450] Each occurrence of a BDT supplementary component business value domain usage rule localized information class shall contain:

- **Language Code (mandatory):** A code which identifies the language. *Internet Engineering Task Force RFC 3066 of January 2001* shall be used as the authoritative source for code values.
- **Name (optional):** The name of the usage rule in a language other than English.
- **Business Term (optional, repetitive)**: A synonym term in another language under which the usage rule is commonly known and used in a business expression in that language.

[D451] BDT content component business value domain usage rule localized metadata shall be in the language identified by the language and locale code.

9 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.

[Note] – Context Mechanism

The context mechanism is being more robustly defined in a separate UN/CEFACT Context Methodology specification. Once the final version of that specification is published, this section will be deprecated.

9.1 Overview

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

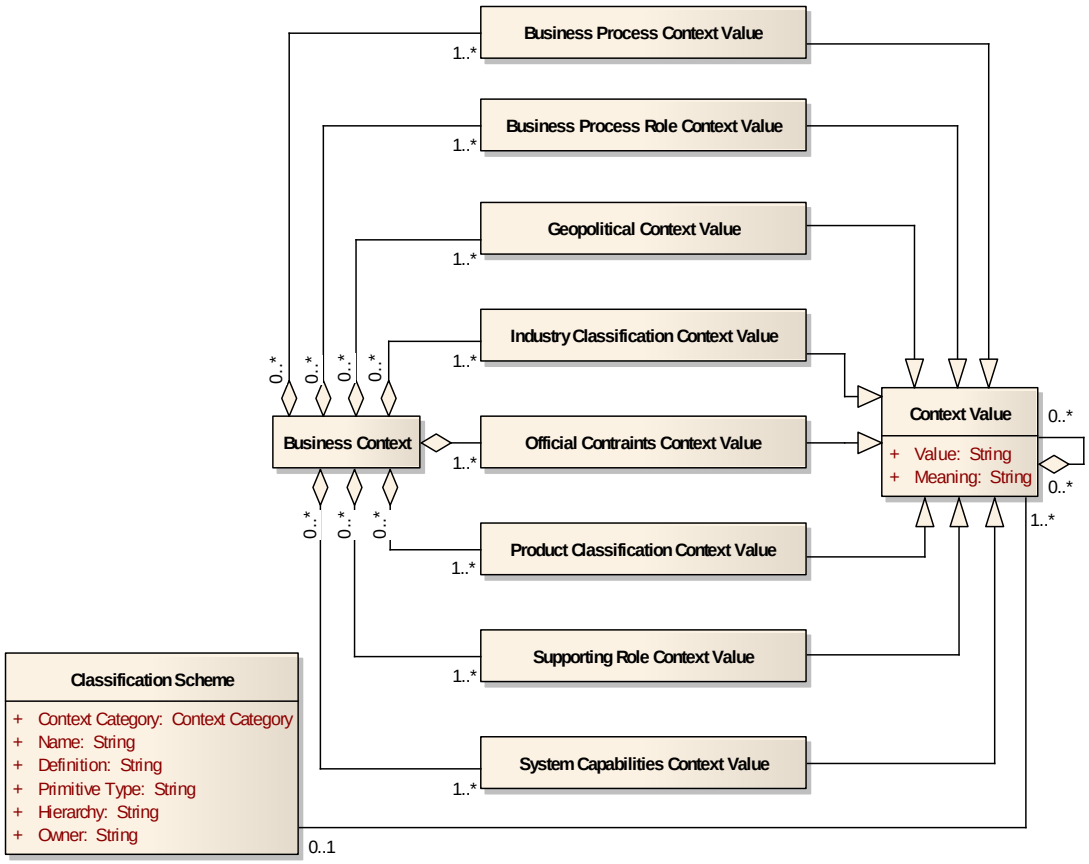


Figure 9-1. Core Components Context Definition Model

The CCs have no context independent of their use.

9.2 Business Context

[X1] Business context shall contain the combination of values for all approved context categories so as to define a unique and meaningful business context.

In order to ensure uniqueness, every business context will have assigned a:

- **Unique Identifier (mandatory):** The identifier that references the business context in a unique and unambiguous way.
- **Version Identifier (mandatory):** An indication of the evolution over time of the business context instance.

[X2] Each business context shall have a unique identifier within the library of which it is a part.

[X3] Each business context shall have a unique version identifier within the library of which it is a part.

9.3 Context Values

Each business context will contain the combination of values for all approved context categories so as to define a meaningful business context. Each business context will contain a value for each defined context category in order to describe the business context in an unambiguous and formal way.

[X4] When describing a specific business context, a value or set of values shall be assigned to each of the approved context categories.

[X5] Context values shall be defined as one of the eight recognized types—business process context value, product context value, industry context value, geopolitical context value, official constraints context value, business process role context value, supporting role context value or system capabilities context value.

[X6] Each context value shall include the following attributes:

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

[Note] – Context Value

The context value is derived from a business process model which presumably uses values that have their meaning defined somewhere. For example, if the value is taken from a code list (specified in the classification scheme), then the meaning of the code should be provided by the code list specification. As an alternative solution, the meaning could optionally be a uniform resource identifier that points to the definition.

9.4 Context Classification Scheme

Context values may belong to a particular classification scheme. The classification scheme defines all relevant information about the context value to allow it to be unambiguously understood and used. Context values that belong to a particular

classification scheme that allows a hierarchy, may have a hierarchical contains relation with another context value belonging to the same classification scheme.

[X7] Context classification schemes shall include the following attributes:

- **Context category (mandatory):** Name used to identify the approved context category for which the classification scheme can be used.
- **Name (mandatory):** Name under which the classification scheme is known.
- **Definition (mandatory):** Definition of the classification scheme.
- **Primitive Type (mandatory):** Primitive type that is used for the representation of a context value in the classification scheme.
- **Hierarchy (mandatory):** Indicator describing whether the classification scheme supports a hierarchical description of the context.
- **Owner (mandatory):** Organization that is responsible for the classification scheme.

9.5 Categories

Context categories exist to allow users to uniquely identify and distinguish between different business contexts. Eight context categories have been identified (Table 9-1). Each of the identified categories, unless otherwise stated, uses a standard classification to provide values for the category. Constraint rules, and therefore BIEs, are tied to a particular set of standard classifications for identifying and distinguishing contexts.

Table 9-1. Approved Context Categories

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

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

9.5.1 Business Process Context

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

[X8] Assigned business process contexts shall be from the standard hierarchical classification: provided as part of the UN/CEFACT *Catalogue of Common Business Processes*.

[X9] Business process context values may be expressed as a single business process, or as a hierarchical set of business processes.

[X10] Business process context values may be taken from extensions to the business processes described in the *UN/CEFACT Catalogue of Common Business Processes* as provided for in that document.

[X11] When business process extensions are used, they shall include full information for each value sufficient to unambiguously identify which extension is providing the value used.

9.5.2 Product Classification Context

The product classification context describes those aspects of a business situation related to the goods or services being exchanged by, or otherwise manipulated, or concerned, in the business process. Recognized code lists exist that provide authoritative sources of product classification contexts.

[X12] A single value or set of values may be used in a product classification context.

[X13] If a hierarchical system of values is used for product classification context, then these values may be at any level of the hierarchy.

[X14] If more than one classification system is being employed, an additional value specifying which classification scheme has supplied the values used shall be conveyed.

[X15] Product classification context code values shall be taken from recognized code lists to include:

- Universal Standard Product and Service Specification (UNSPSC)
  - Custodian: GS1
- Standard International Trade Classification (SITC Rev .3)
  - Custodian: United Nations Statistics Division (UNSD)
- Harmonized Commodity Description and Coding System (HS)
  - Custodian: World Customs Organization (WCO)
- Classification Of the purposes of non Profit Institutions serving households (COPI)
  - Custodian: UNSD (This provides a mapping between the first three.)

9.5.3 Industry Classification Context

The industry classification context provides a description of the industry or sub-industry in which the business process takes place.

[X16] An industry classification context may contain a single value or set of values at any appropriate level of the value hierarchy.

[X17] The industry classification context value hierarchy must be identified.

[X18] Industry classification context code values shall be taken from recognized code lists to include:

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

[Note] – Industry Classification Schemes

There are many other industry classification schemes that may be used for industry classification context.

9.5.4 Geopolitical Context

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

[X19] Geopolitical context shall consist of appropriate continent, economic region, country, and region identifiers.

[X20] Geopolitical context may associate one or more values with any component.

[X21] Geopolitical context shall employ the following values:

- Continent
- Country – ISO 3166.1

- Country Sub-entity - ISO 3166.2
- Economic Region
- Global
- Multi lateral Organizations

[X22] At any level of geopolitical context, a value may be a single value, a named aggregate or cross-border value.

[X23] Geopolitical context values shall be structured as follows:

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

[X24] Points in geopolitical context values shall be specified by either a single value, or combination of values.

[X25] The full path of the geopolitical context value must be used to understand the hierarchy when complex constructs are employed.

[X26] A specific level in the geopolitical context value is understood to inherit all of the properties within its specific path except where otherwise specified.

[X27] Geopolitical context values shall be taken from ISO 3166.1 and 3166.2.

9.5.5 Official Constraints Context

The official constraints context category describes those aspects of the business situation that result from legal or regulatory requirements and similar official categories. This category contains two distinct parts:

- Regulatory and Legislative. These are normally unilateral in nature and include such things as customs authority regulations.
- Conventions and Treaties. These are normally bi- or multilateral agreements and as such are different from regulatory and legislative constraints.

[X28] The official constraints context shall consist of at least two values:

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

Because there is no known global classification of all official constraints contexts as used here, any implementation must provide a set of recognized official constraints classifications for use within the appropriate core components registry implementation.

[X29] Individual core component implementations shall register used official constraint classification schemes with the appropriate supporting core components registry implementation.

9.5.6 Business Process Role Context

The business process role context describes those aspects of a business situation that are specific to an actor or actors within the business process. Its values are taken from the set of role values provided by the *UN/CEFACT Catalogue of Common Business Processes*. A business process role context is specified by using a value or set of values from this source.

[X30] Business process role context values shall be taken from an approved list provided by the business process model library being employed.

[X31] The *UN/CEFACT Catalogue of Common Business Processes* shall be the definitive source of business process role context values for all UN/CEFACT BIEs.

9.5.7 Supporting Role Context

The supporting role context identifies those parties that are not active participants in the business process being conducted but who are interested in it. A supporting role context is specified with a value or set of values from a standard classification.

[X32] Supporting role context values shall be taken from the UN/EDIFACT code list for DE 3035 party roles.

[Note] – Code List Duplication

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

9.5.8 System Capabilities Context

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

[X33] Systems capabilities context values shall consist of pairs of values. Each pair shall be comprised of an identification of the referenced classification scheme and the value(s) being employed.

[Note] – Information Systems Classification

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

9.6 Context Values

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

[X34] The **`In All Contexts`** value shall be a valid value for every context category except for official constraints context.

[X35] The value **`None`** shall be a valid value for official constraints context.

10 Definition of Terms

Aggregate Business Information Entity (ABIE) – A collection of related pieces of business information that together convey a distinct business meaning in a specific business context. Expressed in modelling terms, it is the representation of an object class, in a specific business context.

Aggregate Business Information Entity Property – A business information entity property for which the permissible values are expressed as a complex structure, represented by an aggregate business information entity.

Aggregate Core Component (ACC) – A collection of related pieces of business information that together convey a distinct business meaning, independent of any specific business context. Expressed in modelling terms, it is the representation of an object class, independent of any specific business context.

Aggregate Core Component Property – A unique property of the Aggregate core component that is related to the concept of the aggregate core component. An aggregate core component property is either an ASCC or a BCC.

Aggregation – A a special form of Association that specifies a whole-part relationship between the aggregate (whole) and a component part.

Artefact – A piece of information that is produced, modified, or used by a process. An artefact can be a model, a model element, or a document. A document can include other documents. CCTS artefacts include all registry classes and all subordinate named constructs of a registry class.

Associated Aggregate Business Information Entity – An aggregate business information entity that is either a uml **`aggregationkind=shared`** or **`aggregationkind=composite`** association to an associating aggregate business information entity. An associated aggregate business information entity is the child in a parent child association between aggregate business information entities.

Associated Aggregate Core Component – An aggregate core component that is a uml **`aggregationkind=shared`** association to an associating aggregate core component. An associated aggregate core component is the child in a parent child association between aggregate core components.

Associating Aggregate Business Information Entity – An aggregated business information entity that is either a uml **`aggregationkind=shared`** or **`aggregationkind=composite`** association to an associated aggregate business information entity. An associating aggregate business information entity is the parent in a parent child association between aggregate business information entities.

Associating Aggregate Core Component – An aggregated core component that has a uml **`aggregationkind=shared`** association to an associated aggregate core component. An associating aggregate core component is the parent in a parent child association between aggregate core components.

Association Business Information Entity (ASBIE) – A business information entity which defines the role of a specific aggregate business information entity (known as the associated ABIE) associated to another aggregate business information entity

(known as the associating ABIE). An association business information entity functions as an ABIE property of the associating ABIE.

Association Business Information Entity Property – A business information entity property for which the permissible values are expressed as a complex structure, represented by an aggregate business information entity.

Association Core Component (ASCC) – a core component which defines the role of a specific aggregate core component (known as the associated ACC) associated to another aggregate core component (known as the associating ACC). An association core component functions as an ACC property of the associating ACC.

Association Core Component Property – A core component property for which the permissible values are expressed as a complex structure, represented by an aggregate core component.

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

Based On – Use of an artifact that has been restricted according to the requirements of a specific business context.

Basic Business Information Entity (BBIE) – A business information entity that represents a singular business characteristic of a specific aggregate business information entity in a given business context. A basic business information entity is based on a basic core component and has a basic business information entity property that is based on a business data type which defines it value domain.

Basic Business Information Entity Property – A business information entity property for which the permissible values are expressed by simple values, represented by a data type.

Basic Core Component (BCC) – A core component which constitutes a singular business characteristic of a specific aggregate core component. It has a unique business semantic definition. A basic core component represents a basic core component property and is therefore of a core data type which defines its value domain. Basic core components function as properties of aggregate core components.

Basic Core Component (BCC) Property – A core component property for which the permissible values are expressed by simple values, represented by a data type.

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

Business Data Type – A data type consisting of one and only one business data type content component that carries the actual content plus zero or more business data type supplementary components giving essential extra definition to the CDT content component. Business data types have business semantics.

Business Data Type Content Component – Defines the primitive type used to express the content of a core data type.

Business Data Type Content Component Restriction – The formal definition of a format restriction that applies to the possible values of a core data type content component.

Business Data Type Supplementary Component – Gives additional meaning to the business data type content component.

Business Data Type Supplementary Component Restrictions – The formal definition of a format restriction that applies to the possible values of a business data type supplementary component.

Business Domain – A distinct group of profit making enterprises, such as chemical domain, oil and gas domain, automotive domain.

Business Information Entity (BIE) – A business information entity is a context specific instantiation of a core component that constitutes a piece of business data or a group of pieces of business data with a unique business semantic definition in a specific business context.

Business Information Entity (BIE) Property – A business characteristic belonging to the Object Class in its specific business context that is represented by an aggregate business information entity.

Business Libraries – A collection of approved process models specific to a line of business (e.g., shipping, insurance).

Business Process – The business process as described using the *UN/CEFACT Catalogue of Common Business Processes*.

Business Process Context – The business process name(s) as described using the *UN/CEFACT Catalogue of Common Business Processes* as extended by the user.

Business Process Role Context – The actors conducting a particular business process, as identified in the *UN/CEFACT Catalogue of Common Business Processes*.

Business Semantic(s) – A precise meaning of words from a business perspective.

Business Term – This is a synonym of the dictionary entry name under which the artefact is commonly known and used in business. A CCTS artefact may have several business terms.

Cardinality – An indication of the minimum and maximum occurrences for a characteristic: not applicable (0..0), optional (0..1), optional repetitive (0..*) mandatory (1..1), mandatory repetitive (1..*), fixed (n..n) where n is a non-zero positive integer.

Catalogue of Business Information Entities – This represents the approved set of Business Information Entities from which to choose when applying the core component discovery process

CCL – see core component library.

Classification Scheme – This is an officially supported scheme to describe a given context category.

Composition – A composition is a strong form of aggregation association that requires that the component part only belongs to a single parent object, and only exists as long as that parent object exists.

Context – Defines the circumstances in which a business process may be used. This is specified by a set of context categories known as business context.

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

Controlled Vocabulary – A supplemental vocabulary used to uniquely define potentially ambiguous words or business terms. This ensures that every word within any of the core component names and definitions is used consistently, unambiguously and accurately.

Core Component (CC) – A core component is a semantic building block for creating clear and meaningful data models, vocabularies, and information exchange packages. Core components are used as the basis for creating business information entities.

Core Component Library – The core component library is the part of the registry/repository in which core components shall be stored as registry classes. The core component library will contain all the registry classes.

Core Component Property – A business characteristic belonging to the object class represented by a basic core component property or an association core component property.

Core Data Type (CDT) – A data type consisting of one and only one core data type content component that carries the actual content, plus one or more core data type supplementary components giving essential extra definition to the core data type content component. Core data types do not have business semantics.

Core Data Type Content Component – Defines the primitive type used to express the content of a core data type.

Core Data Type Supplementary Component – Gives additional meaning to the business data type content component.

Data Type Term – A component of the name of the data type dictionary entry name which represents the value domain. A data type term is taken from a common list that is also used to determine allowed representation terms. Whereas representation terms are never qualified, as they represent the data type, data type terms can be qualified to reflect restrictions on the value domain.

Definition – This is the unique semantic meaning of a core component, business information entity, business context or data type.

Dictionary – A collection of dictionary entry names for CCTS components.

Dictionary Entry Name – This is the official name of a CCTS component.

Facet – A facet is a constraining value that represents a component restriction of a BDT content or supplementary component so as to define its allowed value space.

Formal Constraint Language – A normative expression from a recognized constraint language specification such as the Unified Modeling Language Object Constraint Language.

Geopolitical Context – Geographic factors that influence business semantics (e.g., the structure of an address).

Industry Classification Context – Semantic influences related to the industry or industries of the trading partners (e.g., product identification schemes used in different industries).

Invariant – a form of constraint whose value must remain true throughout execution of a process.

Library – a collection of CCTS components for a specific purpose, organization or group of organizations.

Message Assembly – The process whereby business information entities are assembled into a usable message for exchanging business information.

Naming Convention – The set of rules that together comprise how the dictionary entry name for CCTS artefacts is constructed.

Object Class – The logical data grouping (in a logical data model) to which a data element belongs (ISO11179). The object class is the part of a core component or business information entity dictionary entry name that represents an activity or object... The object class is represented by an object class term. Object classes have explicit boundaries and meaning and their properties and behaviour follow the same rules

Object Class Term – A component of the name of a core component or business information entity which represents the object class to which it belongs.

Official Constraints Context – Legal and governmental influences on semantics (e.g. hazardous materials information required by law when shipping goods).

Package – A package is a collection of semantically unique Business Information Entities in a given context.

Pre-Condition – A condition that must be true just prior to execution of a process.

Post-Condition – A condition that must be true just after execution of a process.

Primitive Type – A primitive type, also known as a base type or built-in type, is the basic building block for the representation of a value as expressed by more complex data types.

Product Classification Context – 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).

Property Term – A semantically meaningful name that represents a distinguishing characteristic of the object class. Property terms occur naturally in the definition of the artefact to which it belongs...

Qualified Business Data Type – A qualified business data type contains restrictions on a business data type content or business data type supplementary component(s).

Qualifier Term – A word or group of words that help define and differentiate an item (e.g. a business information entity or a business data type) from its associated items (e.g. from a core component, a core data type, another business information entity or another business data type).

Registry – An information system that manages and references artifacts that are stored in a repository. The term registry implies a combination of registry/repository.

Registry class – The formal definition of all the common information necessary to be recorded in the registry by a registry artefact – a core component, a business information entity, or a data type.

Repository – an information system that stores artifacts.

Representation Term – The type of valid values for a basic core component or basic business information entity.

Restriction – restriction is the process of deriving a new data structure from an existing data structure under the following rules:

- you can reduce the cardinality range of any field from the existing data structure;
- you can restrict the range of allowed values for any field with a simple data type (e.g. string, number);
- you can add a semantic restriction which narrows the business scope of any field.

All valid instances of a new restricted data structure must also be valid instances of the existing data structure from which the new data structure was derived.

Supporting Role Context – 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 Context – This context category exists to capture the limitations of systems (e.g. an existing back office can only support an address in a certain form).

UMM Information Entity – A UMM information entity realizes structured business information that is exchanged by partner roles performing activities in a business transaction. Information entities include or reference other information entities through associations."

Unique Identifier – The identifier that references a registry class instance in a universally unique and unambiguous way.

Usage Rules – Usage rules describe a constraint that describes specific conditions that are applicable to a component in the model. Usage rules are at least expressed as free form text and may also be expressed in a formal language.

User Community – A user community is a group of practitioners, with a publicized contact address, who may define Context profiles relevant to their area of business. Users within the community do not create, define or manage their individual context needs but conform to the community's standard. Such a community should liaise closely with other communities and with general standards-making bodies to avoid overlapping work. A community may be as small as two consenting organizations.

Value Domain – A set of allowed values.

Version – An indication of the evolution over time of an instance of a core component, data type, business context, or business information entity.

XML schema – A generic term used to identify the family of grammar based XML document structure validation languages to include the more formal W3C XML Schema Definition Language, ISO 8601 Document Type Definition, or Schematron.

11 References

Electronic Business Extensible Markup Language (ebXML) -- Part 5: ebXML Core Components Technical Specification, ISO/TS 15000-5:2005 Version 2.01(ebCCTS)

Information Technology – Specification and standardization of data elements – Part 1: Framework for the specification and standardization of data elements, International Standardization Organization, ISO 11179-1:1999

Information Technology – Metadata registries (MDR) – Part 2: Classification, ISO 11179-2:Second Edition 2005-11-15

Information Technology – Metadata registries (MDR) – Part 3: Registry Metamodel and Basic Attributes, ISO 11179-3(e): Second Edition 2003/Cor 1:2004

Information Technology – Metadata registries (MDR) – Part 4: Formulation of Data Definitions, ISO 11179-4:Second Edition 2004-07-15

Information Technology – Metadata registries (MDR) – Part 5: Naming and Identification Principles, ISO 11179-5:Second Edition 2005-09-01

Information Technology – Metadata registries: Registration, ISO 11179-6: Second Edition 2005-01-15

Information Technology -- Programming languages, their environments and system software interfaces -- Language-independent datatypes - ISO/IEC 11404:1996 - 1st edition.

Tags For the Identification of Languages, Internet Engineering Task Force (IETF) RFC 3066 of January 2001

UN/CEFACT Core Components Technical Specification – Part 8 of the ebXML Framework, 15 November 2003

UN/CEFACT Modeling Methodology (UMM): UMM Meta Model – Foundation Module Version 1.0 Technical Specification, October 2006

UN/CEFACT Modeling Methodology (UMM): UMM Meta Model – Base Module Version 1.0 Technical Specification, October 2006

Unified Modeling Language™(UML) Superstructure, Version 2.1.1, 2007-02-03

Unified Modeling Language™(UML) Infrastructure, Version 2.1.1, 2007-02-04

12 Disclaimer

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

Copyright Statement