Feedback from OASIS UBL TC to Draft Core Components Specification 1.8

document id

Version	0.1	editor	Bill Burcham
	April 10, 2002		Sterling Commerce

Much of the contention over element naming in UBL stems from the imprecise treatment of "properties" in the UN Core Components Technical Specification [CC-UN]. While that specification *does* talk extensively about "property terms" – which are part of a "dictionary entry name" for a "data element" (a la [NAMING-ISO]), we are left to *infer* the existence and makeup of a "first class" property concept.

The term "property" is used often in that specification[1], but it is never formally defined. Additionally, the term "child field" is used in some of the examples in that specification. That term is used synonymously to "property", and is also left undefined. Further, it never appears in any of the conceptual diagrams.

We are trying to give "property terms" to things. What things are we trying to give them to? Well [CC-UN] doesn't tell us.

We propose:

P0: The CC model must include the concept of *property*. Property is the model element named by a *property term* in the same way as a *BIE* or a *CC* is the model element named by an object class (name).

What is property's relationship to the other elements of the CC meta-model?

P1: A property relates an Aggregate Core Component to the Core Components it *contains*.

P2: The generalization-specialization relationship of CCT to RT must be reversed.
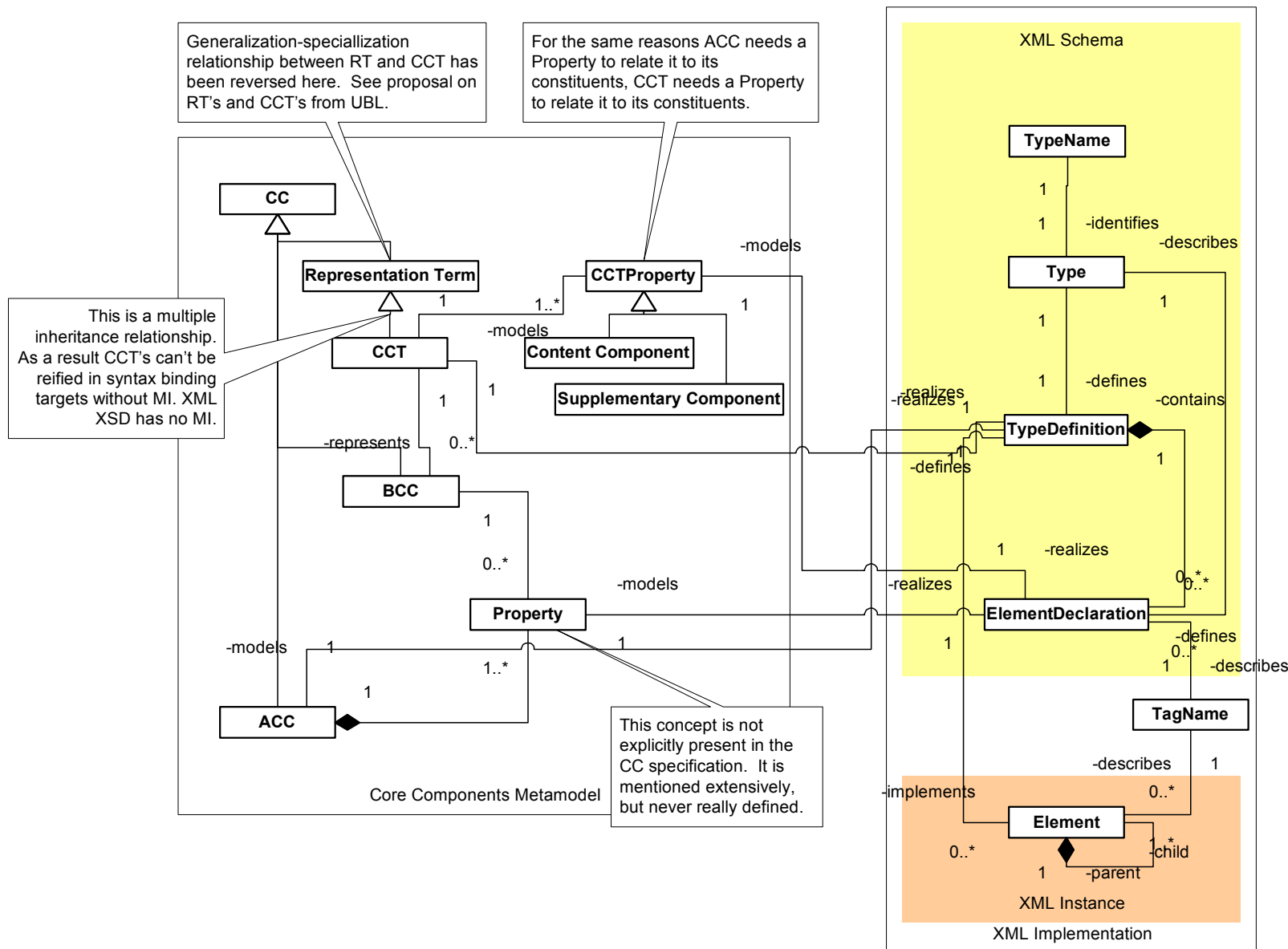
<Arofan's proposal goes here>

When I actually constructed the model (diagram) it became clear that for the same reasons a property is needed to relate an ACC to the BCC's it contains, a property is needed to relate a CCT to the Components it contains.

P3: A "CCTProperty" relates a CCT to the (Content and Supplimentary) Components it contains.

As a result of P0-P3, Figure 6.1 *Core Components Metamodel* should now be as shown in the "Core Components Metamodel" box in this diagram:

---

[1] Section 5.6 lines 838-851; section 5.6.2 lines 892-914

Generalization-speciallization relationship between RT and CCT has been reversed here. See proposal on RT's and CCT's from UBL.

For the same reasons ACC needs a Property to relate it to its constituents, CCT needs a Property to relate it to its constituents.

XML Schema

TypeName

1

1 -identifies

-describes

Type

1

CC

Representation Term

1

1..*

CCTProperty

1

-models

This is a multiple inheritance relationship. As a result CCT's can't be reified in syntax binding targets without MI. XML XSD has no MI.

CCT

-models

Content Component

Supplementary Component

1

1

-realizes
-realizes 1

-defines

-contains

TypeDefinition

1

-defines

1

-represents

0..*

BCC

1

0..*

-realizes

1 -realizes

-models

1

-realizes

ElementDeclaration

0..*
0..*

-defines
0..*

Property

1

TagName

1

-models

1

1..*

ACC

1

1

-describes 1

This concept is not explicitly present in the CC specification. It is mentioned extensively, but never really defined.

Core Components Metamodel

-implements

0..*

Element

-describes

0..*

0..*

-child

1

-parent

XML Instance

XML Implementation

The preceding diagram shows how the proposed Core Components metamodel (with properties modeled) is syntax bound to XSD by UBL.

The syntax binding process to XSD involves creating XSD complex types for ACC's and CCT's. These complex types consist of (local) element declarations – one for each property of the source ACC/CCT. The element's "tag name" is identical to the name of the source property (Property/CCTProperty).

Further, once we identify and describe these properties, what shall we call them? Could a set of rules around role definition satisfy our need to capture recurring component usage patterns (and name them)? Perhaps the central tenet would be:

P4: Role-based property naming: a property's name (property term in the dictionary entry name) should reflect the role played by that property's content *relative to* the CC in which that property is declared.

References

CC-UN	*UN/CEFACT Draft Core Components Specification, Part 1*, 8 February, 2002, version 1.8	
NAMING-ISO	*ISO/IEC 11179,* Final committee draft, Parts 1-6.	