



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

REQUIREMENTS SPECIFICATION MAPPING (RSM)

Documentation Template and conformity rules

Working Draft for Review
The UN/CEFACT Information Content Management Group (ICG) is making this document available to all UN/CEFACT Groups for review
ALL COMMENTS TO BE SENT TO: Michael.conroy@wanadoo.fr
These will processed at the September 2004 Forum meeting.

Version: 0
Release: 2

15

16

Table of Contents

17

18	1	REFERENCE DOCUMENTS.....	3
19	2	DEFINITIONS.....	3
20	3	INTRODUCTION.....	4
21	4	TEMPLATE OUTLINE.....	5
22	4.1	INTRODUCTION	5
23	4.2	REQUIREMENTS SPECIFICATION MAPPING BASIC OUTLINE.....	5
24	4.3	TITLE AND CONTENTS INFORMATION	6
25	4.4	OBJECTIVE SECTION	10
26	4.5	DEFINITION OF TERMS SECTION.....	10
27	4.6	TARGET TECHNOLOGICAL SOLUTION SECTION	10
28	4.7	DOCUMENT TO BE IMPLEMENTED SECTION	10
29	4.7.1	<i>Conceptual model section.....</i>	<i>10</i>
30	4.7.2	<i>Canonical model section.....</i>	<i>10</i>
31	4.7.3	<i>Business information entity reference section</i>	<i>11</i>
32	4.7.3.1	Basic business information entity section.....	11
33	4.7.3.2	Aggregate business information entity section	12
34	4.7.3.3	Associated documentation	14
35	4.7.4	<i>Draft core component reference</i>	<i>14</i>
36	4.7.4.1	Associated documentation	14
37	5	CONCLUSION	14
38	6	EXAMPLE.....	15
39	7	INTERIM UMM PROFILE FOR THE UML CLASS DIAGRAM	
40		ARTIFACT.....	16
41	7.1	PERMITTED CLASS DIAGRAM UML ARTIFACTS.....	16

42

43

43

44 **1 Reference documents**

- 45 ➤ CEFACT/TMWG/N090R10 UN/CEFACTs Modeling methodology
- 46 ➤ UN/CEFACT Modeling Methodology (UMM) User Guide
- 47 CEFACT/TMG/N093
- 48 ➤ Core Components Technical Specification – Part 8 of the ebXML Framework
- 49 Version 2.01
- 50 ➤ OMG Unified Modeling Language Specification

51 **2 Definitions**

52 These definitions are proposed to assist the reader in the use of this document and are

53 taken from the OMG UML Specification. For their formal definition consult the

54 relevant reference document.

Class diagram:	A class diagram shows the static structure of the information model, in particular, the things that exist, their internal structure, and their relationships to other things. A class diagram does not show temporal information. It is a diagram that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships.
Class:	A description of a set of objects that share the same attributes, operations, methods, relationships, and semantics.
Relationship:	A semantic connection among model elements. Examples of relationships include associations and generalizations.
Association:	The semantic relationship between two or more classifiers that specifies connections among their instances. An association may represent <i>an aggregation</i> (i.e., a whole/part relationship). In this case, the association-end attached to the whole element is designated, and the other association-end of the association represents the parts of the aggregation. <i>Composite aggregation</i> is a strong form of aggregation which requires that a part instance be included in at most one composite at a time and that the composite object has sole responsibility for the disposition of its parts. This means that the composite object is responsible for the creation and destruction of the parts. If a composite object is destroyed, it must destroy all of its parts. It may remove a part and give it to another composite object, which then assumes responsibility for it.
Role:	The named specific behavior of an entity participating in a particular context.
Cardinality:	The number of elements in a set.
Multiplicity:	A specification of the range of allowable cardinalities that a set

	may assume. Multiplicity specifications may be given for roles within associations, parts within composites, repetitions, and other purposes. Essentially a multiplicity is a (possibly infinite) subset of the non-negative integers.
Datatype:	A descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Predefined types include numbers, string and time. Userdefinable types include enumerations.
Constraint:	A semantic condition or restriction.
Generalization:	A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element (it has all of its properties, members, and relationships) and may contain additional information.
Sterotype:	A new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes.
Enumeration:	A list of named values used as the range of a particular attribute type. For example, RGBColor = {red, green, blue}. Boolean is a predefined enumeration with values from the set {false, true}.

55 3 Introduction

56 In order to successfully introduce a normalized form of business requirements
57 specifications based on the UMM philosophy, it will be necessary to develop and put
58 into place a phased approach that gradually enables TBG business groups to learn and
59 take increasing advantage of UMM.

60 During this transitional period, initial requirements will for the most part be orientated
61 around single processes that take account of UN/EDIFACT and the introduction of
62 XML implementations that are “document” based. As the groups improve their
63 knowledge of the UMM capabilities the focus could extend to other “non-document”
64 information objects.

65 The first workflow document template is the BRS. This complementary document
66 details the second template to be used in the UMM introductory phase where the TBG
67 business groups are required transform business requirements specifications into a
68 more formal form that is adapted to the UN/CEFACT standardization process and that
69 will ensure the consistent handling and registration of the specifications in question.
70 In essence this mapping process will transform the business requirements into
71 technical specifications.

72 In order to enable this a *Business Requirements Specification (BRS)* shall be
73 accompanied by a *Requirements Specifications Mapping (RSM)*. The RSM in
74 question transforms the aspects of a requirement that are of greatest concern to
75 business interests into those that are of concern to technical design interests.
76

76

77 **4 Template outline.**

78 **4.1 Introduction**

79 This outline provides the technical information that necessarily accompany a business
80 requirements specification in order it to be technically transformed into a given
81 technology solution.

82 **4.2 Requirements specification mapping basic outline**

83 The Requirements Specification Mapping (RSM) shall have the following basic
84 outline. This outline is considered to be the minimum that must be contained in the
85 RSM. It takes into consideration the use of a UMM profile for UML for the class
86 diagrams of documents. This does not exclude additional information being provided.

87 **1. Requirements specification mapping**

88 **1.1. Objective**

89 **1.2. Definition of terms**

90 **1.3. Target technology solution(s)**

91 **1.4. Document to be implemented**

92 **1.4.1. Conceptual model**

93 **1.4.2. Canonical model**

94 **1.4.3. Business information entity reference**

95 **1.4.3.1. Core Components Repository Used**

96 **1.4.3.2. Basic business information entities**

97 **1.4.3.3. Aggregate business information entities**

98 **1.4.3.4. Associated documentation**

99 **1.4.4. Draft core component reference**

100 **1.4.4.1. Associated documentation**

101 Each of these sections will be outlined in more detail in the rest of this document.

102

- 108 There are 6 pieces of information that have to be provided on this page:
109 1. The identification of the BRS that the document belongs to.
110 2. The unique identification of the document within the UN/CEFACT system.
111 3. The title of the document
112 4. The version of the document
113 5. The release of the document
114 6. The date of the document release (i.e. date of TBG final approval).
115

115

116

Document Change history log			
Date of change	Version	Paragraphs changed	Summary of changes

117

Figure 2

118 The second page contains the change history log for the document as outlined in
119 figure 2. This page retraces all the changes that have occurred between version and
120 release changes.

121 The third page provides the table of contents as outlined in figure 3. The table of
122 contents may be complemented as necessary but the basic paragraphs shall always be
123 provided.

Requirements Specification Mapping Table of contents	
1.	Objective
2.	Definition of terms
3.	Target technological solution
4.	Document to be implemented
4.1.	Conceptual model
4.2.	Canonical model
4.3.	Business information entity reference
4.3.1.	Core Components Repository Used
4.3.2.	Basic business information entities
4.3.3.	Aggregate business information entities
4.3.4.	Associated documentation
4.4.	Draft core component reference
4.4.1.	Associated documentation

124

125

Figure 3

126 These three basic pages shall be present in every Requirements Specification
127 Mapping. The rest of the document shall be composed of text and the necessary
128 UMM diagrams that are required to accompany the text. The UMM worksheet
129 information may be provided if this improves the comprehension of the specification.

130 Every RSM shall be accompanied with the XMI (XML Metadata Interchange) files
131 containing the canonical models that have to be provided with a technological
132 solution and the XMI files containing any draft core components that are being
133 presented with the document. Additional documentary files may be provided as
134 required. The XMI files will form the basis of the master data that is maintained
135 within the UN/CEFACT repository.

136 A standard XSL presentation shall be provided for each XMI artifact file (these being
137 initially Class diagrams and core component diagrams).

138 **4.4 Objective section**

139 This section shall define the basic business objective of the RSM. It shall identify in
140 context the complete set of class diagrams that have to be transformed into the target
141 technology solution.

142 **4.5 Definition of terms section**

143 All terms that are specific to the creation of the technological solution shall be defined
144 in this section.

145 **4.6 Target technology solution section**

146 This section shall provide the required technological solutions that have to be
147 developed. In some cases where merely draft core components are being transmitted
148 for integration in the repository this section shall indicate that no technological
149 solutions are to be provided.

150 **4.7 Document to be implemented section**

151 This section shall completely describe the electronic document that has to be
152 developed. If several documents are being specified for development then this section
153 shall be repeated for each of the concerned documents. This section may be divided
154 into several sections as indicated hereunder.

155 **4.7.1 Conceptual model section**

156 This section shall provide a pictorial representation of the conceptual model class
157 diagram that has been defined in the BRS. It shall provide a summary description of
158 the model. It shall highlight any specific conditions or constraints that are necessary
159 for the successful transformation of the model into its canonical form.

160 **4.7.2 Canonical model section**

161 The canonical model is a UMM compatible formalisation of the conceptual model
162 class diagram that makes use of standard core components. In some cases the
163 canonical model may be identical to the conceptual model. This would be the case
164 where the modeling analyst develops the conceptual model based on the UMM
165 compatible artifacts and the required core components.

166 The canonical model shall be used as the basis to achieve the transformation into the
167 required technology solution. It shall be shown in the RSM documentation in pictorial
168 form and it shall be provided in an associated file in an XMI form. It is the XMI file

169 that will be considered the master data for registration in the UN/CEFACT repository.
170 It is with this file that will be validated for consistency.

171 The canonical model contains the transformation non-standard core components that
172 are found in the conceptual model into standard core components.

173 **4.7.3 Business information entity reference section**

174 This section shall contain the complete list of all the business information entities that
175 are used within the canonical model. This section shall indicate the core component
176 repository release that has been employed in its elaboration.

177 The section is divided into three parts, the basic business information entities, the
178 aggregate business information entities and associated documentation. These content
179 of these sections are described in the following paragraphs.

180 **4.7.3.1 Basic business information entity section**

181 This section shall identify all the Basic business information entities (a business
182 information entity represents a singular business characteristic (class attribute) of a
183 specific object class (class) in a specific business context (BRS)) that are sourced
184 from the UML class attributes. The purpose of this section is to identify any
185 restrictions that have to be applied to the core component type in the context in which
186 it is being used. The restrictions may be of two type; restrictions that are applied to
187 the format or representation of the datatype, or restrictions/extensions that are applied
188 to an enumeration. Each BBIE shall be identified in the following table:

BBIE Term	Business	Core component type	Core component dictionary Entry Name	Restrictions	Code list extensions / Restrictions
InstatEnvelope. EnvelopeId		textType		Size limit 35 characters	

189 The table shall be filled in as follows:

- 190 1. The name of the BBIE. Each name is composed of its class name and the
191 attribute name as identified in the class.
- 192 2. The core component type that has been used to identify its datatype.
- 193 3. If the BBIE is based on an existing basic core component, the name of the
194 basic core component is indicated. BBIE may not necessarily have
195 corresponding basic core components since their particular characteristics do
196 not have a generic or generalized scope.
- 197 4. The restrictions that have to be applied to the generic datatype. These
198 restrictions may be, for example, size, format or particular pattern matching.
- 199 5. Any codelist restrictions or extensions that have to be applied to the BBIE.
200 Code list extensions may be required in a context where a standardized code
201 list does not exist.

202 4.7.3.2 Aggregate business information entity section

203 This section shall identify all the aggregate business information entities that have
204 been based upon an aggregate core component. The purpose of this table is to identify
205 the restrictions and/or extensions that have to be applied to the generic class in order
206 to satisfy the business requirement expressed in the aggregate business information
207 entity.
208

208 Each ABIE that is based upon an ACC shall be identified in the following table:

ABIE Business Term	Aggregate core component dictionary Entry Name	Restrictions (BCCs to be omitted)	Extensions
Address	Address	Identifier Format Additional Street Block Building Building Department Door Floor House Name House Number Inhouse Mail Line One Line Two Line Three Line Four Line Five Lot Post Box Room Suite Wing	

209 The table is completed as follows:

- 210 1. The ABIE class name that identified the class in the UML model.
- 211 2. The identification of the ACC that is the origin of the ABIE.
- 212 3. The identification of each BCC that is not to be employed in the ABIE.
- 213 4. The identification of ABIE that is used as an additional attribute to extend the
- 214 original ACC.

215

216 4.7.3.3 Associated documentation

217 This section shall identify any associated documentation that has to be registered with
 218 the canonical model in respect to the business information entities that have been
 219 identified. The associated documentation shall be provided in the form of one file per
 220 document. Each document shall be described in the following table:

Document name	File name	Description	BBIE or ABIE Cross reference
---------------	-----------	-------------	---------------------------------

221 4.7.4 Draft core component reference

222 This section shall identify all draft core components that are being proposed for
 223 registration in the core component repository. All new core components shall be
 224 provided as UML classes that define the core component and provide all its relevant
 225 characteristics. The UML classes shall be provided in the form of an XMI file. This is
 226 necessary in order to ensure that the core components can be loaded into UML
 227 development software for use in the development of compliant canonical models or
 228 eventually compliant conceptual models.

229 The XMI files will be validated for compliance with the UMM profile for UML in
 230 order to ensure consistency throughout the repository.

231

Dictionary entry name	Definition	Object class	Property term	Datatype qualifier	Representation term	CC type	Etc.
-----------------------------	------------	-----------------	------------------	-----------------------	------------------------	------------	------

232 This table shall be completed in compliance with the CCTS.

233 4.7.4.1 Associated documentation

234 This section shall identify any associated documentation that has to be registered with
 235 the canonical model in respect to the basic or aggregate core components that have
 236 been provided for registration in the core component repository. The associated
 237 documentation shall be provided in the form of one file per document. Each document
 238 shall be described in the following table:

Document name	File name	Description	BCC or ACC Cross reference
---------------	-----------	-------------	-------------------------------

239 5 Conclusion

240 The requirements set out in this document shall be considered the minimum
 241 requirement for a RSM document. The document may be completed with additional
 242 sections wherever it is deemed necessary to provide complementary information in
 243 order to ensure the successful transformation into a technological solution.

244 **6 Example**

245 The following example is provided to demonstrate how to move from a conceptual
246 model to a canonical model on a step-by-step basis.

- 247 1. Change all associations to compositions.
- 248 2. Delete root association cardinality.
- 249 3. Eliminate generalizations where possible.
- 250 4. Identify ABIEs
- 251 5. Assign core component types

252 Comment: A worked example using the INSTAT message will be inserted at a later
253 time.

254 7 Interim UMM profile for the UML class diagram 255 artifact

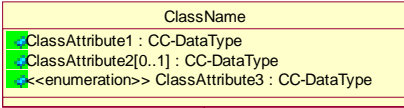
256 This section is provided as an interim solution to ensure the consistent development of
257 UMM class diagrams in the UML until the standard UMM profile for UML artifacts
258 have been developed by the TMG. It will also serve as the initial contribution to the
259 TMG project to develop the profile in question.


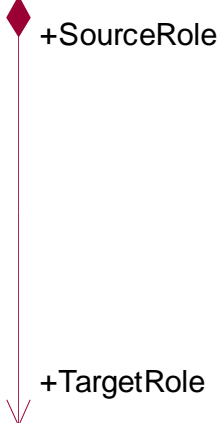
260 Through the application of this profile the well formedness of any provided UML
261 class diagrams will be ensured.

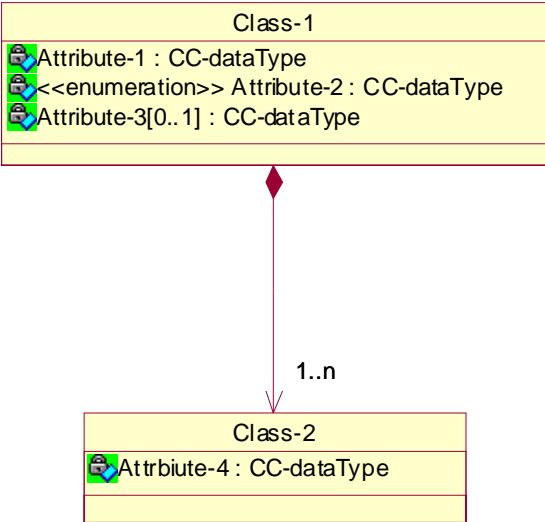
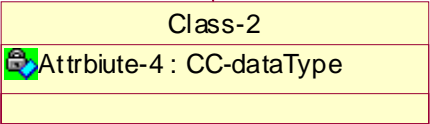
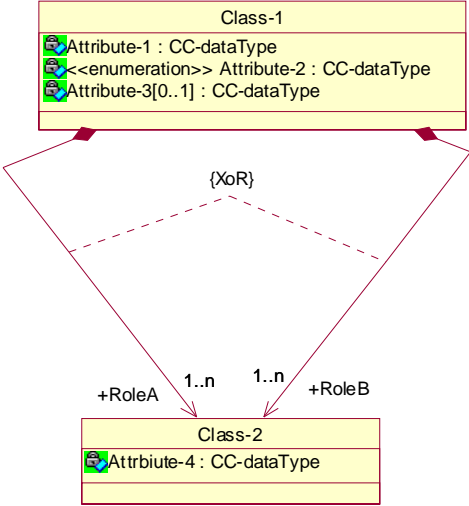
262 All UML class diagrams must respect the following rules in order for it to be
263 registered in the UN/CEFACT repository of canonical models.

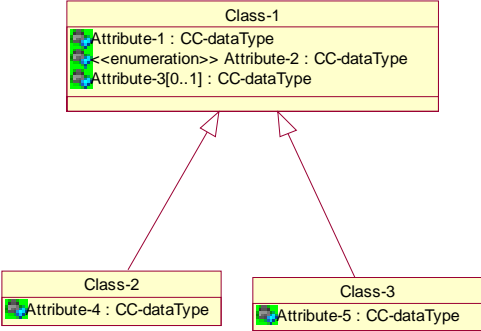
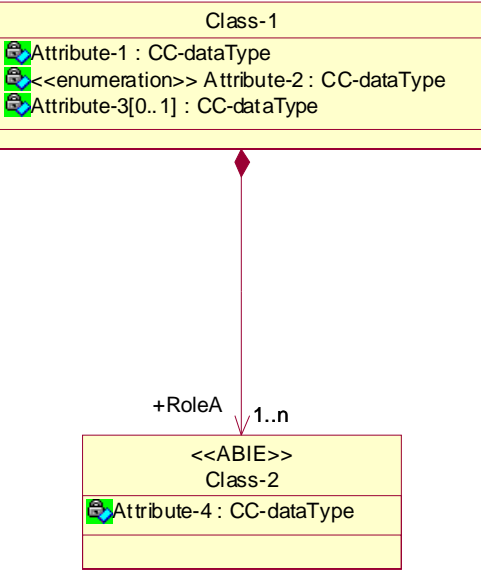
264 7.1 Permitted class diagram UML artifacts

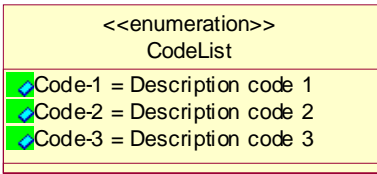
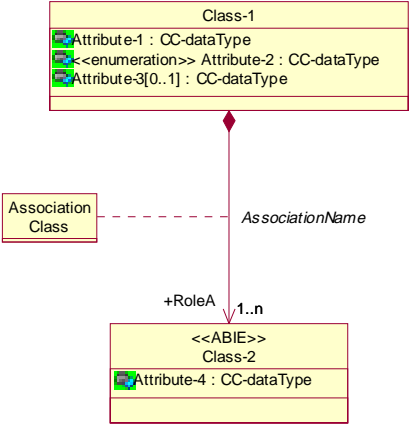
265 The following UML artifacts are permitted in UMM compliant class diagrams.

UML artifact	Description
	<p>Class:</p> <p>A description of a set of objects that share the same attributes, operations, methods, relationships, and semantics.</p> <p><i>Note: in the initial implementation of UML classes the well formedness or the use of operations and methods have not been defined.</i></p> <p><i>All classes shall be normalized. That is to say:</i></p> <ol style="list-style-type: none"> 1. <i>An attribute shall represent a single piece of information</i> 2. <i>An attribute shall appear only once (I.e. it cannot be repeated)</i> 3. <i>An attribute shall have a distinct name</i> 4. <i>Each instance of a class must be uniquely identifiable</i> 5. <i>There is no positional dependence between the attributes</i> 6. <i>All attributes contribute to the definition of the class</i>

	<p>Association: The semantic relationship between two or more classes that specifies connections among their instances.</p> <p><i>The only relationship permitted in a UMM compliant class diagram is a composite aggregation.</i></p> <p>Composite aggregation is a strong form of aggregation which requires that a part instance be included in at most one composite at a time and that the composite object has sole responsibility for the disposition of its parts. This means that the composite object is responsible for the creation and destruction of the parts. If a composite object is destroyed, it must destroy all of its parts. It may remove a part and give it to another composite object, which then assumes responsibility for it.</p> <p><i>All compositions shall be unidirectional.</i></p>
	<p>Role: The named specific behavior of an entity participating in a particular context.</p> <p><i>A role identifies a specific case of a generic class. In general only target roles should be used. Source roles should be avoided whenever possible. Source roles tend to identify constraints as opposed to behaviour.</i></p> <p><i>A rolename may be concatenated with a classname to provide a specific type of a generic class.</i></p>

 <pre> classDiagram class Class1 { Attribute-1 : CC-dataType Attribute-2 : <<enumeration>> CC-dataType Attribute-3[0..1] : CC-dataType } class Class2 { Attribute-4 : CC-dataType } Class1 "1" *-- "1..n" Class2 </pre>	<p>Cardinality: The number of elements in a set.</p> <p><i>The cardinality shall be used for an attribute uniquely to indicate a conditional attribute with the convention “[0..1]” that immediately follows the attribute name. No other attribute cardinality is permitted. A conditional attribute shall have an associated dependency table to describe the conditionality constraint.</i></p> <p><i>The cardinality of an association shall only appear at the target end of the association.</i></p>
 <pre> classDiagram class Class2 { Attribute-4 : CC-dataType } </pre>	<p>Datatype: A descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Predefined types include numbers, string and time. User definable types include enumerations.</p> <p><i>An attribute shall use a core components datatype.</i></p>
 <pre> classDiagram class Class1 { Attribute-1 : CC-dataType Attribute-2 : <<enumeration>> CC-dataType Attribute-3[0..1] : CC-dataType } class Class2 { Attribute-4 : CC-dataType } Class1 "1" -- "1..n" Class2 : +RoleA Class1 "1" -- "1..n" Class2 : +RoleB </pre>	<p>Constraint: A semantic condition or restriction.</p> <p><i>Only XoR constraints are allowed between associations. All associations constraints must be defined in a dependency table.</i></p>

 <pre> classDiagram class Class-1 { Attribute-1 : CC-dataType <<enumeration>> Attribute-2 : CC-dataType Attribute-3[0..1] : CC-dataType } class Class-2 { Attribute-4 : CC-dataType } class Class-3 { Attribute-5 : CC-dataType } Class-1 < -- Class-2 Class-1 < -- Class-3 </pre>	<p>Generalization: A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element (it has all of its properties, members, and relationships) and may contain additional information.</p> <p><i>Generalisations shall be avoided in the modelisation of documents.</i></p>
 <pre> classDiagram class Class-1 { Attribute-1 : CC-dataType <<enumeration>> Attribute-2 : CC-dataType Attribute-3[0..1] : CC-dataType } class Class-2 { <<ABIE>> Attribute-4 : CC-dataType } Class-1 *-- "1..n" Class-2 : +RoleA </pre>	<p>Stereotype: A new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes.</p> <p><i>Stereotypes shall only be used within a canonical model to identify classes that have been built from ACCs. Classes that have not been built from an ACC shall not be stereotyped.</i></p> <p><i>An attribute of a class that represents the same concept, as a BCC shall be stereotyped BBIE within a canonical model provided that this is not a class that originates from an ACC.</i></p> <p><i>The only other stereotype permitted within a class in a canonical model shall be the “enumeration” stereotype that shall be used to identify a code list.</i></p> <p><i>ACC and BCC Stereotypes shall be used in the core components repository to identify aggregated core components and basic core components accordingly.</i></p> <p><i>An ASCC stereotype is not required within the core component repository since the nature of the association is explicitly expressed in a composition association.</i></p> <p><i>An ASBIE stereotype shall not be used in a Canonical model since it is</i></p>

	<p><i>explicitly expressed through the use of a composition association and a role.</i></p>
 <pre> classDiagram class CodeList { <<enumeration>> Code-1 = Description code 1 Code-2 = Description code 2 Code-3 = Description code 3 } </pre>	<p>Enumeration: A list of named values used as the range of a particular attribute type.</p> <p><i>Enumerations shall be used to identify code lists.</i></p> <p><i>In the canonical model an enumeration shall be used to identify a restriction on a generic codelist</i></p>
 <pre> classDiagram class Class1 { <<class>> Attribute-1 : CC-dataType <<enumeration>> Attribute-2 : CC-dataType Attribute-3[0..1] : CC-dataType } class AssociationClass { <<association class>> } class Class2 { <<ABIE>> <<class>> Attribute-4 : CC-dataType } AssociationClass ..> Class1 : AssociationName AssociationClass ..> Class2 : +RoleA Class1 *-- Class2 : 1..n </pre>	<p><i>No association classes are permitted.</i></p> <p><i>No association names are permitted</i></p>