



1 **Briefing Paper: UBL's current status**
2 **with respect to schemas for**
3 **implementing Core Component**
4 **Types.**

5 **June 10, 2004**

6 **Document identifier:**

7 draft-mcgrath-UBLandCCTSschemas-0p1.sxw (and .doc)

8 **Location:**

9 **Author:**

10 Tim McGrath <tmcgrath@portcomm.com.au>

11 **Abstract:**

12 This paper attempts to describe the history and rationale behind the current schemas
13 used by UBL for implementing Core Component Types.

14 **Status:**

15 This is v.01 of the paper intended for consideration by the OASIS UBL ATG2 Liaison and
16 other interested parties.

17 Copyright © 2004 The Organization for the Advancement of Structured Information Standards
18 [OASIS]

19 **Table of Contents**

20	1 Introduction.....	3
21	2 Background.....	4
22	3 Schemas developed jointly by UBL,OAG and ATG2.....	5
23	4 Schemas developed for UBL 1.0.....	6
24	5 Alignment with OAGIS.....	8
25	6 Open Items for Core Component Type Schemas.....	10

26 **1 Introduction**

27 This paper has been prepared to consolidate a complex history of events leading up to the
28 present situation with respect to the development of a common set of schemas for realizing
29 ebXML Core Component Types.

30 The imminent UN/CEFACT ATG2 meeting has the opportunity to build upon the experience of
31 UBL and OAG and facilitate the process of alignment.

32 Of direct significance to UBL are the proposed actions to align our schemas with those of the
33 OAG. Particularly those relating to the formal attribute naming of Supplementary Components.
34 The exercise of reverse-engineering the Core component types exposed the structure of the
35 underlying data model and its supplementary components. This now enables us to have rules and
36 formalism in how the schemas can be derived from the models using the correct logical
37 components as stated in the Core Component Technical Specification.

38 UBL's desired outcome from ATG2 would be acceptance of the alignment plan with OAG as
39 outlined in section 5 and progress towards solutions to the open issues outlined in section 6.

40 These recommendations are based on minimizing the impact of change as well as satisfying the
41 requirements of the ebXML Core Component Technical Specification.

42 2 Background

43 Section 8. of the UN/CEFACT ebXML Core Component Technical Specification describes the
44 approved Core Component Types, the approved Core Component Type Content and
45 Supplementary Components, and permissible Representation Terms.

46 In March 2002 UBL produced its first realization of these types in XML Schema, as part of the
47 Op64 review package. There was only one schema module and all Core Component Types,
48 Supplementary Components and their secondary representation terms where defined in it. This
49 was based on the CCTS version 1.7 and as far as we were aware was the first XSD
50 implementation of these structures. Because the schema was automatically incorporated as part
51 of the schema generation Perl script it was largely invisible to UBL developers. SAP (through
52 Gunther Stuhec) did all of the maintenance on this.

53 This schema was updated to CCTS version 1.8 for the Op65 review in August 2002.

54 By the time we got to UBL Op70 in January 2003 to reflect CCTS 1.90, the schema now included
55 an attempt to identify Secondary Representation Terms separately from Core Component Types.
56 This release also saw the introduction of a Core Components Parameters schema to define
57 annotations used in all other schemas. Once again, this remained largely below the horizon for
58 UBL developers as it was built into the schema generation process.

59 In April 2003 Gunther presented his position paper on Common Core Components to the UBL
60 Naming and Design Rules group. This formally defined XSD datatypes for both Core Component
61 Types and Secondary Representation Terms.

62 UBL Op81¹ in August 2003 saw the Core Component Type and Core Component Parameter
63 schema modules transfer authorship to Chin Chee-Kai for use with the UBLish schema generator.

1 UBL Op80 in June 2003 was only a model release and so we did no work on the schemas.

64 3 Schemas developed jointly by UBL, OAG and 65 ATG2

66 In October 2003, Gunther Stuhec of SAP AG and Garret Minakawa of Oracle (together with Mark
67 Crawford of ATG2?) developed a new set of schemas. This saw the Core Component Types
68 schema module broken out into three. The Core Component Types schema, the Representation
69 terms schema (for defining datatypes for secondary representation terms of core component
70 types) and the Data Types schema (for customized extensions to datatypes defined in the other
71 two schemas).

72 With some minor modification, these were incorporated into UBL 1.0-Beta in November 2003.
73 The reasons for modification were:

- 74 · in Representationterm.xsd: the word "Representation" was mis-spelt in two statements (as
75 "Repepresentation")
- 76 · in Representationterm.xsd: it referenced the OAGI WG not UBL
- 77 · in Representationterm.xsd: there was no definition for the secondary representation term
78 'Picture' (the primary is Binary Object).
- 79 · analysis of the UBL-CoreComponentTypes found that some were not in the CCTS Version
80 2.0. These were: Name, NameType, ElectronicAddressType, ElectronicAddress,
81 GloballyUniquelidentifierType and GloballyUniquelidentifier
- 82 · some of the schemas, were not parseable XML files. They lacked either proper namespace
83 definitions, or, in certain places proper schema import elements. Xerces 2.5.0 discovered 3
84 errors in the RepresentationTerms schema and a potential problem with xsi:schemaLocation
85 values. Specifically these related to the XSD rule that a restricted type must use element types
86 that are equal or narrower than the same elements from the base type. The types in three of
87 these elements were incompatible with their base types.
- 88 · the Dictionary Entry Name for the CCT known as Text was actually Quantity. Type.

89 Also of significance is that OAG as part of their OAGIS 9.0 release in December 2003 made
90 similar changes to the original schemas. Some of the issues they encountered where:

- 91 · Derivation by Restriction Across Namespaces. According to the XML Schema specification,
92 derivation by restriction by restriction is not allowed if the element/type is based on another
93 type in a different namespace that has also been derived by restriction
94 (<http://www.w3c.org/TR/xmlschema-1/#coss-ct>). An example of this was in the IndicatorType
95 definition.
- 96 · OAGIS 9.0 chose to use cct:DecimalContent for all decimal CCTs (Amount, Measure,
97 Quantity). This restricts totalDigits to 28 and fractionDigits to 6
- 98 · In OAGIS 9.0, some supplementary components such as currencyID and unitCode are
99 required in the Core Components Schema. In UBL 1.0-beta, these are optional and made
100 mandatory when used as Representation Terms.
- 101 · UBL generally uses derivation by restriction whereas OAGIS uses derivation by extension.

102 Garret Minakawa made us aware of these changes in December 2003.

103 4 Schemas developed for UBL 1.0

104 As part of the UBL 1.0 development we identified other issues with the core component type
105 schemas that meant further changes. These were:

106 · **Formal Attribute Naming of Supplementary Components**

107 In November 2003, UBL established a Code List subcommittee to investigate ways to represent
108 code lists and their values. Naturally this group focused on the Core Component Type known as
109 Code. Type and its supplementary components as the data structure on which the representation
110 had to work. This exposed some inconsistencies with the naming of the attributes used for
111 supplementary components.

112 Up to and including UBL 1.0-Beta there was no consistent rule to naming attributes. In other
113 words, we had no NDR rule for naming supplementary components (and still don't). We have
114 carried through the original names suggested by SAP and Gunther Stuhec in their first
115 implementation.

116 For example, the Supplementary Component "Amount Currency. Identifier" was given the name
117 "currencyID". But, the Supplementary Component "Amount Currency. Code List Version.
118 Identifier" was given the name "codeListVersionID". There is no way to formally derive these
119 names from the Dictionary Entry Name or its ISO 11179 rules.

120 What is equally significant is that the names used are ambiguous. For example, the attribute
121 called "codeListVersionID" within an Amount is actually "Amount Currency. Code List Version.
122 Identifier". That is, it is not the code list version of the amount, but of it's currency.

123 The conclusion was that we should use the same rule for Supplementary Components that we
124 use for giving UBL names for Business Information Entities. That is using the
125 ObjectClass+PropertyTerm+RepresentationTerm rule based on the ISO 11179 and UBL NDR
126 naming rules.

127 The Core Component Technical Specification clearly defines the Dictionary Entry Names for each
128 supplementary components. From these we can derive the Object Class, Property Terms and
129 Representation terms involved. By applying the same formula used for other UBL Names we can
130 construct formal names for these attributes.

131 This approach is encouraged by the fact we are now using data models of core component types
132 so these rules are implemented by default when we generate the schemas, because they use the
133 same logic for naming components.

134 As an example, using the Supplementary Component "Amount Currency. Code List Version.
135 Identifier" is given the name "amountCurrencyCodeListVersionID".

136 It should be noted that we need to include the name of the Object Class when the Object Class of
137 the supplementary component is not the same as the Core Component Type itself. A clearer
138 example is with the Code core component type. It has two supplementary components, "Code
139 List. Uniform Resource. Identifier" and "Code List Scheme. Uniform Resource. Identifier". The
140 attribute name for both cannot be "uniformResourceIdentifier", so we need the Object Class of
141 "Code List" or "Code List Scheme" to keep the meaning clear.

142 The exercise of reverse-engineering the Core component types exposed the true structure of the
143 underlying data model and its supplementary components. We now have rules and formalism in
144 how the schemas can be derived from the models using the correct logical components as per the
145 Core Component Technical Specification.

146 Appendix A shows the model behind the core component types and their supplementary
147 components.

148 · **Codes and Identifiers not using xsd:normalizedString**

149 It was agreed by the UBL plenary that we should use xsd:normalizedString not xsd:token for
150 codes and identifiers.

151 · **Schema Modules renamed**

152 It was agreed by the UBL plenary that we should call the Representation Term schema the
153 Unspecialized Data Types schema and the original Data Types schema would be renamed as the
154 Specialized Data Types schema.

155 **5 Alignment with OAGIS**

156 It has been a long standing principle that UBL and OAG would try to align their implementations
157 of schemas for Core Component Types and Data Types. The intention is that this will provide
158 input into the work of a mutually agreed upon standards organization such as UN/CEFACT ATG2
159 or ISO.

160 In October 2003 both groups started with a common initial set of schemas. Since that time these
161 have evolved separately to accommodate design and implementation issues both within OAGIS
162 9.0 and UBL 1.0.

163 To assist ATG2 in finalizing its recommendation , both UBL and OAG have assessed the impact
164 of re-aligning their schema modules.

165 To this end, Garrett Minakawa (representing OAG) and Tim McGrath (representing UBL) have
166 reviewed the current OAGIS 9.0 and UBL 1.0 schemas.

167 They identified five areas of misalignment and proposed the following immediate course of action
168 to align these schemas.

169 **1. Naming of Supplementary Components as attributes.**

170 .- Analysis

171 UBL have adopted a naming convention for Supplementary Components based
172 on the ObjectClass + PropertyTerm + RepresentationTerm rule that applies
173 to BIEs.

174 OAG have informal naming rules inherited from the initial schemas.

175 .- Proposal

176 OAG consider adopting the same naming rules as UBL.

177 **2. Use of XSD normalizedString for code, identifier and text components.**

178 .- Analysis

179 OAG use the built-in XSD type,"token", for all code, identifier and text components
180 (where there is no specific built-in type, such as "language").

181 UBL uses the built-in XSD type,"normalizedString", for all code and identifier components
182 and the built-in XSD type,"string", for all text components (where there is no specific built-
183 in type, such as "language").

184 .- Proposal

185 OAG consider the built-in XSD type,"normalizedString", for all code, identifier and text
186 components (where there is no specific built-in type, such as "language").

187 UBL consider the built-in XSD type,"normalizedString", for all text components (where
188 there is no specific built-in type, such as "language").

189 **3. Use of XSD built-in datatypes requiring the "format" Supplementary Component**

190 These are: Date Time, Indicator and Numeric types

191 · Analysis

192 OAG explicitly define an attribute for "format" in the Core Component Type schema. This
193 is then restricted(prohibited) in the data type schema.

194 UBL do not define an attribute for "format" in the Core Component Type schema. This
195 follows UBL Naming and Design rule [STD1]:

196 *"For every ccts:CCT whose supplementary components map directly onto the properties
197 of a built-in xsd:datatype, the ccts:CCT MUST be defined as a named xsd:simpleType in
198 the ccts:CCT schema module."*

199 · Proposal

200 UBL consider relaxing NDR rule STD1 to allow adoption of the OAG approach.

201 **4. Restrictions on Binary Object for Graphic, Picture, Sound and Video**

202 **data type.**

203 · Analysis

204 OAG define different attributes for use in data types derived from Binary Object (Graphic,
205 Picture, Sound and Video). For example, in OAG a Graphic type has
206 characterSetCode,encodingCode,URI and filename whereas in UBL, a Graphic type has
207 only mimeTypeCode. (NB this is actually a UBL modeling error, it was supposed to have all
208 Supplementary Components except the mimeTypeCode).

209 · Proposal

210 UBL consider adopting OAG restrictions for Graphic, Picture, Sound and Video data type.

211 **5. Patterns for Indicator data type.**

212 · Analysis

213 OAG define a pattern of "true" or "false" for their Indicator data type. UBL has no pattern.

214 · Proposal

215 UBL consider adopting OAG pattern of "true" and "false" for the Indicator data type.

216 **6 Open Items for Core Component Type Schemas**

217 The experience of UBL and OAG has identified the need for additional Naming and Design rules
218 when realizing Core Component Types in XSD. These are items which ATG2 could arbitrate on.
219 They are:

220 · **Namespaces.**

221 OAGIS and UBL use different notation and naming in their namespace declarations. This should
222 not be a major issue since it is expected that OAGIS and UBL will eventually use the same set of
223 common core component schema files once they are officially approved and hosted by a mutually
224 agreed upon international standards organization such as UN/CEFACT ATG2 or ISO.

225 Once this occurs, OAGIS and UBL will simply reference the namespace names adopted by the
226 mutually agreed upon standards organization. As long as the content of the common core
227 component schema files remain unchanged, there should be no visible impact to end users.

228 · **Annotations.**

229 OAGIS and UBL have different documentation/annotation standards but again, this is not
230 expected to be an issue once the common core component schema files are implemented by a
231 mutually agreed upon international standards organization.

232 · **XML Schema Namespace Prefix.**

233 OAGIS uses “xs:” as the namespace prefix for “http://www.w3.org/2001/XMLSchema”. UBL uses
234 the prefix “xsd:”. As with namespaces and annotations, this is not expected to be an issue once
235 the common core component schema files are implemented by a mutually agreed upon
236 international standards organization.

237 · **complexType Naming Convention of Representation Terms.**

238 UBL has appended the term “Type” to the name of the type used for its representation terms (e.g.
239 “AmountType” vs. “Amount”). OAG has not done this.

240 · **Name of Representation Terms schema**

241 UBL uses “UnspecializedDataType” and OAG uses “RepresentationTerm”

242 · **Abbreviation for Identifier (ID vs. Id)**

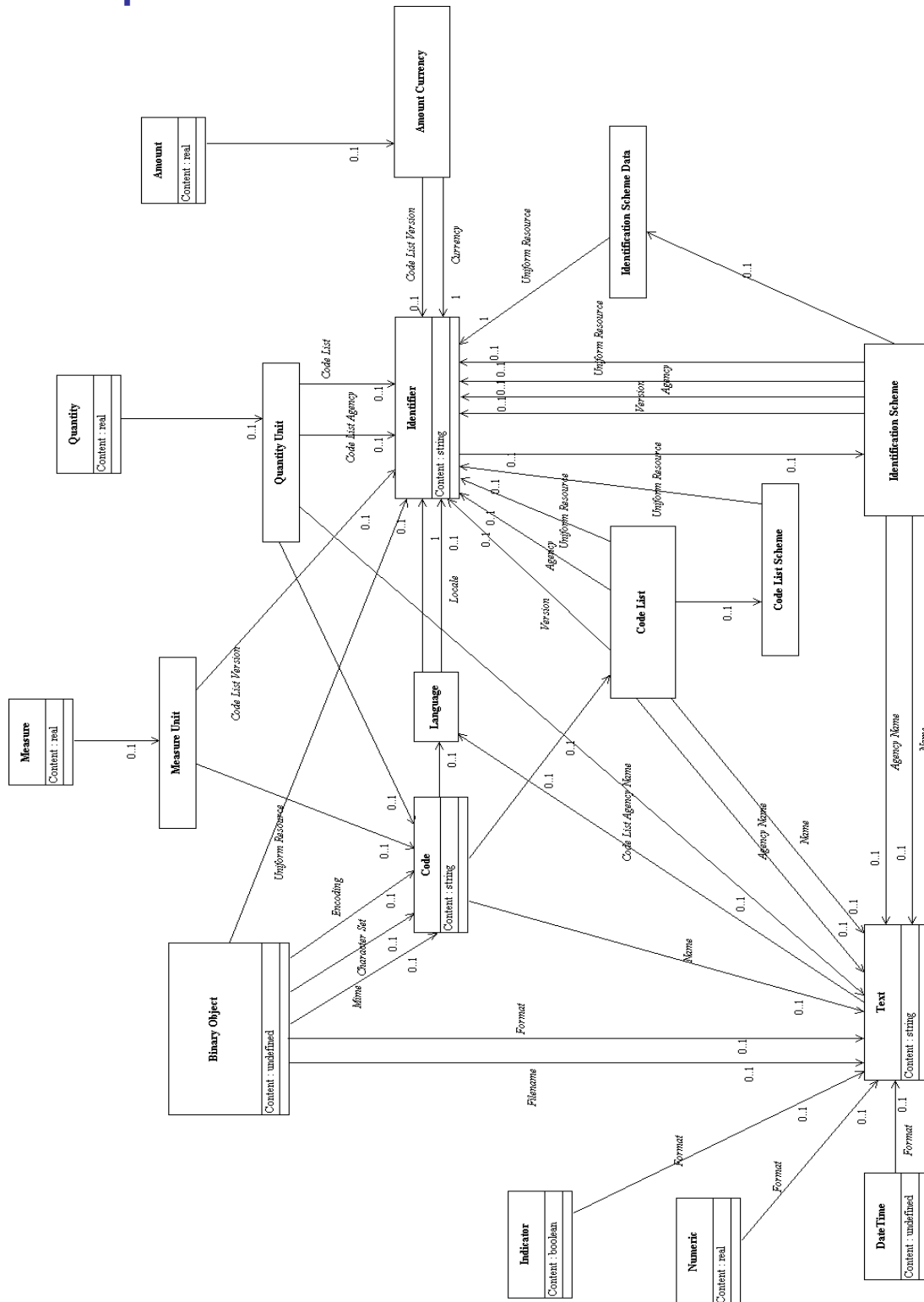
243 UBL uses “ID” and OAG uses “Id”.

244 · **Representing prohibited attributes**

245 There is no consistent method for representing prohibited attributes (and attributes with no
246 changes from the base type) when using derivation by restriction.

247
248

Appendix A. A class diagram of the Core Component data model



Appendix B.Notes

250 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
251 that might be claimed to pertain to the implementation or use of the technology described in this
252 document or the extent to which any license under such rights might or might not be available;
253 neither does it represent that it has made any effort to identify any such rights. Information on
254 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
255 website. Copies of claims of rights made available for publication and any assurances of licenses
256 to be made available, or the result of an attempt made to obtain a general license or permission
257 for the use of such proprietary rights by implementers or users of this specification, can be
258 obtained from the OASIS Executive Director.

259 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
260 applications, or other proprietary rights which may cover technology that may be required to
261 implement this specification. Please address the information to the OASIS Executive Director.

262 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS]
263 2004. All Rights Reserved.

264 This document and translations of it may be copied and furnished to others, and derivative works
265 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
266 published and distributed, in whole or in part, without restriction of any kind, provided that the
267 above copyright notice and this paragraph are included on all such copies and derivative works.
268 However, this document itself does not be modified in any way, such as by removing the copyright
269 notice or references to OASIS, except as needed for the purpose of developing OASIS
270 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
271 Property Rights document must be followed, or as required to translate it into languages other
272 than English.

273 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
274 successors or assigns.

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