Briefing Paper: UBL's current status with respect to schemas for implementing Core Component Types.

June 10, 2004

Document identifier:

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

Location:

Author:

Tim McGrath <tmcgrath@portcomm.com.au>

Abstract:

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

Status:

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

Table of Contents

1 Introduction

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

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

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

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

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

2 Background

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

In March 2002 UBL produced its first realization of these types in XML Schema, as part of the 0p64 review package. There was only one schema module and all Core Component Types, Supplementary Components and their secondary representation terms where defined in it. This was based on the CCTS version 1.7 and as far as we were aware was the first XSD implementation of these structures. Because the schema was automatically incorporated as part of the schema generation Perl script it was largely invisible to UBL developers. SAP (through Gunther Stuhec) did all of the maintenance on this.

This schema was updated to CCTS version 1.8 for the 0p65 review in August 2002.

By the time we got to UBL 0p70 in January 2003 to reflect CCTS 1.90, the schema now included an attempt to identify Secondary Representation Terms separately from Core Component Types. This release also saw the introduction of a Core Components Parameters schema to define annotations used in all other schemas. Once again, this remained largely below the horizon for UBL developers as it was built into the schema generation process.

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

UBL 0p81[1] in August 2003 saw the Core Component Type and Core Component Parameter schema modules transfer authorship to Chin Chee-Kai for use with the UBLish schema generator.

1 UBL 0p80 in June 2003 was only a model release and so we did no work on the schemas.

3 Schemas developed jointly by UBL,OAG and ATG2

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

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

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

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

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

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

4 Schemas developed for UBL 1.0

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

- **Formal Attribute Naming of Supplementary Components**

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

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

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

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

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

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

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

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

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

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

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

- **Codes and Identifiers not using xsd:normalizedString**

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

- **Schema Modules renamed**

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

5 Alignment with OAGIS

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

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

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

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

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

1. Naming of Supplementary Components as attributes.

- Analysis

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

  OAG have informal naming rules inherited from the initial schemas.

- Proposal

  OAG consider adopting the same naming rules as UBL.

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

- Analysis

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

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

- Proposal

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

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

3. Use of XSD built-in dataypes requiring the “format” Supplementary Component

These are: Date Time, Indicator and Numeric types

· Analysis

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

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

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

· Proposal

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

4. Restrictions on Binary Object for Graphic, Picture, Sound and Video data type.

· Analysis

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

· Proposal

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

5. Patterns for Indicator data type.

· Analysis

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

· Proposal

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

6 Open Items for Core Component Type Schemas

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

- **Namespaces**.

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

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

- **Annotations.**

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

- **XML Schema Namespace Prefix.**

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

- **complexType Naming Convention of Representation Terms.**

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

- **Name of Representation Terms schema**

UBL uses “UnspecializedDataType” and OAG uses “RepresentationTerm”

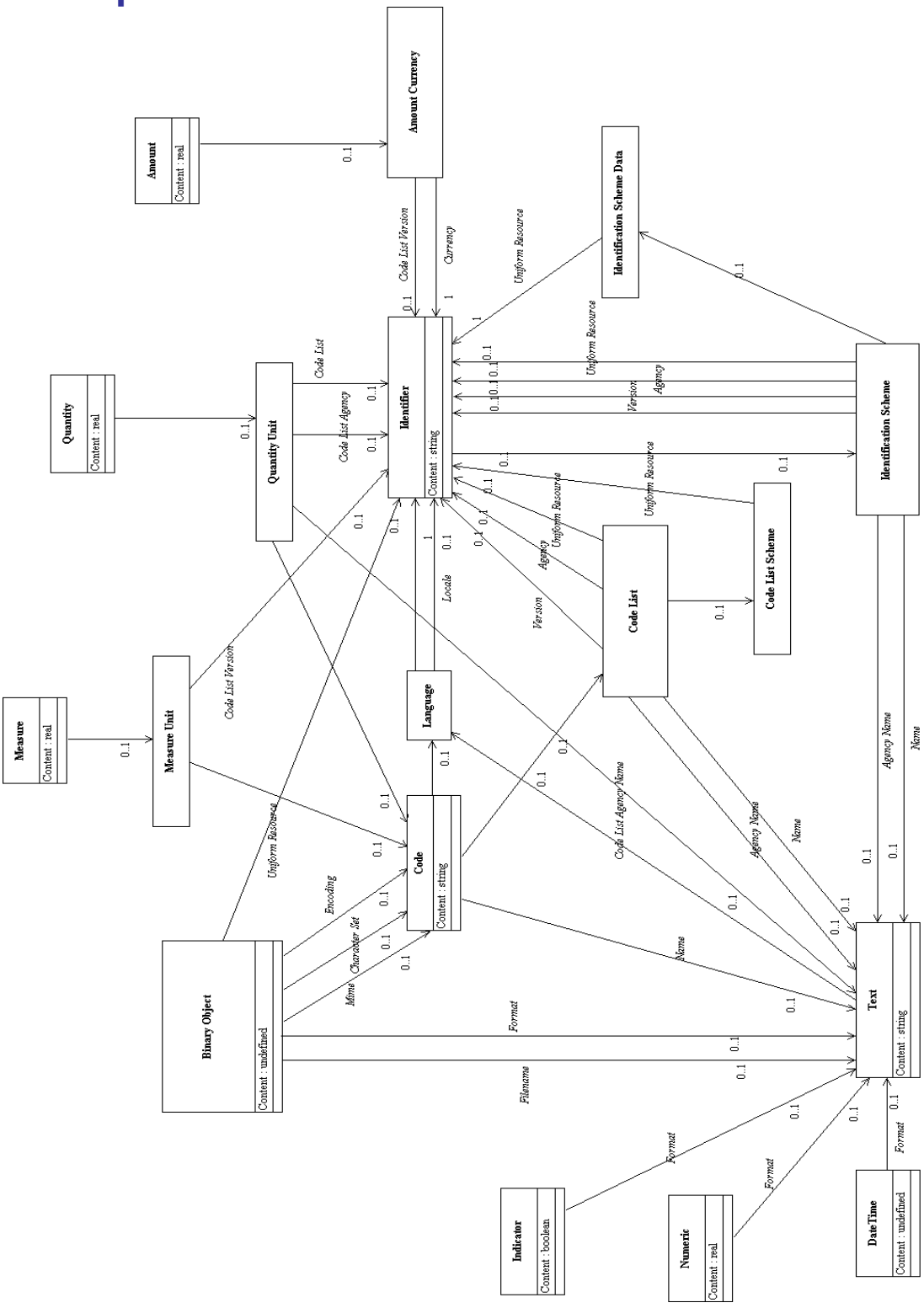
- **Abbreviation for Identifier (ID vs. Id)**

UBL uses “ID” and OAG uses “Id”.

- **Representing prohibited attributes**

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

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



Appendix B.Notes