Creating A Single Global Electronic Market

ebXML The role of context in the re-usability of Core Components and Business Processes

ebXML Core Components

February 16, 2001

Version 1.01

1 Status of this Document

This is an ebXML specification for approval by the eBusiness community.

The document formatting is based on the Internet Society's Standard RFC format. This document has not been harmonized with the ebXML Core Components efforts. Distribution of this document is limited to the ebXML community.

This version:

ebXML The role of context in the re-usability of Core Components and Business Processes Ver 1.01

2 ebXML participants

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

Editing team: Mike Adcock, APACS
Sue Probert, Commerce One
James Whittle, e CentreUK
Gait Boxman, TIE
Thomas Becker, SAP

Contributors:

Tom Warner
Jim Dick
Rob Jeavons
David Connelly
Arofan Gregory
Martin Bryan
Mike Adcock
Eduardo Gutentag
Matthew Gertner
Polly Jan
Sharon Kadlec
Sally Wang
James Wertner
Todd Freter
Henrik Reiche
Chris Nelson
Martin Roberts
Samantha Rolefes

3 Table of Contents

4 Introduction

4.1 Summary of Contents of Document

It describes those contexts that have been identified as most critical in describing the use of generic Core Components for business information purposes. It also suggests source lists of context such as, for example ISO 3166 for country related contexts.

The document will also describe how new context categorisations can be added and used. This might include adding new categories, or refining existing ones. The refinement may include both addition and subtraction of sections of a context taxonomy.

This document contains the context definitions, the recommended sources; and examples of how these contexts may be applied in business use.

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 RFC 2119 [Bra97].

4.2 Context Defined

When a business process is taking place, the context in which it is taking place can be specified by a set of contextual categories and their associated values. For example, if an auto manufacturer is purchasing paint from a chemical manufacturer, the context values might be as follows:

Contextual Category	**Value**
Process	Procurement
Product Classification	Paint
Region (buyer)	France
Region (seller)	U.S.
Industry (buyer)	Not required (generic)
Industry (seller)	retail

Rules indicate which context values (or combination thereof) must be present in order for them to be applied, as well as the action to be undertaken if a match occurs. Actions include adding additional information to a functional unit, making this information optional, required or eliminating optional information. We might, for instance, specify that addresses associated with organizations in the U.S. region be required to include a state (which might otherwise be optional). Note that these contextual changes are made individually to the Core Components that make up a business document, and not to the business document itself.

Despite this underlying simplicity, complications arise in certain cases that make real-world implementation of context rules extremely tricky. Broadly speaking, these complications relate to scenarios where two rules both match the context, but have conflicting results, or where different results are reached depending on the order in which matching rules are applied. The following examples illustrate these two cases (and refer to the sample context given above):

- One rule could require that if the buyer is in the U.S. region, product description should not be included in invoice line items. Another specifies that if the seller is in France, the product description (in French) shall be included.
- One rule could require that if the buyer's industry is automotive, the product category should be added to the invoice line items. Another specifies that if a product category information entity exists and the seller's industry is chemicals, an attribute should be added to the product category to indicate the toxicity of the products in the category. If the toxicity requirement were applied first, the attribute would not be added (since the product category was not yet present). The outcome therefore depends on the order in which the rules are applied.

The problem with these types of situations is not so much that there is no way to resolve them. It is rather that there are many possible solutions with no clear way of deciding which to choose, and all are sufficiently complex to place a significant burden on the implementer.

Additional complications result from the potentially hierarchical nature of context values. For example, the possible values for region belong in a hierarchical space (e.g. continent, country, region, city, etc.). The region specification can therefore be very general or very specific. Since rules can match a general value (e.g. apply if the organization is in North America) or a specific value (e.g. apply if the organization is in Omaha, Nebraska), there must be some way of determining which rules to apply (any combination including all of them) if several match. This is because, in some cases, a specific rule may complement the general rule, while in others it may override it.

The main reason for using context is to allow reuse of components to enable maximum interoperability between trading partners. By working from a common set of components and agreeing the context for business processes, partners can clearly understand the data that is required to take part in a Business Process.

The following set of scenarios explain when context may be applied:

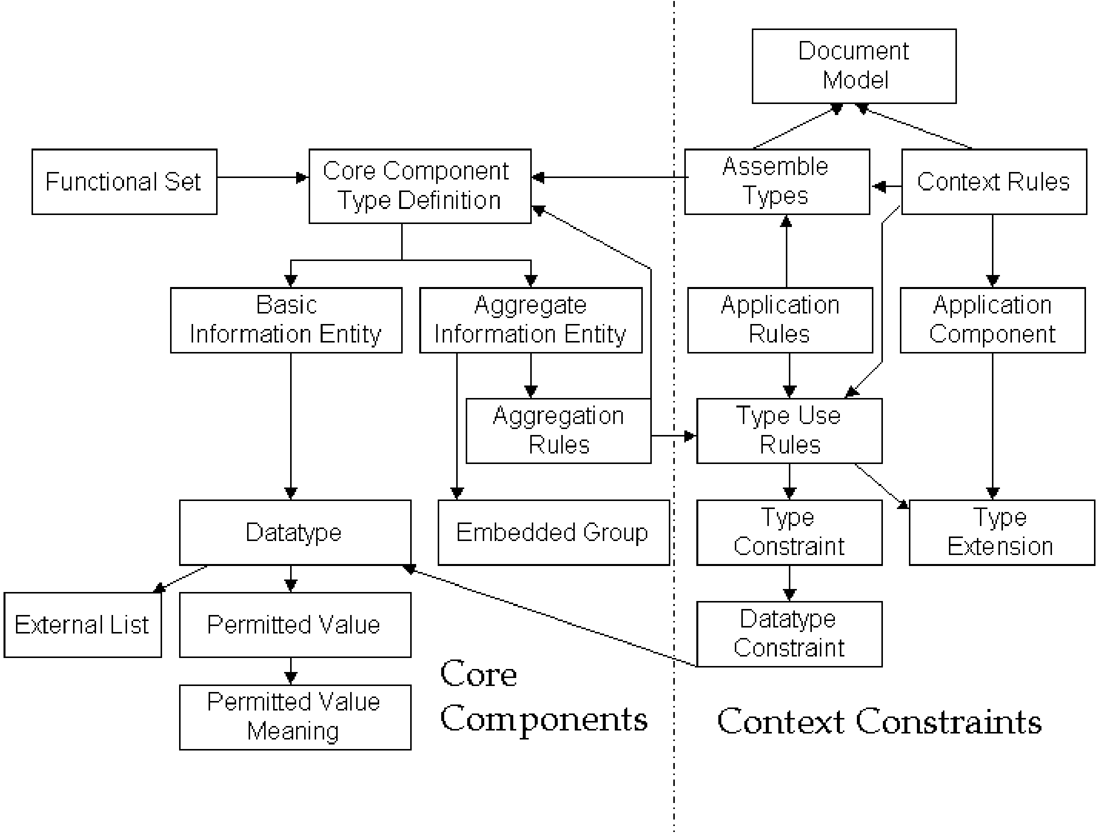
- Design Time - to create the minimum useful schema.
- Integration Time - Identify and help resolve data requirements conflicts required for business transactions
- Run Time - to express the business relationships between data.
    - Used by Trading Partners to validate the runtime document instances

- Navigation of the registry to find other data sets.
    - Need to hold the data about the context in the rules.
- Discovery Process for creating Core Components or extensions.
    - Core Components are discovered along with the business Context in which they are used.
    - Strip out context for Rules and Components
- Developing Document Templates for the Registry

5 Using Context Descriptors

5.1 Context-controlled Core Component Metamodel

The diagram below summarises the formal model for the Context Controlled Core Component Metamodel. The left-hand side of this diagram identifies the units specific to the definition of Core Components. The right-hand side of the diagram identifies the units used in defining Context Constraints. Type Use Rules are used both to manage component reuse within core components and to manage constraints within Context Constraints. The formal model for the Context-controlled Core Component Metamodel can be seen in Appendix 1.



5.2 Core Component Type Definitions

A Core Component Type Definition defines a reusable type of core component for which no pre-determined use name has been assigned.

Each definition is given a globally unique Identifier, which should be suitable for use as a database key.

A human-readable name for the type (ending in the word Type, e.g. AddressType), and a brief description of the purpose of the type, are also required.

By default a Core Component Type Definition is deemed to be restrictable or extendable. If this is not the case the isRestrictable or isExtendable boolean properties must be set to False.

5.2.1 Basic Information Entity

Where the types of data that are permitted for a Basic Information Entity are defined by an external agency the name of the MaintenanceAgency and the agency assigned identifier (id) must be recorded.

A formal definition of the relevant Datatype, defined in accordance with Part 2 of the XML Schema specification, must be associated with each Basic Information Entity.

If a data type is associated with an externally defined list of permitted values, then the URL of a resource that defines the set of currently approved permitted values should be recorded as an External Value List object.

If the list of permitted values is defined as part of the core component definition a Permitted Value List must be created. The list consists of one or more Permitted Values identified by a name that is unique within the list, each of which should be assigned one or more Permitted Value Meanings, each of which consists of a statement of the meaning assigned to the value and the IETF RFC1766 language code identifying the language in which the meaning has been defined.

5.2.2 Aggregate Information Entity

For each component forming part of an Aggregate Information Entity an Aggregation Rule that identifies a Type Use Rules object must be created. The Type Use Rules record the Name assigned to the referenced type within the location and, optionally, an explanation of the use to which the embedded component is being put within this component.

Where there are constraints on the number of times an embedded component can be used these are recorded as the MinMaxConstraints property.

Where there are constraints on the order in which sub-components within the aggregate are to be used an Embedded Group must be defined to identify whether the constraint applies to the use of a choice or sequence of objects.

5.2.3 Functional Set

A Functional Set is a set of two or more Core Component Type Definitions or Functional Sets that can be used to record information related to a single function in different ways.[1]

[1] For example, a location could be recorded as a postal address, a United Nations location code or as a set of co-ordinates as generated by a Global Positioning System. Which of this set of equivalent functions would be chosen for a particular message is context dependent.

5.3 Context Constraints

A Document Model is created by applying a set of Context Rules to a set of Core Component Type Definitions that have been “assembled” to meet a defined business process.

The Assemble Types object identifies the base Core Component Type Definitions, applies an appropriate sequence to the components and renames embedded components as required within the business process.
The Constraint Rules define modifications to be made to existing Core Component Type Definitions when used within specific contexts, and any Application Component needed to extend a core component or the document model.

Individual constraints are associated with a particular value within a named Taxonomy stored as a named context classification within an ebXML repository.

Where the constraint requires that the base definition of a core component be redefined the constraints are defined as a Type Constraint. Where the constraint applies to a facet of a Datatype definition it forms a Datatype Constraint that is associated with a specific Datatype.

5.4 Seeding Core Components

Lower level core components, either basic or aggregate information entities, can be re-used within higher level aggregates. Fundamentally, they are used "in the context of" the higher level aggregate. This is a purely structural context, not a business context, creating stereotype (i.e. fundamental or generic) information entities.

Recognizing that there are situations in which equivalent information can be expressed in several ways, relevant core components can be grouped together into Functional Sets. These provide a means by which a limited choice of stereotype information entities can be offered as alternative ways of specifying information for a particular function, e.g. a location can be specified as an address, a GPS reference, or a UN Locode. While the functional set is still a stereotype, the choice is dependent on a business context or contexts.

5.5 Using Core Components

Use of a core component without any modification in a particular business context creates a Substitute Information Entity. This is registered under a unique business name formed from the context and the stereotype component names.

Note: This is essential to record the industry sector(s) that use the substitute information entity, the context(s) in which they are used, and all the substitute information entities that use the Core Component.

Use of a core component with extensions (or indeed reductions) in a particular business context creates a Process Specific Entity. This is registered under a unique business name formed from the context and the stereotype component names.

Note: This is essential to record the industry sector(s) that use the substitute information entity, the context(s) in which they are used, and all the process specific entities that use the Core Component.

Substitute information entities and process specific entities are collectively Context Constrained Information Entities. Registration of all these, however numerous, is essential to achieve maximum re-use, to avoid "re-inventing the wheel", and to gain interoperability.

5.6 Building Business Documents

Business documents are built by drawing on the repository 'library' of components. The context descriptors that are registered for each component are used to select the appropriate context constrained information entities for the business document that is being built.

If no appropriate context constrained information entity exists, a new one must be created, according to the principles described in the previous section, and ideally using an existing stereotype. Registration of the new process specific information entity adds to the range of available context descriptors.

5.7 Beyond Re-use

If no appropriate existing stereotype exists, an industry grouping may need to:

- create additional Basic components for pieces of information which do not have already-defined Core Components. These are Domain Basic Components.
- use Core Component(s) to construct a non-core Aggregate Component, called a Domain Complex Component.
- use Core Component(s) and Domain Components to construct a non-core Complex Component, also known as a Domain Complex Component.
- use Domain Component(s) to construct a non-core Complex Component. These are also Domain Complex Components.

Ideally, Domain Components need to be recorded in the same detail as Core Components, complete with relevant Context(s). They are part of extensibility and ought to be registered so as to avoid 're-inventing the wheel'. Newcomers can re-use Domain Components and register any additional Context(s).

At some point, non-core Domain Components can become Core Components, according to criteria that judge the degree of re-use, and by a process yet to be defined.

5.8 Non-compliance Issue

This section raises two basic issues:

1) Extensibility
2) Registration

Registering Domain Components cannot be completely policed. Groups or companies might decide to use Core Components, extend them and invent their own Domain Components and never register them.

As a consequence, the use of these Domain Components will be limited to single use and will not become part of the ebXML standards community. Exact equivalents may well be re-invented in a different way, with different naming, and formally registered as a Domain Components.

Unregistered Domain Components:

- will hinder communication and interoperability between different communities.
- must not be, in any circumstances, be favored over formally registered equivalents.

6 Context Classifications

6.1 List of discovered Context Drivers

A large number of different context descriptors were considered, some of which were selected for full inclusion and definition.

This approach defines component reuse within business documents and now requires that some implementation take place before final decisions can be made regarding the value of all of these descriptors. It is intended that the thinking around possible descriptors not be discarded until their worth can better be judged.

- Region: *(Geopolitical)*
- Industry:
- Business Process:
- Product:
- Legislative: *(Legal)*
- Role
- Temporal:
- Information Structural Context:
- Application Processing:
- Service Level:
- Business Purpose:
- Virtual Marketplace:
- Contractual:

6.2 Classifications

These context classifications are the ones recommended by ebXML CCWG. It has been recognized that other classification schemes may be needed, and that it will be possible to reference other classification schemes for any of the identified context descriptors.

6.3 Business Process Context

The Business Process context relies on a classification based on the list of core business processes, but contains some additional information. It will be possible to indicate that some minor variations have been made to an existing core process; that a process not in the core is being used; or that an extension may be made at any level of the classification, to accommodate existing business processes.

Further, to be used meaningfully in qualifying variation within information entity structure, business process context descriptors may need to go to a finer level of detail than merely specifying the overall business process of which they are a part. This is especially true in a case where both trading partners may be adding information to a single functional aggregate at different points in the business process, and the optionality of that information is being determined by where in the process the information entity is used. (An example of these concepts can be found in the ebXML Methodology for the Discovery and Analysis of Core Components Ver 1.01, under section 9.4.3.6.1 Grammar)

The requirement to identify a particular event in the overall business process is complicated by the fact that there may be many players involved in a single business process, and even in a single "leg" of the overall exchange. This occurs when one or both trading partners have agents, as is often the case in payments processing where the trading partner's banks are involved in the exchange, and providing services to facilitate the overall business process. The existence of a portal - where a wide range of "en route" services may be provided - further complicates the issue.

6.4 Regional context

6.5 Regional classification

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

Global
    [Continent]
        [Economic Region]
            [Country] - ISO 3166.1
                [Region] - ISO 3166.2

There is no single hierarchy. At any level of the hierarchy, a value may be a single value, a named aggregate, or cross-border value. These values are structured as follows:

> **Single Value:** A single value as shown in the example under 6.5.1 List of Values, 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 named aggregates or cross-border constructions), 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 "Bidirectional", indicating the direction of cross-border context. Values may be named aggregates or single values.

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

6.5.1 List of Values

The following example shows an extract of the basic, single-value hierarchy of recommended values, based on the common ISO 3166 Country Codes.

Europe
  Eastern Europe
    AL – ALBANIA
    AM – ARMENIA
    etc.

6.6 Official Constraints Context

The official constraints context driver describes data use contexts, which are the result of standards, legal or regulatory requirements, contractual or business agreements, and similar "official" drivers. This classification is outlined as follows:

- Regulatory And Legislative (includes customs)
- Standards (includes ISO, Milspecs, etc.)
- Guidelines (best practices, unofficial standards)
- Conventions And Treaties (these are different from Regulatory and Legislative)
- Contractual And Trading Partner Agreement

This classification shall be structured as either:

- A free-text field with a qualifying text field to put in "schema" or reference describing what is contained in the text field (legal reference system, for example).
- A free text "code" field with the ability to reference the source.

6.7 Product Context

Definition:

The goods or services that the exchange of information describes or enables.

The subject of the transaction, eg the set of things that is being described.

6.7.1 Sources for Recommended Classifications

- United Nations Standard Product and Service Code (UN/SPSC)
  Custodian: United Nations
- Standard International Trade Classification
  (SITC Rev .3)
  Custodian: United Nations Statistics Division (UNSD)
- World Trade Organization (WTO)
  The "Harmonized Commodity Description and Coding System" (HS)
  Custodian WTO

- Classification Of the purposes of non Profit institutions serving households (COPI)
  Custodian: UNSD (This provides a mapping between the first three.)

Note: Others may include EAN.UCC codes or Global Trade Item Number (GTIN).

6.7.2 Structure

Context rules may be associated with each structure level, and more than one value may be specified for defining the use of a particular information entity.

6.8 Industry Context

Definition:

The industry or sub-industry in which the information exchange takes place.

An Industry is an organisation or group of organisations involved in service, commercial or institutional activity.

6.8.1 Sources for Recommended Classifications

- International Standard Industrial Classification(ISIC)
  Custodian: UNSD
- United Nations Standard Product and Service Code(UN/SPSC)
  Custodian: United Nations
  (Top level Segment (digits 1 and 2) used to define industry.)

6.8.2 Structure

Hierarchical structure as defined by existing standard. Context rules may be associated with each structure level, and more than a single value may be specified when describing the use of an information entity.

6.9 Role Context

Definition:

Roles: Roles specify the party types (buyer, seller, assembler, catalog publisher, etc.) that interactively perform interface activities that collaboratively achieve a business objective.

Role Types: The ebXML Business Process Methodology Guidelines, which is a specialization of the UN/CEFACT Unified Modeling Methodology (UMM), specifies that roles must be one of the following role types:

Organisational: As the name implies, the “Organisational” role is for playing the role of an “organization” such as an enterprise, a company, or a factory to cite a few examples. Only an organization performs a particular role in an e-business process. An employee does not perform these activities. Authorization to perform an activity is granted at an organizational level.

Employee: The “Employee” role is used in business interactions that are performed by employees of an organization. An employee for business/legal reasons can only perform an employee role. Usually the details of the employee must be captured and stored/transmitted to another partner for auditing/liability processes when the two partner roles are not in the same organization. Authorization to perform an activity is granted on an employee level.

Functional: The “Functional” role is for the cases when either an employee or an organization can perform the interaction. So the functional role can be either an organizational or an employee role.

Initiator / Responder:

Initiator: The Initiator is the role that initiates the business process and contains the start state and initial activity.

Responder: The Responders is the role that interacts with the initiator in a business process and commercial transaction.

Sources for Recommended Classifications
Code List 3035 (UN/EDIFACT)
Data Element 98 (X12)

6.9.1 Structure

The Business Process will at least identify the initiator and responder role, which should be used by the context rules. In this case, there MUST be agreement between the classification used by the ebXML Business Process specifications and the classification used by the context rules.

For example, a business process specifies that a partner of party type “Buyer” sends a “Purchase Order Request” business document to a partner of party type “Seller”. The “Seller” responds with a “Purchase Order Acceptance” business document. In this case, the business process identifies the initiator role (i.e. the “Buyer”) and the responder role (i.e. the ”Seller”).

In this example, the roles are:
Buyer: An employee or organization that buys products for a partner type in the supply chain. The role type is *functional.*

Seller: An organization that sells products to partners in the supply chain. The role type is *organizational.*

The following business documents are needed to execute this business process:

Purchase Order Request: a request to accept a Purchase Order for fulfillment.

Purchase Order Acceptance: a document that confirms the creation, change or cancellation of a Purchase Order.

The “Purchase Order Request” business document must contain at least the following elements, based on the roles identified in the business process:

“BuyerParty” - derived from the core component “Party”

“SellerParty” - derived from the core component “Party”

So in order to assemble the “Purchase Order Request” business document, the following context rules will be applied, among others (these examples use XML syntax only):

```
<Rule Order =” 1” >
     <Condition Test=” Role = ‘Buyer’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>

<Rule Order =” 2” >
     <Condition Test=” Role = ‘Seller’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>
```

The buyer might want to identify its preferred carrier in the “Purchase Order Request” business document. In this case, another functional role is involved of party type “Carrier”. This implies another element in the “Purchase Order Request” business document:

“CarrierParty” derived from the core component “Party”

In this example, the business process does NOT define the “Carrier” role!

```
<Rule Order =” 3” >
     <Condition Test=” Role = ‘Carrier’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>
```

The seller might want to identify in the “Purchase Order Acceptance” business document, which department accepted the Purchase Order or even which employee accepted the Purchase Order. In this scenario, there might be an organizational role of party type “Procurement” and an employee role of party type “Purchasing Manager”. So the “Purchase Order Acceptance” might have the following party elements (which are all derived from the core component “Party”):

“BuyerParty”, i.e. the initiator role as identified in the business process
“SellerParty”, i.e. the responder role as identified in the business process
“CarrierParty”, *not identified by the business process*
“ProcurementParty”, *not identified by the business process*
“PurchasingManagerParty”, *not identified by the business process*

So in this example, the context variable “Business Process” identified two roles: “Buyer” and “Seller”. There are other roles though that are not directly relevant for the business process, but do affect the business documents exchanged as part of this business process, i.e. “Carrier”, “Procurement” and “Purchasing Manager”.

There might be a hierarchical dependency between the roles mentioned above that affects the order in which context rules need to be applied. For example:

```
<Rule Order =” 1” >
     <Condition Test=” Role = ‘Buyer’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>

<Rule Order =” 2” >
     <Condition Test=” Role = ‘Seller’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>

     <Rule Order =” 3” >
          <Condition Test=” Role = ‘Procurement’” >
               <Action ApplyTo=“ Party” >
               </Action>
          </Condition>
     </Rule>

          <Rule Order =” 4” >
               <Condition Test=” Role = ‘Purchasing
               Manager’” >
                    <Action ApplyTo=“ Party” >
                    </Action>
               </Condition>
          </Rule>

<Rule Order =” 5” >
     <Condition Test=” Role = ‘Carrier’” >
          <Action ApplyTo=“ Party” >
          </Action>
     </Condition>
</Rule>
```

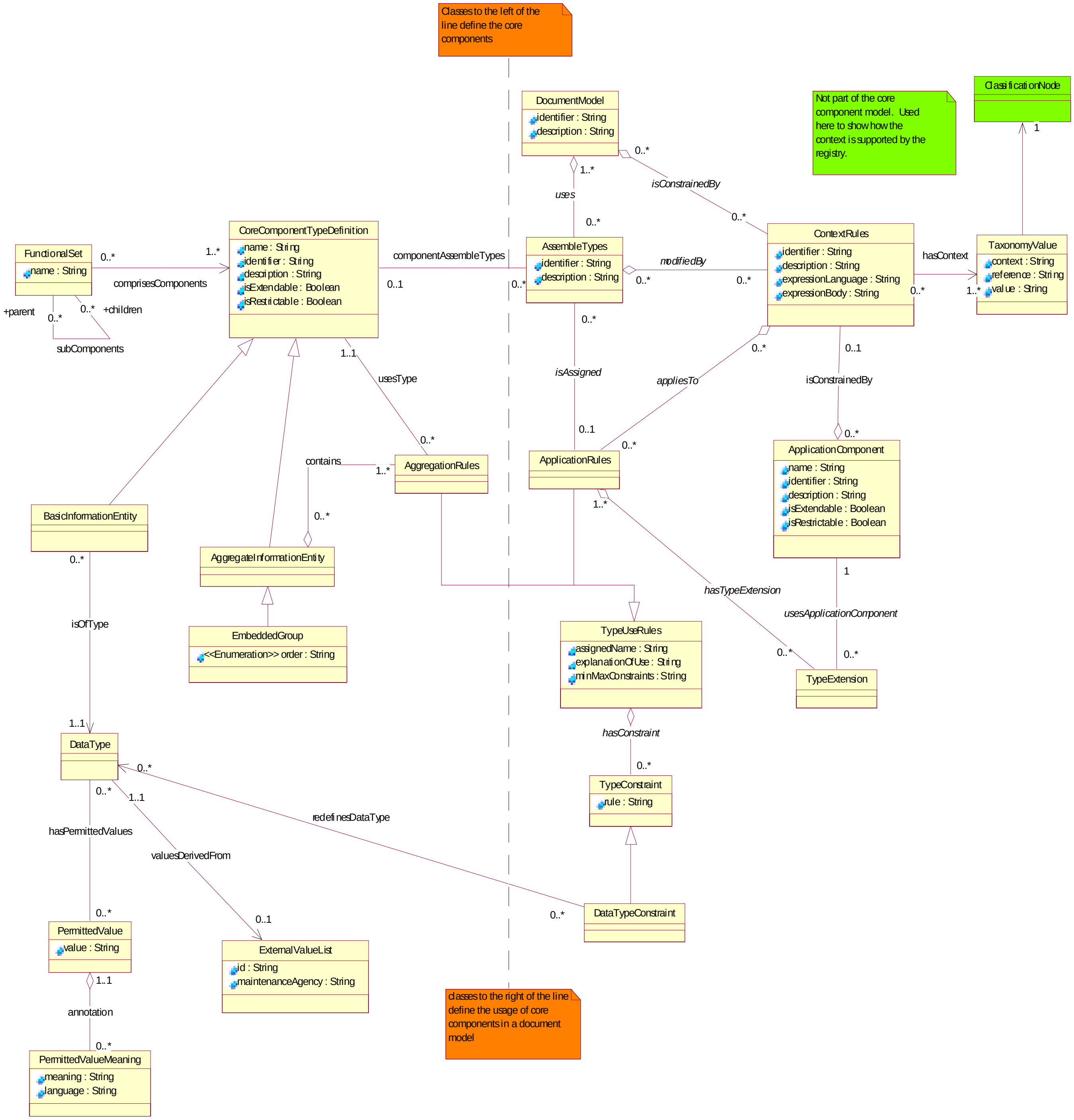
7 Registry Support for Taxonomies

7.1 Set of Data required to be published

The Registry Metamodel supports the requirement of attaching an arbitrary number of Classification Nodes to any Registered Entry. This is achieved by means of a Classification which can be associated with a Registered Entry, each instance of the Classification identifies a Classification Node. The top level node in the Classification Node tree can identify the type of classification (e.g. Geography) by means of its name. If this name does not give the unambiguous context within which the Registered Entry is classified then the Classification may optionally be associated with another ClassificationNode that provides the context for the Classification (e.g. LocatedIn).

The Classification Node is in itself a Registered Entry and by this means benefits from the versioning facility of the Registry.

8 Appendix 1


Classes to the left of the line define the core components
DocumentModel
identifier : String
description : String
0..*
1..*
uses
isConstrainedBy
0..*
0..*
Not part of the core component model. Used here to show how the context is supported by the registry.
ClassificationNode
1
FunctionalSet
name : String
0..*
1..*
comprisesComponents
+parent
0..*
0..*
+children
subComponents
CoreComponentTypeDefinition
name : String
identifier : String
description : String
isExtendable : Boolean
isRestrictable : Boolean
componentAssembleTypes
0..1
0..*
AssembleTypes
identifier : String
description : String
modifiedBy
0..*
0..*
ContextRules
identifier : String
description : String
expressionLanguage : String
expressionBody : String
hasContext
0..*
1..*
TaxonomyValue
context : String
reference : String
value : String
1..1
usesType
0..*
isAssigned
0..1
appliesTo
0..*
0..*
0..1
isConstrainedBy
0..*
AggregationRules
contains
1..*
0..*
ApplicationRules
1..*
ApplicationComponent
name : String
identifier : String
description : String
isExtendable : Boolean
isRestrictable : Boolean
BasicInformationEntity
0..*
AggregateInformationEntity
EmbeddedGroup
<<Enumeration>> order : String
isOfType
hasTypeExtension
1
usesApplicationComponent
0..*
0..*
TypeExtension
TypeUseRules
assignedName : String
explanationOfUse : String
minMaxConstraints : String
hasConstraint
0..*
TypeConstraint
rule : String
1..1
DataType
0..*
0..*
1..1
redefinesDataType
hasPermittedValues
valuesDerivedFrom
0..*
0..1
0..*
DataTypeConstraint
PermittedValue
value : String
1..1
annotation
0..*
PermittedValueMeaning
meaning : String
language : String
ExternalValueList
id : String
maintenanceAgency : String
classes to the right of the line define the usage of core components in a document model

9 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.

10 Contact Information

Team Leader
Name Arofan Gregory
Company Commerce One
Street Vallco Parkway
city, state, zip/other Cupertino, CA
Nation US

Phone:
EMail: arofan.gregory@commerceone.com

Vice Team Lead
Name Mike Adcock
Company APACS
Street Mercury House, Triton Court, 14 Finsbury Square
city, state, zip/other London EC2A 1LQ
Nation UK

Phone: +44-20-7711-6318
EMail: mike.adcock@apacs.org.uk

Editor
Name James Whittle
Company e centre^{UK}
Street 10,
Maltravers Street
city, state, zip/other London WC2R 3BX
Nation UK

Phone: +44-20-7655-9022
EMail: james.whittle@e-centre.org.uk

11 Copyright Statement

To be defined