WCO Data Model
XML Guidelines
2nd Edition

Amendment History				
Change Number	Description	Page(s) affected	Version	Effective Date
1	Release of 1st Edition of WCO Data Model XML Guidelines together with WCO Data Model Version 3	-	1st Edition	December 2009
2	Merge WCO XML Schema Customization Guide into this Guidelines.	18-29	2nd Edition	September 2012
3	Add usage of “Date Time” data type in the ”Basic Rules” section.	3		
4	Add rules in the “Basic Rules” section on maintaining the message structure of a WCO XML Message.	3-4		
5	Add the “Extensions” section.	7-10		
6	Add the “Namespaces” section.	13-14		
7	Add “Binary Object” data type, remove “Date” data type and add the “formatCode” XML attribute for “Date Time” data type in the Annex.	30-33		

TABLE OF CONTENTS

List of Tables

List of Figures

SECTION I BACKGROUND AND GENERAL PRINCIPLES

1 Purpose

These Guidelines have been established to assist developers with a basic set of rules to establish WCO XML messages. The Guidelines include definitions of what the final XML message looks like. The Guidelines do not stipulate the use of any specific XML specification such as DTD or XML Schema, these are technical methods to describe the structure and which method used (if any), is left at the discretion of each implementer. Nevertheless, the Guidelines contain a package of XML artefacts[1] which are intended to illustrate and facilitate the identification of information content.

2 Background

The Extensible Markup Language (XML) offers a way to define formats for exchanging business data and enables the development of open and flexible applications for automating electronic transactions over public networks. The function of the XML message is to carry and convey information.

XML based applications have flourished based on non-standard semantics and structures which limit the interoperability between and within different business domains. To overcome these limitations, a consistent framework[2] for the development of XML messages is established by referencing the open specifications developed through public processes such as the Electronic Business Extensible Markup Language (UN/CEFACT/ebXML) Core Components Technical Specification V2.01 (CCTS V2.01), ISO 11179 and UN/CEFACT XML Naming and Design Rules V2.0. While it is recognized that these are evolving standards and frameworks, it is however necessary to underscore the need for the Guidelines to produce specifications that promote cross border and cross domain interoperability, which is crucial for any e-business standard.

[1] The XML artefacts are not intended for use in validating the information content. Therefore these artefacts will not contain detailed restrictions placed on the data elements, e.g. lengths and code lists.

[2] The framework covers the processes of deriving WCO XML messages from the WCO Data Model.

3 Basic Rules

- The WCO Data Model is the only source for business rules, repetitions, formats, core data types, code sets, class definitions and message structure. The business rules set by the administration that is defining the subsets, will be the supplementary source.

- For all document assembly based on the WCO DM, the root class would become the root element for the corresponding XML representation.

- The XML document shall be supplied with a namespace which provides process information for XML parsers. An explanation on namespace assignment can be found under title 8.

- The class name is the same as the XML tag name to be noted in UpperCamelCase, e.g. TransportEquipment becomes <TransportEquipment>.

- The XML tag name for a class attribute is the concatenation of the Property Term and the Representation Term of the Dictionary Entry Name of the class attribute in UpperCamelCase with separators and spaces removed, e.g. the attribute with WCO Ref 029 "Border Transport Means. Discharge Completed. Date Time" is represented as <DischargeCompletedDateTime>. An explanation on Dictionary Entry Names of WCO data elements can be found under title 9. When forming the XML tag names, the following rules have to be observed:

- The Representation Term "Identifier" shall be abbreviated as "ID" in the XML tag. For example, the tag name for WCO Ref 165 "Transport Equipment. Seal. Identifier" is represented as <SealID>.

- If the word "Identification" is the final word of the Property Term and the Representation Term is "Identifier", then the word "Identification" shall be removed from the XML tag name. If the word "Indication" is the final word of the Property Term and the Representation Term is "Indicator", then the word "Indication" shall be removed from the XML tag name. For example, the tag name for WCO Ref R033 "Crew Member. Identification. Identifier" is represented as <ID>.

- If the Representation Term is "Text", "Text" shall be removed from the XML tag name. For example, the tag name for WCO Ref R011 "Carrier. Name. Text" is represented as <Name>.

- In the case that the class is a sub-class of a super-class, the "missing" attribute names should be taken from the super-class, e.g. class AdditionalDocument with WCO Ref D006, the attribute name comes from super-class Document, i.e. type which becomes <TypeCode>.

- The data types may be supplemented with metadata as specified in the UN/CEFACT Core Components Technical Specification V2.01. For example: WCO Ref 131, Total Gross Weight requires a value, say 250 and an optional Measurement Unit Code requires a value, say KGM for kg. The XML attribute names listed in the column "Attribute Name" in the Annex shall be used to represent the supplementary metadata. In this example, the data item Total Gross Weight is represented by <TotalGrossMassMeasure unitCode="KGM">250</TotalGrossMassMeasure>. In this Annex, all attributes that can be used have been mentioned and the choice is left to the implementer and the needs of the users of the XML specifications.

- Regarding “Date Time” data type, the XML attribute “formatCode” is used to specify the format of the date/time content. WCO XML and EDIFACT messages will both use codes such as 102, 304 and 602 to represent the date time format CCYYMMDD, CCYYMMDDHHMMSSZZZ and CCYY respectively. For example, the Arrival Date of a Border Transport Means can be represented as <ArrivalDateTime formatCode="102">2007-10-15</ArrivalDateTime>.

- The XML element for a class shall contain XML elements for the associated classes only which are directly associated to that class in order to maintain the message structure presented in the class diagrams. In other words, the XML element for any class which is not directly associated to a class shall not be presented under the XML element for that class. e.g. <Commodity> shall not be presented under <Declaration> in the CRI message as Commodity class is not directly associated to Declaration class in the CRI class diagram. To include <Commodity> in the CRI message, XML elements for the intermediate classes, i.e. <Consignment> and <ConsignmentItem>, shall be added as shown below:

```
<Declaration xmlns="urn:wco:datamodel:WCO:CRI:1">
        <Consignment>
                <ConsignmentItem>
                        <Commodity>
                                …
                        </Commodity>
                </ConsignmentItem>
        </Consignment>
</Declaration>
```

- The XML elements for a class shall contain XML elements for the class attributes of that class. In situation where the occurrence of the XML element for a class attribute is optional (i.e. minimum occurrence is zero) and no data is applicable, the XML element for that class attribute shall be omitted. In the following example, optional XML elements <CountrySubDivisionID> and <CountrySubDivisionName> are omitted because both XML elements do not store any data. The XML elements for other class attributes are presented under <Address>.

As a result,

```
<Address>
        <CityName>Brussels</CityName>
        <CountryCode>BE</CountryCode>
        <CountrySubDivisionID></CountrySubDivisionID>
        <CountrySubDivisionName></CountrySubDivisionName>
        <Line>Rue du Marché, 30</Line>
        <PostcodeID>B-1210</PostcodeID>
</Address>
```

becomes

```
<Address>
        <CityName>Brussels</CityName>
        <CountryCode>BE</CountryCode>
        <Line>Rue du Marché, 30</Line>
        <PostcodeID>B-1210</PostcodeID>
</Address>
```

- The XML elements for the associated classes to a class shall be presented under the XML element for that class. In situation where the occurrence of the XML element for an associated class is optional (i.e. minimum occurrence is zero) and there is no XML element presented under the XML element for that associated class, the XML element for that associated class shall be omitted. In the following example, <Pointer> which does not contain any XML element shall be omitted. <Pointer> contains no XML element because the XML elements, i.e. <DocumentSectionCode>, <SequenceNumeric> and <TagID>, which do not store data are omitted.

Consequently,

```
<Amendment>
        <ChangeReasonCode>112</ChangeReasonCode>
        <Pointer>
                <DocumentSectionCode></DocumentSectionCode>
                <SequenceNumeric></SequenceNumeric>
                <TagID></TagID>
        </Pointer>
</Amendment>
```

becomes

```
<Amendment>
        <ChangeReasonCode>112</ChangeReasonCode>
</Amendment>
```

A sample segment for a CRI message with the basic rules applied is shown below:

```
<Declaration xmlns="urn:wco:datamodel:WCO:CRI:1">
        <ID>CRI0745557</ID>
        <IssueDateTime formatCode=”304”>2007-10-11T09:00:00+08:00<IssueDateTime>
        <TotalGrossMassMeasure unitCode=”KGM”>100000</TotalGrossMassMeasure>
        <TypeCode>422</TypeCode>
        <BorderTransportMeans>
                <ArrivalDateTime formatCode=”102”>2007-10-15</ArrivalDateTime>
                <BuiltYearDateTime formatCode=”602”>1991</BuiltYearDateTime>
                <CargoFacilityLocationName>HKHKC</CargoFacilityLocationName>
                <DepartureDateTime>2007-10-11T13:00:00+08:00</DepartureDateTime>
                <FirstArrivalLocationID>HKHKC</FirstArrivalLocationID>
                <ID>9365790</ ID>
                <JourneyID>4550</JourneyID>
                <Name>ABC GENERAL</Name>
                <RegistrationNationalityID>KR</RegistrationNationalityID>
                <ScheduledConveyanceID>8645663</ScheduledConveyanceID>
                <StayID>07353415</StayID>
                <TypeCode>1501</TypeCode>
                <CrewMember>
                        <ID>K04587446</ID>
                </CrewMember>
                <CrewMember>
                        <ID>L56687427</ID>
                </CrewMember>
                <Master>
                        <BirthDateTime formatCode=”102”>1961-12-01</BirthDateTime>
                        <ID>H00456688999</ID>
                        <Name>PETER PAN</Name>
                </Master>
        </BorderTransportMeans>
        <Carrier>
                <ID>554688-001</ID>
                <Name>ABC LTD</Name>
        </Carrier>
        …
</Declaration>
```

4 Mini Message

Apart from the standard procedure messages for the WCO governmental procedures, individual cross-border regulatory agency (CBRA) may impose local requirements over the standard procedure messages. In such cases, not all WCO data elements and/or classes in the standard procedure message are required and a subset or a trimmed-down version of the standard procedure message, i.e. a mini message, would be needed.

Mini messages are also needed for meeting the requirements of the WCO SAFE Framework of Standards (FoS). Under SAFE FoS, CBRAs should require the exporter/importer/carrier or his/her agent to submit only particular sets of data elements[3] as advance electronic declaration. Such advance electronic declarations include advance export goods declaration (AEX), advance import goods declaration (AIM), advance export cargo declaration (ACRE) and advance import cargo declaration (ACRI) which should be subsets based on EX12, IM12, CRE and CRI respectively according to the ISCM Guidelines[4].

The following basic steps shall be followed when deriving a mini message:

- Choose the appropriate standard procedure message to be trimmed down. For instance, select EX12 as a base message for the mini message AEX.
- Retain the data elements and classes of the standard procedure message that are required according to the business scenario.
- Purge those data elements and classes of the standard procedure message that are not required.

[3] Refer to pages I/17 - I/21 of the “ WCO Framework of Standards to Secure and Facilitate Global Trade” published by WCO in June 2005.

[4] Customs Guidelines on Integrated Supply Chain Management published by WCO in June 2004.

5 Extensions

In international data exchange the use of extensions should be avoided. On national level it may not be possible to avoid the use of extensions because not all the national requirements can be added to the WCO Data model.

5.1 Principles

In case extensions are used the main things to note are:

1. Don’t use extensions for things supported by the language itself
2. Make sure extensions are optional outside its intended domain
3. Put extension elements in a different namespace
4. Encourage the use of schemas to define extension elements
5. Consider the impact of sending an “extended” message to a system that does not understand the extension
6. Build systems so that they ignore extensions they do not understand
7. If possible, report that an extension has been ignored (as a warning, not an error)

5.2 Mechanism

An optional XML schema called Extended Data Set is proposed to be introduced.

This schema will include elements that are not in the WCO Data Model, especially those that are not foreseen as data elements that would be commonly required by Members under a Cross-border regulatory requirement.

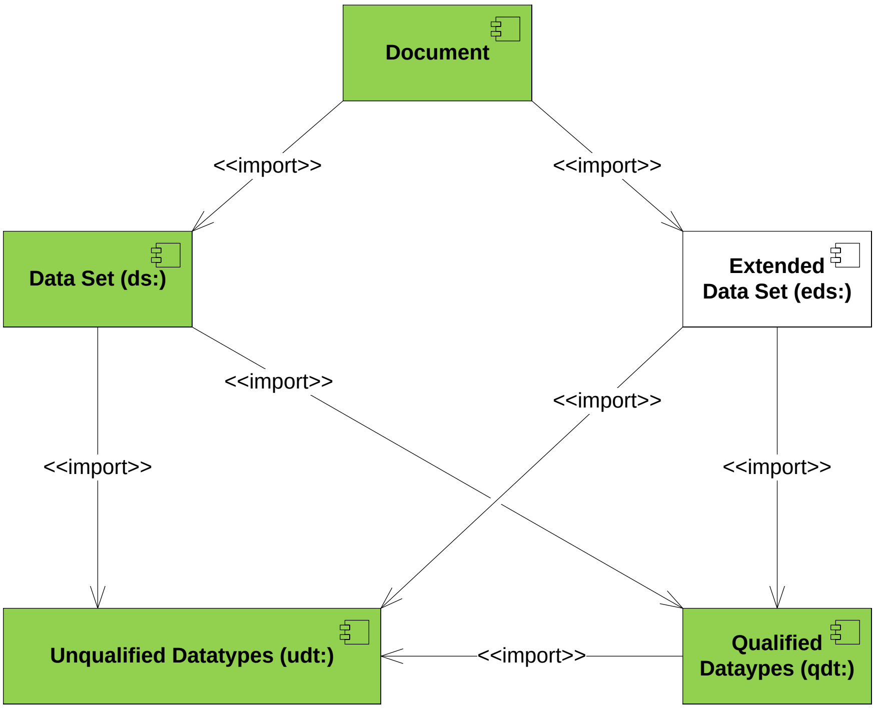


Figure 1: Schema design to be followed in the Extension mechanism

The colored components already exist in the data model schema specifications. The Extended Data Set to the right side is the extension component proposed. Users of Extended Data Set may harm the prospects of full interoperability between the family of users of WCO Data Model. Therefore, the use of extensions is therefore not encouraged.

5.3 Governing Rules

1. Extended Data Set should be used with great care and should never be used for data that is properly conveyed in standard WCO elements allowed elsewhere in the document.
2. Extended Data Set should be used only as a last resort for data that cannot be accommodated by the present structures of the WCO Data Model.
3. Use of Extended Data Set will require a special agreement between the users of the specific customization set, requiring specific agreement between relevant parties.
4. The Extended Data Set will be governed by publication and maintenance procedures outside the scope of WCO Data Model.

5.4 Example Technical Solution

Technical solutions for extension and code list extension are described below.

Technical solution for XML extension is based on the design principles. There are four extension examples regarding data elements and classes:

1. Include a localized data element e.g. "Consignment. Consolidation. Indicator" is added to class "Consignment"
2. Include a localized class and its related data elements e.g. class "Tax Payer" is added under class "Declaration"
3. Include a WCO data element which is not covered in a particular WCO procedure document e.g. data element ID 156 "Border Transport Means. Departure. Date Time" is added to EX1
4. Include a WCO class which is not covered in a particular WCO procedure document. E.g. class "FreeTradeZone" is added to EX1 under class "GoodsShipment"

The following abstracted example illustrated how these extension examples are represented in XML. Namespace prefix "eds" is used to identify the extended data elements and classes.

```
<Declaration xmlns="urn:wco:datamodel:HK:EX1:1" xmlns:eds="urn:wco:datamodel:HK:EDS:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:wco:datamodel:HK:EX1:1
..\maindoc\HK_EX1_1p0.xsd">
        <ID>0A93ID35W0BRXS</ID>
        ...
        <BorderTransportMeans>
<!-- ================================================================ -->
<!-- ===== Example 3: Include a WCO data element which is not covered in a particular WCO
procedure document ===== -->
<!-- ================================================================ -->
                <eds:DepartureDateTime formatCode="102" >2001-08-23</eds:DepartureDateTime>
                ...
        </BorderTransportMeans>
        ...
        <GoodsShipment>
                <SequenceNumeric>1</SequenceNumeric>
                <Consignment>
                        <SequenceNumeric>1</SequenceNumeric>
<!-- ================================================================ -->
<!-- ===== Example 1: Include a localized data element ===== -->
<!-- ================================================================ -->
                        <eds:ConsolidationIndicator>true</eds:ConsolidationIndicator>
                </Consignment>
                ...
<!-- ================================================================ -->
<!-- ===== Example 4: Include a WCO class which is not covered in a particular WCO procedure
document ===== -->
<!-- ================================================================ -->
                <eds:FreeTradeZone>
                        <eds:Name>Huangpu Free Trade Zone</eds:Name>
                </eds:FreeTradeZone>
        </GoodsShipment>
        ...
<!-- ================================================================ -->
```

```
<!-- ===== Example 2: Include a localized class and its related data elements ===== -->
<!-- ============================================================== -->
        <eds:TaxPayer>
                <eds:Name>XYZ Co Ltd</eds:Name>
                <eds:Contact>
                        <eds:Communication>
                                <eds:ID>2961-5000</eds:ID>
                                <eds:TypeID>TE</eds:TypeID>
                        </eds:Communication>
                </eds:Contact>
        </eds:TaxPayer>
</Declaration>
```

Regarding technical solution for code list extension, the same design principle is applied to Code data type. An XML attribute "listName" which represents supplementary component "Code. List Name. Text" is used to present the name of a list of codes. If the content of a code is an extended code value, a value such as "Extended Code List of UNTDED 1001" is assigned in the XML attribute. If the content is a standard code value (no matter standard code list or extended code list is used), the XML attribute is not shown.

For instance, "Additional Document. Type. Code" which is used to distinguish different types of additional documents adopts UNTDED 1001 code list. However, some documents such as Consignment Note are not codified in the UNTDED code list. As such, extended code value such as "CN" is needed as shown below:

```
<AdditionalDocument>
        <ID>11928890044</ID>
        <TypeCode listName="Extended TDED 1001 Code List">CN</TypeCode>
</AdditionalDocument>
```

For standard codes such as 741 which stands for Master Air Waybill in UNTDED code list, XML attribute is not shown.

```
<AdditionalDocument>
        <ID>A1087749844</ID>
        <TypeCode>741</TypeCode>
</AdditionalDocument>
```

By adopting such approach, sender can identify extended codes used and prepare a standard WCO XML message by removing XML tags with extended codes.

If sender and receiver agree to exchange extended codes, users can opt to use XML attribute "name" which represents supplementary component "Code. Name. Text" to present the textual equivalent of the code content. As such, the receiver can manipulate the extended codes as textual description is given. For example, receiver knows the code description of extended code "CN" is "Consignment Note" as shown below and can further process the content:

```
<AdditionalDocument>
        <ID>11928890044</ID>
        <TypeCode listName="Extended TDED 1001 Code List" name="Consignment
Note">CN</TypeCode>
</AdditionalDocument>
```

6 Document Metadata

Standard procedure messages and mini messages require metadata to differentiate themselves from each other as well as providing supplementary information regarding the messages. An XML envelope <DocumentMetadata> will be used to encapsulate the required metadata as well as the standard procedure message or mini message itself. Within the <DocumentMetadata> envelope, the metadata are represented by tag pairs which include:

- <WCODataModelVersion> : The WCO Data Model Version of the encapsulated WCO Document type or a WCO Document type from which the encapsulated mini-message is derived. e.g. 3.0. This tag pair is mandatory and will occur once.
- <WCODocumentName> : The abbreviated name of the WCO Document type that is encapsulated or from which the encapsulated mini-message is derived, e.g. IM1, CRE, etc. The full list of abbreviated name of the WCO Document type can be found in table 1 Document names (see title 8). This tag pair is mandatory and will occur once.
- <CountryCode> : The country, represented using ISO 3166-1 Country Code, that specified this customized version of mini-message. This tag pair is optional and will occur at most once.
- <AgencyName> : The name of the Government agency that specified this customized version of mini-message. This tag pair is optional and will occur at most once.
- <AgencyAssignedCountrySubDivisionID> : The country subdivision, as assigned by the corresponding Government agency, that this customized version of mini-message is applicable. This tag pair is optional and will occur at most once.
- <AgencyAssignedCustomizedDocumentName> : The abbreviated name, as assigned by the aforesaid Government agency, of this customized version of mini-message , e.g. HKAIM. This tag pair is optional and will occur at most once.
- <AgencyAssignedCustomizedDocumentVersion> : The version number, as assigned by the aforesaid Government agency, of this customized version of mini-message , e.g. "Revision 7.1b". This tag pair is optional and will occur at most once.

The <DocumentMetadata> envelope shall be supplied with a namespace which provides process information for XML parser. The namespace shall be urn:wco:datamodel:WCO:DM:1.

The following example shows the structure of the XML message after encapsulating the metadata using the <DocumentMetadata> envelope:

```
        <DocumentMetadata xmlns="urn:wco:datamodel:WCO:DM:1" >
                <WCODataModelVersion>3.0</WCODataModelVersion>
                <WCODocumentName>AIM</WCODocumentName>
                <CountryCode>HK</CountryCode>
                <AgencyName>Customs and Excise Department</AgencyName>
                <AgencyAssignedCustomizedDocumentName>HKAIM
</AgencyAssignedCustomizedDocumentName>
                <AgencyAssignedCustomizedDocumentVersion>Revision
7.1b</AgencyAssignedCustomizedDocumentVersion>
                <Declaration xmlns="urn:wco:datamodel:WCO:IM1:1">
                        ...
                </Declaration>
        </DocumentMetadata>
```

7 Digital Signature

In case authentication and non-repudiation for a WCO message is needed, the W3C enveloping XML signature[5] will be adopted. The WCO message including the metadata, as a data object, will be signed. The XML signature will have the following structure:

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
                ...
        </SignedInfo>
        <SignatureValue/>
        <KeyInfo>
                ...
        </KeyInfo>
        <Object>
                <DocumentMetadata xmlns="urn:wco:datamodel:WCO:DM:1" >
                        ...
                </DocumentMetadata>
        </Object>
</Signature>
```

The 5 outer tag pairs (i.e. <Signature>, <SignedInfo>, <SignatureValue>, <KeyInfo>, <Object>) will follow the syntax and semantics of the W3C enveloping signature.

[5] For details please refer to http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/

8 Namespaces

Namespace is used to differentiate concept within a domain, the universal name (combination of namespace and local name) of different concept should be different. If the concept is the same, the universal name and hence the namespace and the local name should be the same. For example, if Commodity in IM1 and CRE are the same, then the namespace of IM1 and CRE should be the same. If Commodity in IM1 and CRE are different, then the namespace of IM1 and CRE should be different.

Depending on how the concept is treated, i.e. if classes and data elements across documents are the same or not, there are basically three options when deciding how namespace is used in WCO documents. These options would be:

a) All documents in WCO DM use the same namespace
b) Same document family use the same namespace
c) Each document use its own namespace

In option C, classes and data elements in each document have different semantic meaning, e.g., Declaration class in IM1 and CRE are different. This option is proven to be easy to handle and is already widely used. This is why it has been decided to recommend the use of option C. Document names as well as respective abbreviations are listed in table 1.

Below explains the Namespace Assignment Convention for assigning namespace to WCO Data Model (WCO DM) Instance Document[6]. Recommendations of the UN/CEFACT XML Naming and Design Rules V2.0[7] were adopted as far as possible.

8.1 Namespace Assignment Convention

The namespace for an Instance Document is defined using the “xmlns” attribute in the root element”.

8.2 Use of Uniform Resource Name (URN) for Namespace

Namespaces are usually represented in the form of Uniform Resource Names (URNs).

The syntax of the URN that represents the Namespace for Instance Document is
urn:wco:datamodel:[customsAdministration]:[name]:[version]
where
[customsAdministration] = either the string “WCO” or the ISO 3166-1 Alpha-2 Country Code of individual Customs Administration;
[name] = the abbreviated name of the Instance Document in table 1 or the abbreviated name of the customized document assigned by individual Customs Administration;
[version] = major version number; sequentially assigned, starting with the number 1.

For example, the namespace for WCO IM1 Instance Document is

[6] A WCO DM Instance Document is the document, as defined in the WCO Data Model XML Guidelines, to be exchanged between different stakeholders.

[7] Available at: http://www.uncefactforum.org/ATG/atg_news_download.htm

urn:wco:datamodel:WCO:IM1:1.

The namespace for the localized Hong Kong IM1 Instance Document is **urn:wco:datamodel:HK:IM1:1**

Document Name	Abbreviated Name
for Data Modelling purposes	
WCO Overall Declaration	DEC
Response	RES
for Customs Procedures specified in Kyoto Convention	
Import declaration	IM
Export declaration	EX
Cargo Report Import	CRI
Cargo Report Export	CRE
Conveyance Report	CONV
Transit	TRT
for Customs Procedures specified in WCO SAFE FoS	
Advance Export Goods Declaration	AEX
Advance Import Goods Declaration	AIM
Advance Export Cargo Declaration	ACRE
Advance Import Cargo Declaration	ACRI

Table 1: Document names

9 Dictionary Entry Names (DENs) of WCO Data Elements

One or more Dictionary Entry Names (DENs) are assigned to every WCO data element to facilitate adoption of XML as exchange format of the WCO Data Model (WCO DM). In short, some components of each DEN will be transformed into the names of XML tags according to the recommendations of the UN/CEFACT XML Naming & Design Rules V2.0[8] as far as possible.

9.1 Dictionary Entry Name Structure

According to the ISO 11179-5:2005[9] Naming and Identification Principles, a DEN may consist of an object class term, a property term, a representation term, and qualifier term(s). The object class term represents an object, which is a set of ideas, abstractions or things in the real world. The property term represents a characteristic common to all members of an object. The representation term describes the form of representation of the characteristic being described, e.g. Name, Amount, Text, Number, etc. Lastly, qualifier term(s) may be attached to object class term, property term, and representation term if necessary to distinguish one data element from another.

Adhering to ISO 11179-5:2005, three DEN components have been assigned to each WCO data element. The WCO derived the DEN object class terms from the corresponding WCO DM object class names. Similarly, the DEN property terms are derived from the corresponding WCO DM object class attributes. And the representation terms are derived from the form of representation of the corresponding WCO data elements.

The WCO needs not and does not assign any qualifier terms because the name of any object class in WCO DM is already unique, and the name of any class attribute belonging to the same object class is also unique.

9.2 WCO DM DEN Naming Convention

Taken into consideration the recommendations about naming conventions stated in the ISO 11179-5:2005 and naming rule principles adopted by UN/CEFACT Core Components Technical Specification CCTS v2.01[10]), the naming convention as stipulated in sub-sections 3.1 through 3.6 has been formulated. The naming convention is developed for and applied to WCO DM version 3 and beyond.

[8] Available at: http://www.uncefactforum.org/ATG/atg_news_download.htm

[9] Available at: http://standards.iso.org/ittf/PubliclyAvailableStandards/c035347_ISO_IEC_11179-5_2005(E).zip

[10] Available at: http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf

9.2.1 Authority

The WCO is the authority that assigns names and enforces the naming convention.

9.2.2 Semantic Rules

(a) For each DEN, there will be one and only one object class term. The name of the corresponding object class in the WCO DM shall form the object class term in the DEN.

(b) For each DEN, there will be one and only one property term. The name of the corresponding class attribute of the corresponding object class in the WCO DM shall form the property term in the DEN.

(c) For each DEN, there will be one and only one representation term. The representation term shall describe the form of representation as stated in the definition of the corresponding WCO data element.

9.2.3 Syntactic Rules

(a) The object class term shall occupy the first (leftmost) position in the DEN.

(b) The property term shall occupy the next position in the DEN.

(c) The representation term shall occupy the last position in the DEN.

9.2.4 Lexical Rules

(a) The object class term, property term and representation term shall be separated by a dot and single space. For example, “Commodity. Brand Name. Text”.

(b) All the words in the names of the object class terms, property terms, and representation terms shall begin with capital letter. Name parts and words in multi-word terms shall be separated by spaces. For example, the WCO DM object class attribute “customsStatus” shall become the DEN property term “Customs Status”.

(c) Nouns shall be used in singular form only unless the concept itself is in plural form. For example, “Customs” is allowed whereas “Items” is not allowed. Verbs (if any) shall be in present tense.

(d) The allowable abbreviations in object class term, property term and representation term are ID, UCR, LPCO and ISPS.

9.2.5 Uniqueness Rule

The property terms of all DENs that share the same object class term shall be different. The uniqueness will also ensure that the names of XML tags derived are not violating the W3C XML syntax rules.

9.2.6 Assigned DENs

The assigned DENs are recorded under the following four new columns in the WCO data elements spreadsheet:

(a) WCO DM Dictionary Entry Name
(b) Object Class Term
(c) Property Term
(d) Representation Term

It is noted that WCO DM Dictionary Entry Name is a concatenation of the object class term, property term, and representation term separated by the symbol “.”.

SECTION II WCO XML SCHEMA CUSTOMIZATION

10 Characteristics of the Sample XML Schemas

Sample XML schemas for WCO Data Model are provided to CBRAs for reference in constructing their own schemas. These sample XML schemas are built using the Russian Doll model approach, which allows easy customization of the schemas through trimming off unnecessary class / class attributes to suit the specific requirements of individual CBRAs.

The sample XML schemas have made use of UN/CEFACT Unqualified Data Type XML Schema to represent the Core Component Technical Specifications (CCTS) Core Data Types of the WCO data elements, thereby facilitating interoperability between WCO XML instance documents derived from these schemas with instance documents derived from UN/CEFACT XML schemas and UBL (Universal Business Language) XML schemas.

The sample XML schemas also make use of a Data Set (DS) schema as the building block. The DS schema defines each WCO data element in a named type (named complex type or named simple type). This ensures that all the XML leaf elements in the sample XML schemas must be WCO data elements.

The illustrations provided in these guidelines are based on the XML tool “Altova XML Spy”. Users may adjust the instructions in order to apply the method in their own tools. In fact a simple text editor can also be used to carry out the method described in these guidelines, which is also illustrated below.

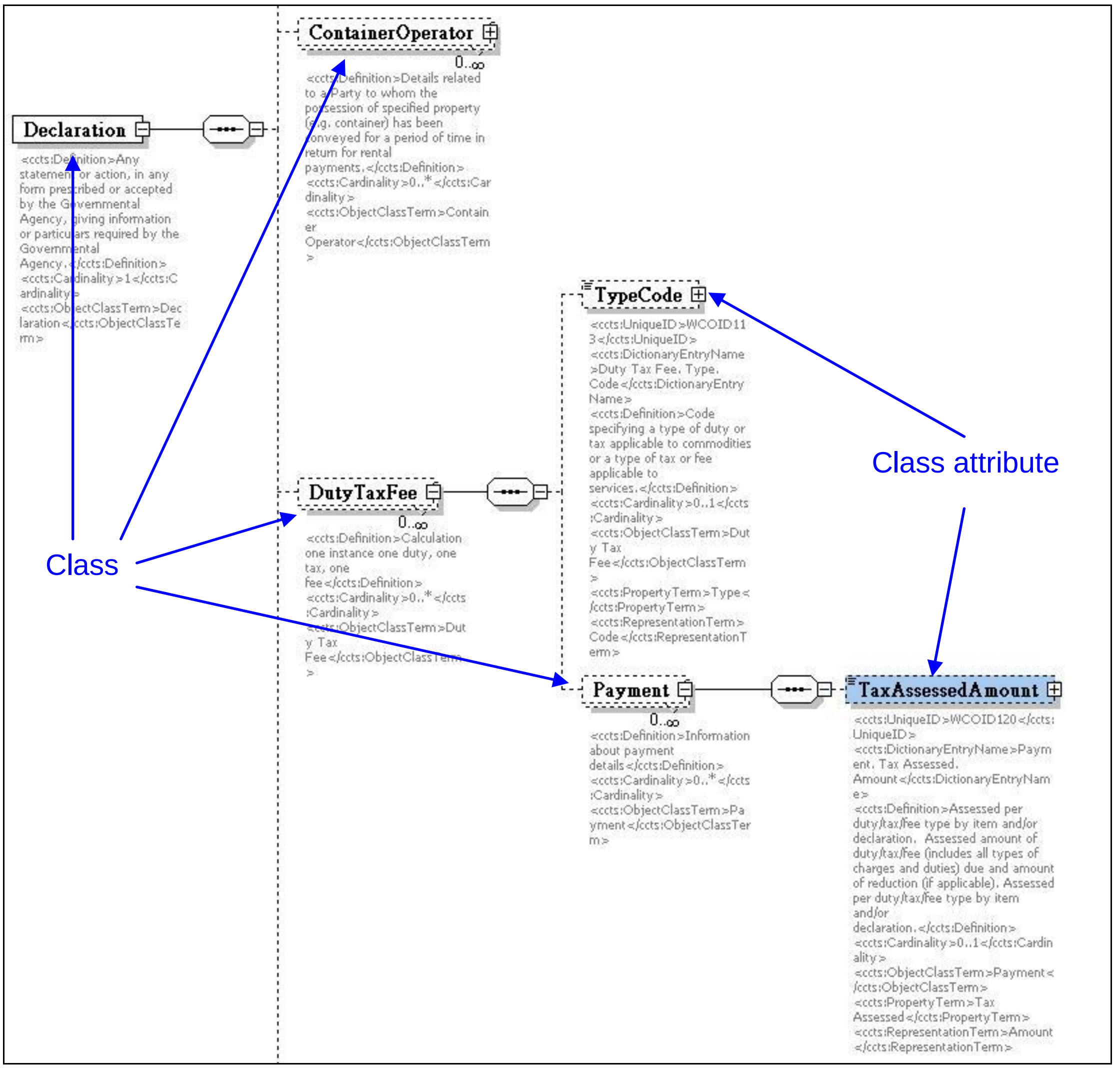
10.1 Purpose of XML Schema Customization

The sample XML schemas are derived from WCO Data Model class diagrams which represent B2G (Business to Government) /G2B (Government to Business) documents involved in particular customs procedures. As the B2G/ G2B overall model contains the maximum set of common data elements used by all CBRAs, not all data elements in each B2G / G2B document will be needed by an individual government agency. Those extra data elements will become XML elements not used in the XML schema and may lead to unnecessary confusion. Besides, individual administration may need to restrict the cardinality of class or class attribute to reflect their local requirements, e.g. local regulation stipulates that only one border transport means information in a Cargo Report Import procedure is needed.

The purpose of this document is to illustrate how individual CBRA can make reference to and customize the sample XML schemas to suit their specific needs, by means of trimming off unnecessary classes/class attributes, and modifying the cardinality of classes/class attributes.

10.2 How Class and Class Attribute are represented in the Sample XML Schema

In the sample XML schema, a class is represented by a non-leaf XML element whereas a class attribute is represented by a leaf XML element. Figure 2 shows how these are represented using the graphical features of an XSD Editor[11] software and Figure 3 shows the representation when using a Text Editor.



```
<ccts:Definition>Any statement or action, in any form prescribed or accepted by the Governmental Agency, giving information or particulars required by the Governmental Agency.</ccts:Definition>
<ccts:Cardinality>1</ccts:Cardinality>
<ccts:ObjectClassTerm>Declaration</ccts:ObjectClassTerm>
```

```
<ccts:Definition>Details related to a Party to whom the possession of specified property (e.g. container) has been conveyed for a period of time in return for rental payments.</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Container Operator</ccts:ObjectClassTerm>
```

```
<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
```

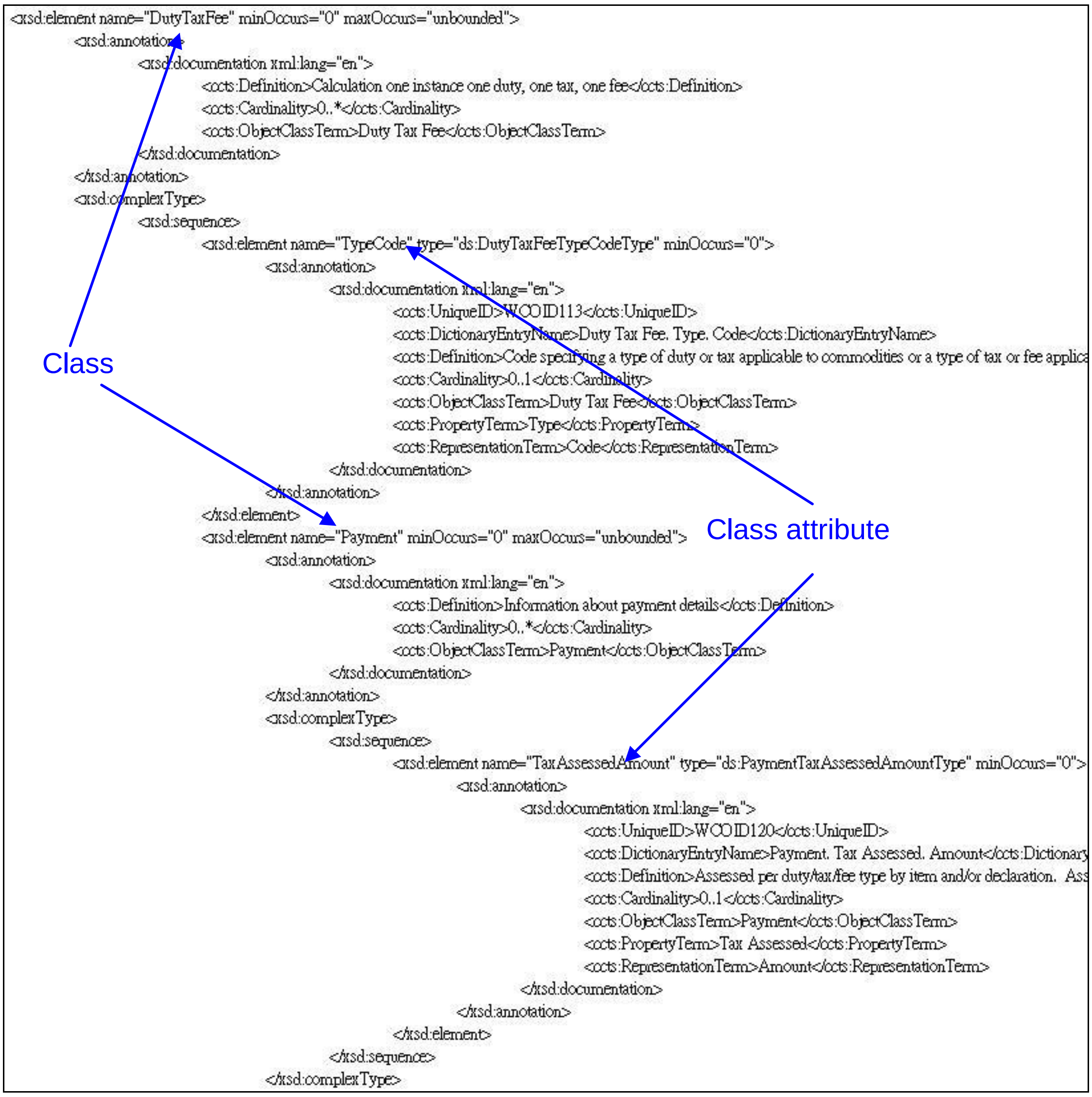
```
<ccts:UniqueID>WCOID113</ccts:UniqueID>
<ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName>
<ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applicable to services.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Type</ccts:PropertyTerm>
<ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
```

```
<ccts:Definition>Information about payment details</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
```

```
<ccts:UniqueID>WCOID120</ccts:UniqueID>
<ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:DictionaryEntryName>
<ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration.  Assessed amount of duty/tax/fee (includes all types of charges and duties) due and amount of reduction (if applicable). Assessed per duty/tax/fee type by item and/or declaration.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm>
<ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>
```

Figure 2: Example of non-leaf and leaf XML elements (XSD Editor)

[11] XSD editor is a software for creating XML Schemas .



```
<xsd:element name="DutyTaxFee" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            <ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
            <ccts:Cardinality>0..*</ccts:Cardinality>
            <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="TypeCode" type="ds:DutyTaxFeeTypeCodeType" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        <ccts:UniqueID>WCOID113</ccts:UniqueID>
                        <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName>
                        <ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applica
                        <ccts:Cardinality>0..1</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                        <ccts:PropertyTerm>Type</ccts:PropertyTerm>
                        <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="Payment" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        <ccts:Definition>Information about payment details</ccts:Definition>
                        <ccts:Cardinality>0..*</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="TaxAssessedAmount" type="ds:PaymentTaxAssessedAmountType" minOccurs="0">
                            <xsd:annotation>
                                <xsd:documentation xml:lang="en">
                                    <ccts:UniqueID>WCOID120</ccts:UniqueID>
                                    <ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:Dictionary
                                    <ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration.  Ass
                                    <ccts:Cardinality>0..1</ccts:Cardinality>
                                    <ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
                                    <ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm>
                                    <ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>
                                </xsd:documentation>
                            </xsd:annotation>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
```

Figure 3: Example of non-leaf and leaf XML elements (Text Editor)

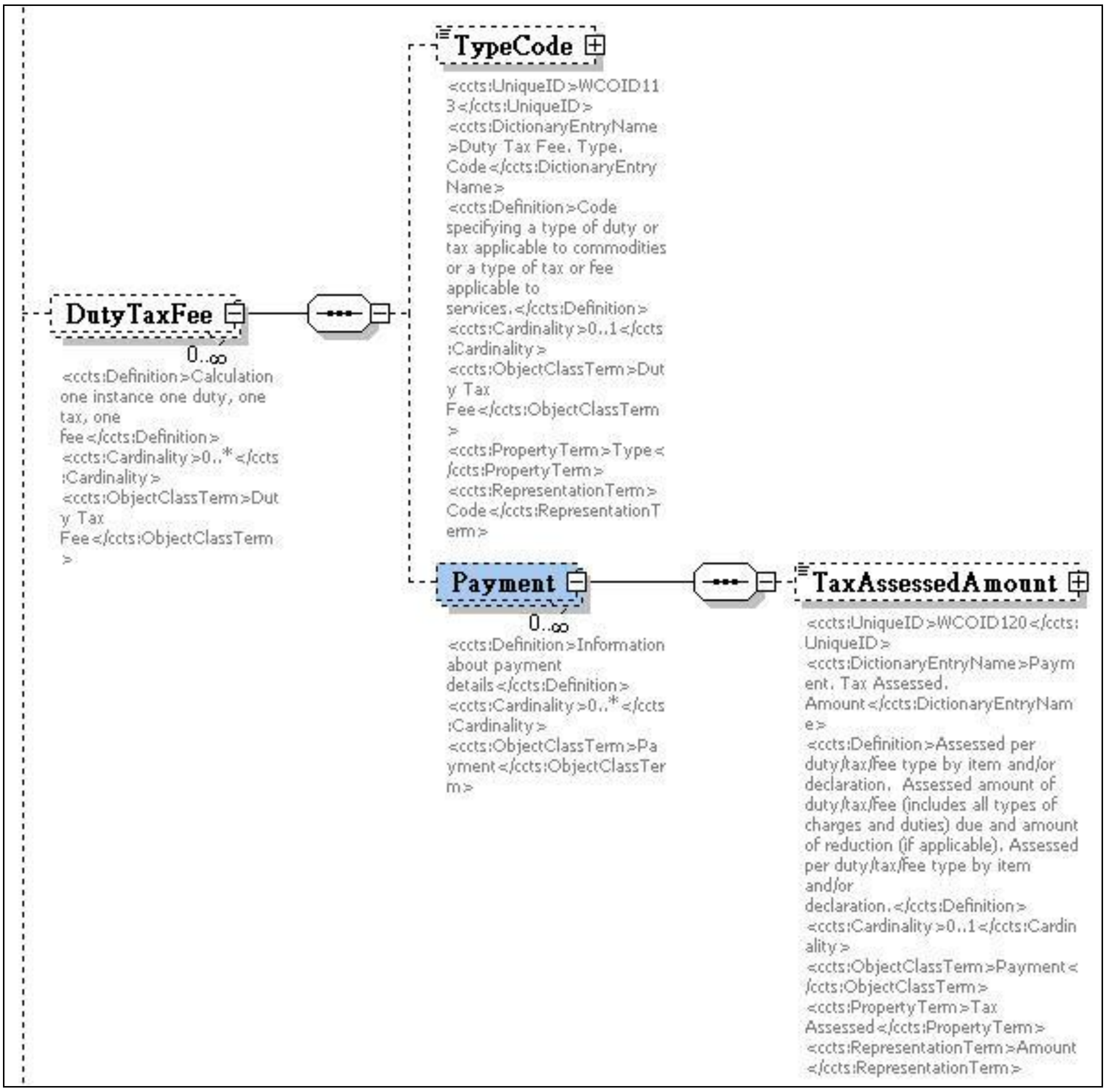
11 Customizing the Sample XML Schema

After determining the requirements of their XML messages, individual CBRAs can use the following three operations to customize the sample XML schema to match their requirements. The three operations include: (1) trim class operation, (2) trim class attribute operation, and (3) modify cardinality operation. The following sections will illustrate how to perform these operations using an XSD Editor (graphical view), and then illustrate how these can be performed using a Text Editor. Nevertheless, readers can also use other suitable tools to carry out the operations.

11.1 Customization using the Graphical View of an XSD Editor.

11.1.1 Trim Class (non-leaf XML element) Operation

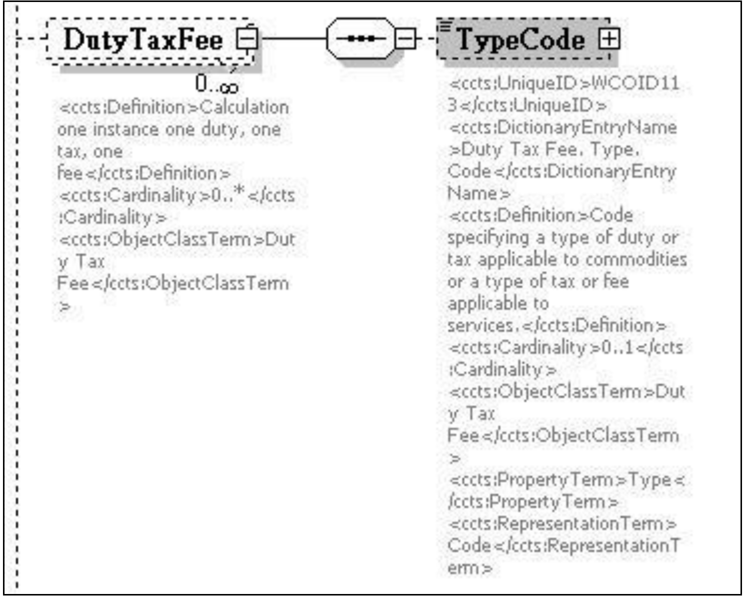
1. Open the XML schema with XSD Editor in the graphical view or tree view.
2. Browse to the XML element representing the class and highlight it by clicking the XML element as shown in Figure 4.



```
<ccts:UniqueID>WCOID113</ccts:UniqueID> <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName> <ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applicable to services.</ccts:Definition> <ccts:Cardinality>0..1</ccts:Cardinality> <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm> <ccts:PropertyTerm>Type</ccts:PropertyTerm> <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>

<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition> <ccts:Cardinality>0..*</ccts:Cardinality> <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>

<ccts:Definition>Information about payment details</ccts:Definition> <ccts:Cardinality>0..*</ccts:Cardinality> <ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>

<ccts:UniqueID>WCOID120</ccts:UniqueID> <ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:DictionaryEntryName> <ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration. Assessed amount of duty/tax/fee (includes all types of charges and duties) due and amount of reduction (if applicable). Assessed per duty/tax/fee type by item and/or declaration.</ccts:Definition> <ccts:Cardinality>0..1</ccts:Cardinality> <ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm> <ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm> <ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>
```

Figure 4: Highlighting a non-leaf XML element (XSD Editor)

3. Press delete key to complete the operation as shown in Figure 5.



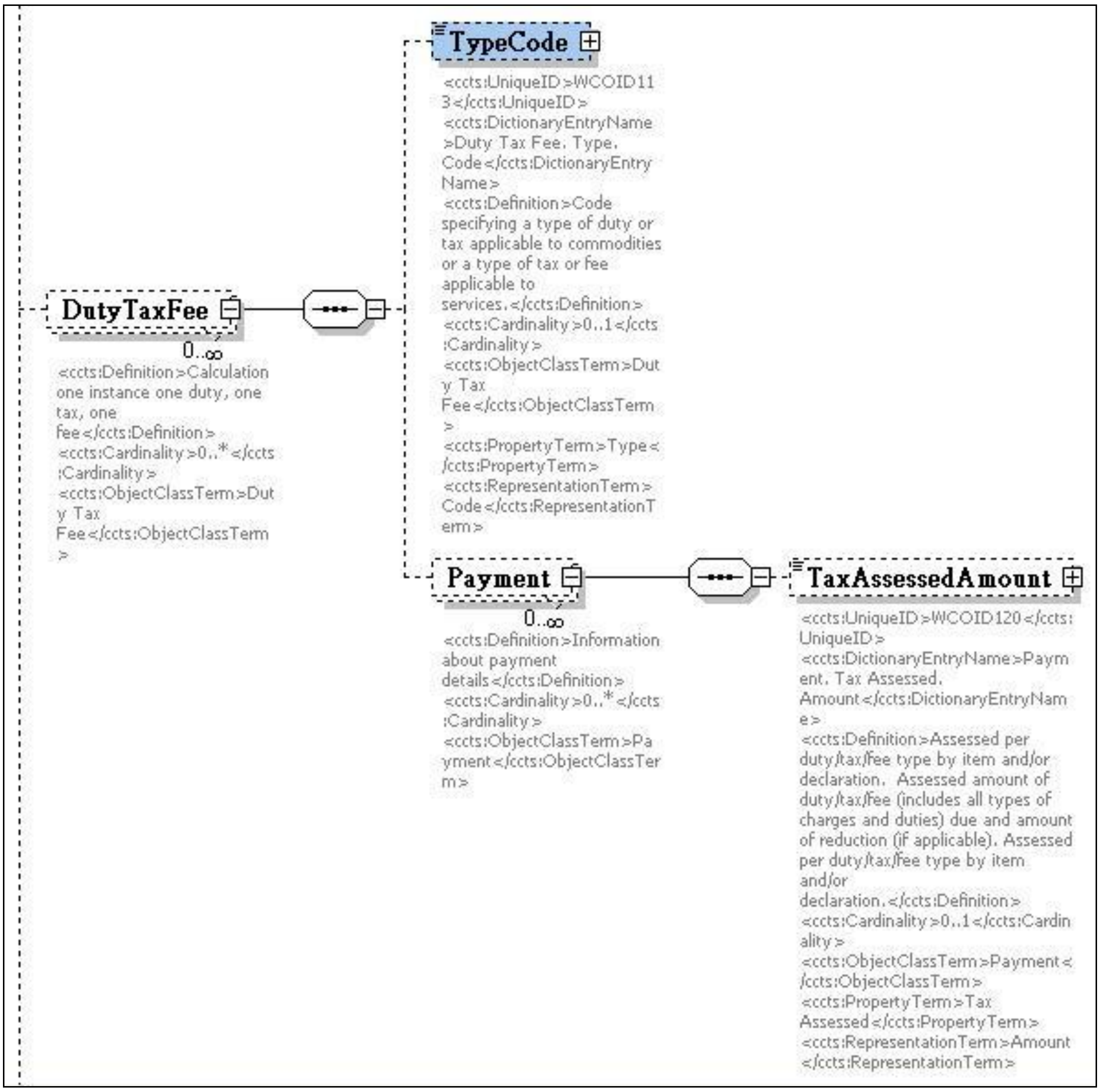
<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition> <ccts:Cardinality>0..*</ccts:Cardinality> <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>

<ccts:UniqueID>WCOID113</ccts:UniqueID> <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName> <ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applicable to services.</ccts:Definition> <ccts:Cardinality>0..1</ccts:Cardinality> <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm> <ccts:PropertyTerm>Type</ccts:PropertyTerm> <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>

Figure 5: Completing the operation (non-leaf XML element / XSD Editor)

11.1.2 Trim Class Attribute (leaf XML element) Operation

1. Open the XML schema with XSD Editor Graphical view or tree view.
2. Browse to the XML element representing the class attribute and highlight it by clicking the XML element as shown in Figure 6.



```
<ccts:UniqueID>WCOID113</ccts:UniqueID>
<ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName>
<ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applicable to services.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Type</ccts:PropertyTerm>
<ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
```

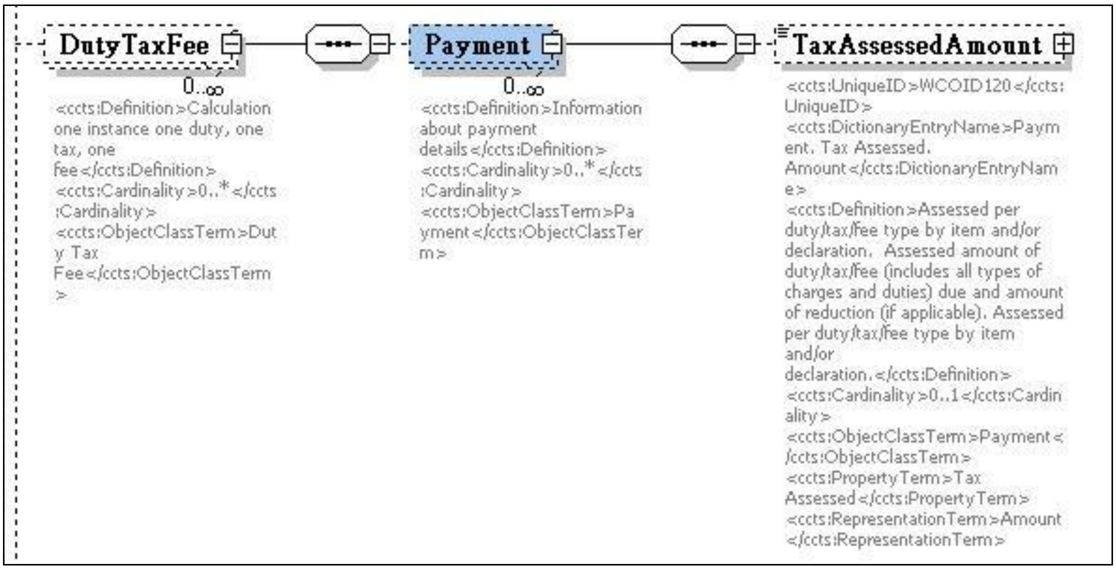
```
<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
```

```
<ccts:Definition>Information about payment details</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
```

```
<ccts:UniqueID>WCOID120</ccts:UniqueID>
<ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:DictionaryEntryName>
<ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration.  Assessed amount of duty/tax/fee (includes all types of charges and duties) due and amount of reduction (if applicable). Assessed per duty/tax/fee type by item and/or declaration.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm>
<ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>
```

Figure 6: Highlighting a leaf XML element (XSD Editor)

3. Press delete key to complete the operation as shown in Figure 7.



<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>

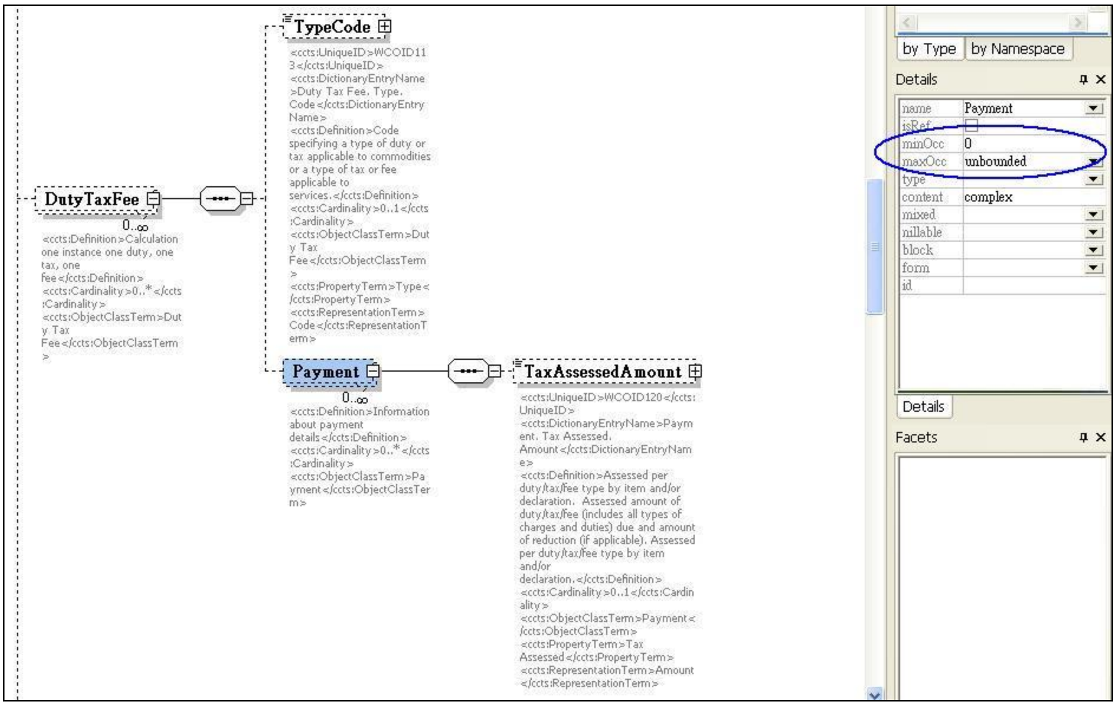
<ccts:Definition>Information about payment details</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>

<ccts:UniqueID>WCOID120</ccts:UniqueID>
<ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:DictionaryEntryName>
<ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration. Assessed amount of duty/tax/fee (includes all types of charges and duties) due and amount of reduction (if applicable). Assessed per duty/tax/fee type by item and/or declaration.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm>
<ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>

Figure 7: Complete the operation (leaf XML element / XSD Editor)

11.1.3 Modify Cardinality Operation

1. Open the XML schema with XSD Editor in the graphical or tree view.
2. Browse to the XML element representing the class/class attribute and highlight it by clicking the XML element as shown in Figure 8.
3. Change the value of minOcc and maxOcc in the Details Window as circled in Figure 8.



<ccts:UniqueID>WCOID113</ccts:UniqueID>
<ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:DictionaryEntryName>
<ccts:Definition>Code specifying a type of duty or tax applicable to commodities or a type of tax or fee applicable to services.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Type</ccts:PropertyTerm>
<ccts:RepresentationTerm>Code</ccts:RepresentationTerm>

<ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>

<ccts:Definition>Information about payment details</ccts:Definition>
<ccts:Cardinality>0..*</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>

<ccts:UniqueID>WCOID120</ccts:UniqueID>
<ccts:DictionaryEntryName>Payment. Tax Assessed. Amount</ccts:DictionaryEntryName>
<ccts:Definition>Assessed per duty/tax/fee type by item and/or declaration. Assessed amount of duty/tax/fee (includes all types of charges and duties) due and amount of reduction (if applicable). Assessed per duty/tax/fee type by item and/or declaration.</ccts:Definition>
<ccts:Cardinality>0..1</ccts:Cardinality>
<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
<ccts:PropertyTerm>Tax Assessed</ccts:PropertyTerm>
<ccts:RepresentationTerm>Amount</ccts:RepresentationTerm>

name	Payment
isRef	□
minOcc	0
maxOcc	unbounded
type	
content	complex
mixed	
nillable	
block	
form	
id	

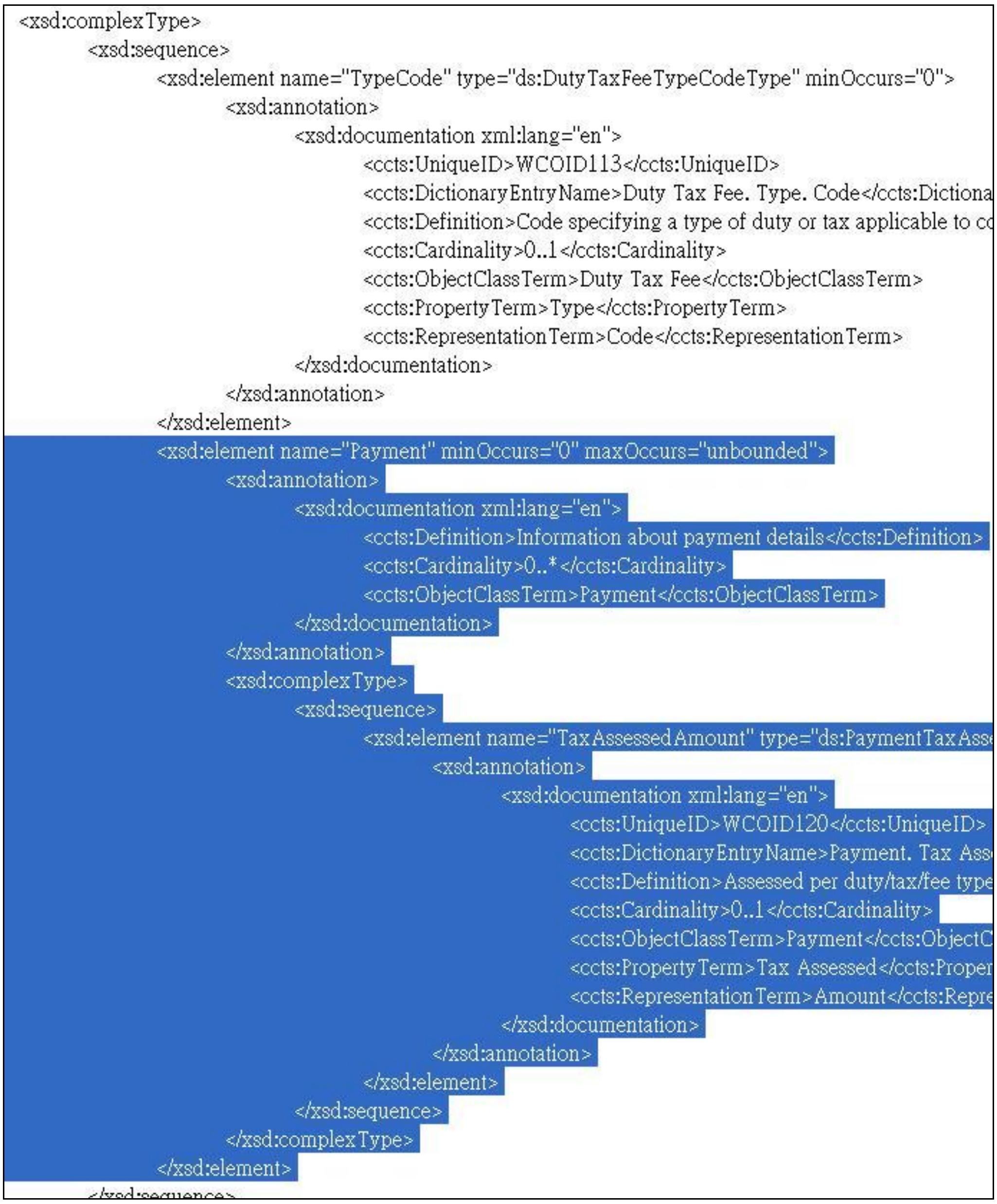
Figure 8: Changing cardinality (XSD Editor)

Note: The value of minOcc must be smaller than or equal to the value of maxOcc. For compatibility, the value of minOcc should be larger than or equal to the original value whereas the value of maxOcc should be smaller than or equal to the original value.

11.2 Customization using Text Editor

11.2.1 Trim Class (non-leaf XML element) Operation

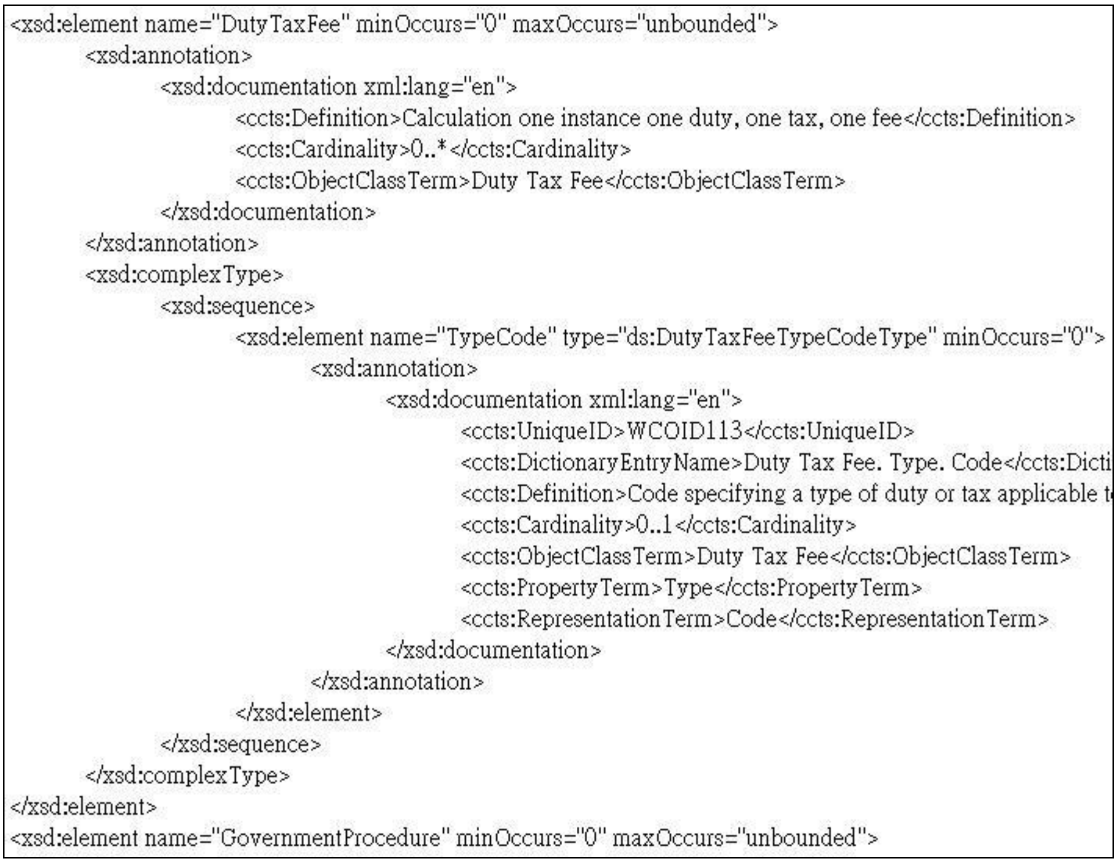
1. Open the XML schema with Text Editor.
2. Search the XML element which represents the class.
3. Select the text between <xsd:element …> and the corresponding </xsd:element> which should be at the same level of the <xsd:element …>, as shown in Figure 9.



```
<xsd:complexType>
	<xsd:sequence>
		<xsd:element name="TypeCode" type="ds:DutyTaxFeeTypeCodeType" minOccurs="0">
			<xsd:annotation>
				<xsd:documentation xml:lang="en">
					<ccts:UniqueID>WCOID113</ccts:UniqueID>
					<ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:Dictiona
					<ccts:Definition>Code specifying a type of duty or tax applicable to co
					<ccts:Cardinality>0..1</ccts:Cardinality>
					<ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
					<ccts:PropertyTerm>Type</ccts:PropertyTerm>
					<ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
				</xsd:documentation>
			</xsd:annotation>
		</xsd:element>
		<xsd:element name="Payment" minOccurs="0" maxOccurs="unbounded">
			<xsd:annotation>
				<xsd:documentation xml:lang="en">
					<ccts:Definition>Information about payment details</ccts:Definition>
					<ccts:Cardinality>0..*</ccts:Cardinality>
					<ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
				</xsd:documentation>
			</xsd:annotation>
			<xsd:complexType>
				<xsd:sequence>
					<xsd:element name="TaxAssessedAmount" type="ds:PaymentTaxAsse
						<xsd:annotation>
							<xsd:documentation xml:lang="en">
								<ccts:UniqueID>WCOID120</ccts:UniqueID>
								<ccts:DictionaryEntryName>Payment. Tax Asse
								<ccts:Definition>Assessed per duty/tax/fee type
								<ccts:Cardinality>0..1</ccts:Cardinality>
								<ccts:ObjectClassTerm>Payment</ccts:ObjectC
								<ccts:PropertyTerm>Tax Assessed</ccts:Proper
								<ccts:RepresentationTerm>Amount</ccts:Repre
							</xsd:documentation>
						</xsd:annotation>
					</xsd:element>
				</xsd:sequence>
			</xsd:complexType>
		</xsd:element>
	</xsd:sequence>
```

Figure 9: Highlighting a non-leaf XML element (Text Editor)

4. Press delete key to complete the operation as shown in Figure 10.

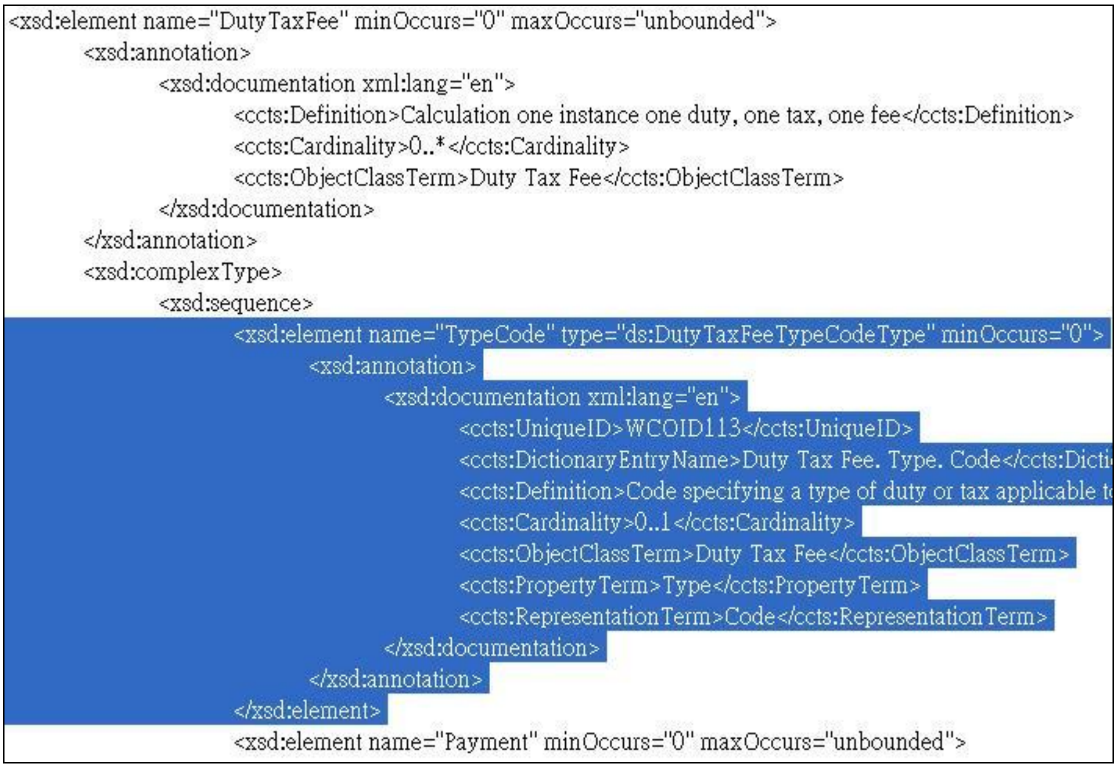


```
<xsd:element name="DutyTaxFee" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
                <xsd:documentation xml:lang="en">
                        <ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
                        <ccts:Cardinality>0..*</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="TypeCode" type="ds:DutyTaxFeeTypeCodeType" minOccurs="0">
                                <xsd:annotation>
                                        <xsd:documentation xml:lang="en">
                                                <ccts:UniqueID>WCOID113</ccts:UniqueID>
                                                <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:Dicti
                                                <ccts:Definition>Code specifying a type of duty or tax applicable t
                                                <ccts:Cardinality>0..1</ccts:Cardinality>
                                                <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                                                <ccts:PropertyTerm>Type</ccts:PropertyTerm>
                                                <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
                                        </xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                </xsd:sequence>
        </xsd:complexType>
</xsd:element>
<xsd:element name="GovernmentProcedure" minOccurs="0" maxOccurs="unbounded">
```

Figure 10: Complete the operation (non-leaf XML element / Text Editor)

11.2.2 Trim Class Attribute (leaf XML element) Operation

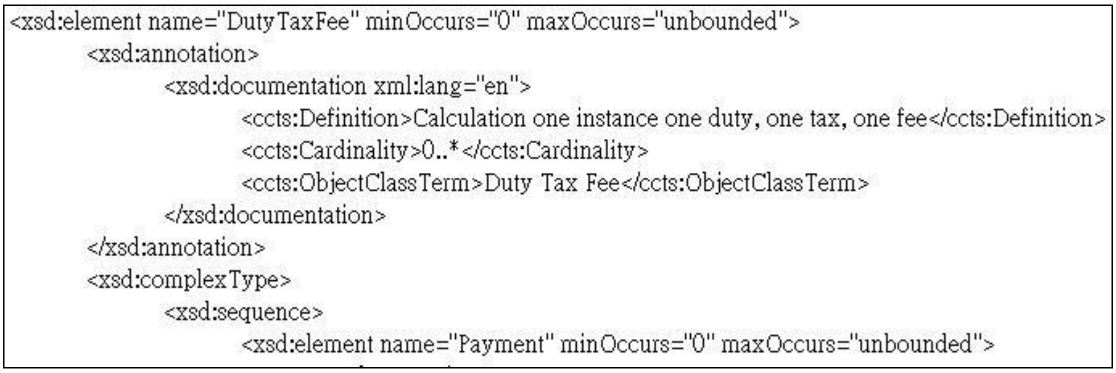
1. Open the XML schema with Text Editor.
2. Search the XML element which represents the class attribute.
3. Select the text between <xsd:element …> and the corresponding </xsd:element> which should be at the same level of the <xsd:element …>, as shown in Figure 11.



```
<xsd:element name="DutyTaxFee" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
                <xsd:documentation xml:lang="en">
                        <ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
                        <ccts:Cardinality>0..*</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="TypeCode" type="ds:DutyTaxFeeTypeCodeType" minOccurs="0">
                                <xsd:annotation>
                                        <xsd:documentation xml:lang="en">
                                                <ccts:UniqueID>WCOID113</ccts:UniqueID>
                                                <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:Dicti
                                                <ccts:Definition>Code specifying a type of duty or tax applicable t
                                                <ccts:Cardinality>0..1</ccts:Cardinality>
                                                <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                                                <ccts:PropertyTerm>Type</ccts:PropertyTerm>
                                                <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
                                        </xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                        <xsd:element name="Payment" minOccurs="0" maxOccurs="unbounded">
```

Figure 11: Highlighting a leaf XML element (Text Editor)

4. Press delete key to complete the operation as shown in Figure 12.

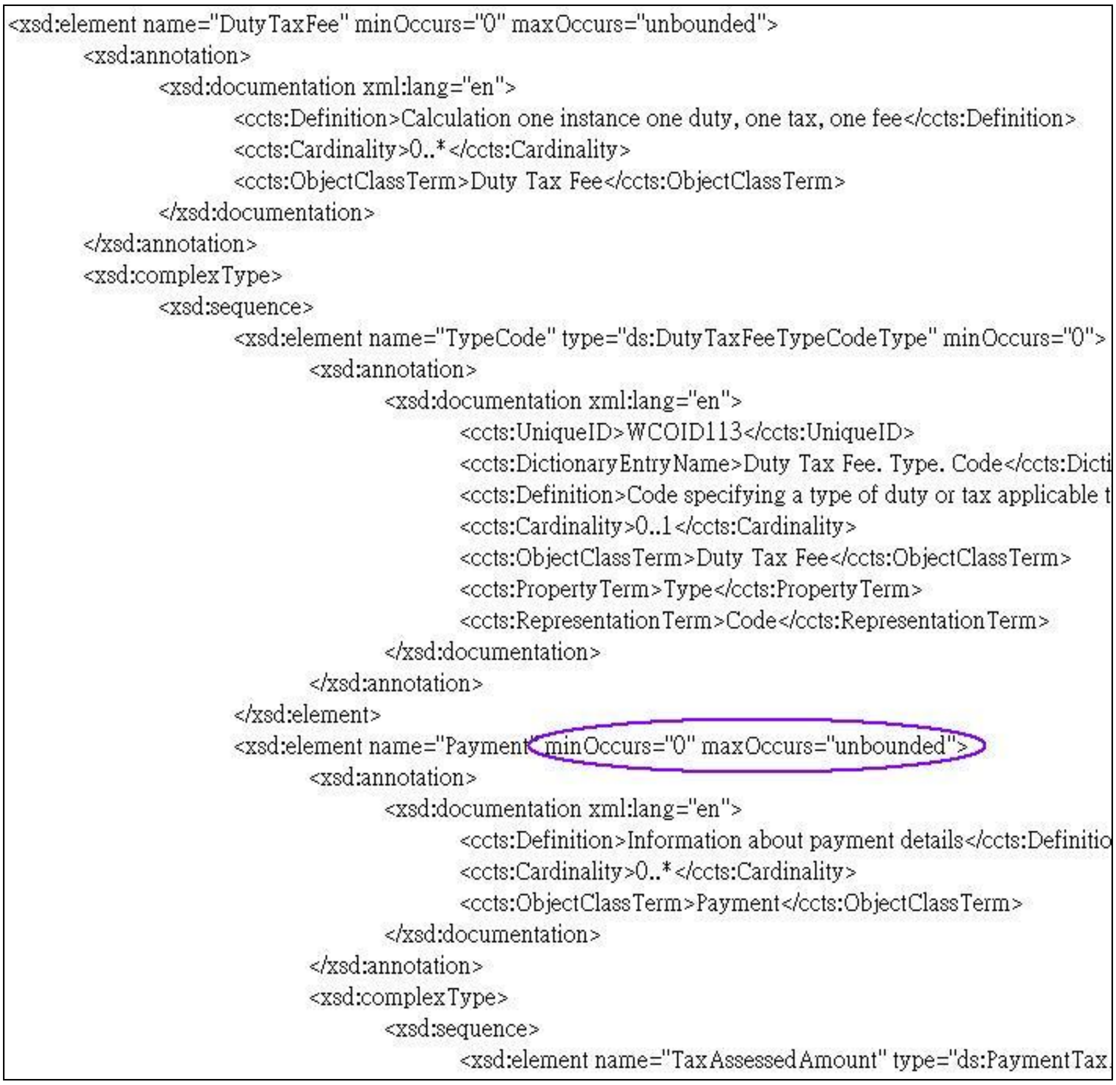


```
<xsd:element name="DutyTaxFee" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
                <xsd:documentation xml:lang="en">
                        <ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
                        <ccts:Cardinality>0..*</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="Payment" minOccurs="0" maxOccurs="unbounded">
```

Figure 12: Complete the operation (leaf XML element / Text Editor)

11.2.3 Modify Cardinality Operation

1. Open the XML schema with Text Editor.
2. Search the XML element representing the class/class attribute.
3. Change the value of minOccurs and maxOccurs, as circled in Figure 13.



```
<xsd:element name="DutyTaxFee" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            <ccts:Definition>Calculation one instance one duty, one tax, one fee</ccts:Definition>
            <ccts:Cardinality>0..*</ccts:Cardinality>
            <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="TypeCode" type="ds:DutyTaxFeeTypeCodeType" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        <ccts:UniqueID>WCOID113</ccts:UniqueID>
                        <ccts:DictionaryEntryName>Duty Tax Fee. Type. Code</ccts:Dicti
                        <ccts:Definition>Code specifying a type of duty or tax applicable t
                        <ccts:Cardinality>0..1</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Duty Tax Fee</ccts:ObjectClassTerm>
                        <ccts:PropertyTerm>Type</ccts:PropertyTerm>
                        <ccts:RepresentationTerm>Code</ccts:RepresentationTerm>
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="Payment" minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                        <ccts:Definition>Information about payment details</ccts:Definitio
                        <ccts:Cardinality>0..*</ccts:Cardinality>
                        <ccts:ObjectClassTerm>Payment</ccts:ObjectClassTerm>
                    </xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="TaxAssessedAmount" type="ds:PaymentTax.
```

Figure 13: Changing cardinality (Text Editor)

Note: The value of minOccurs must be smaller or equal to the value of maxOccurs. For compatibility, the value of minOccurs should be larger than or equal to the original value whereas the value of maxOccurs should be smaller than or equal to the original value. There will be no minOccurs/maxOccurs XML attribute when the occurrence is 1.

Annex

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
1 a)	Amount	Decimal	A number of monetary units specified in a currency where the unit of the currency is explicit or implied.	currencyID	The currency of the amount.	Mandatory	String	Use UN/ECE Rec. 9 code list. The UN/ECE Rec. 9 is also published as ISO 4217.
2 a)	Binary Object	Binary	A set of finite-length sequences of binary octets.	format	The format of the binary content.	Optional	String	
b)				mimeCode	The mime type of the binary object.	Optional	String	Reference IETF RFC 2045, 2046, 2047.
c)				encodingCode	Specifies the decoding algorithm of the binary object.	Optional	String	Reference IETF RFC 2045, 2046, 2047.
d)				characterSetCode	The character set of the binary object if the mime type is text.	Optional	String	Reference IETF RFC 2045, 2046, 2047.
e)				uri	The Uniform Resource Identifier that identifies	Optional	String	

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
					where the binary object is located.			
f)				fileName	The filename of the binary object.	Optional	String	
3 a)	Code	String	A character string (letters, figures, or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute together with relevant supplementary information.	listID	The identification of a list of codes.	Optional	String	
b)				listAgencyID	An agency that maintains one or more lists of codes.	Optional	String	Use UN/EDIFACT data element 3055 code list.
c)				listAgencyName	The name of the agency that maintains the list of codes.	Optional	String	
d)				listName	The name of a list of	Optional	String	

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
					codes.			
e)				listVersionID	The version of the code list.	Optional	String	
f)				name	The textual equivalent of the code content component.	Optional	String	
g)				languageID	The identifier of the language used in the code name.	Optional	String	
h)				listURI	The Uniform Resource Identifier that identifies where the code list is located.	Optional	String	
i)				listSchemeURI	The Uniform Resource Identifier that identifies where the code list scheme is located.	Optional	String	
4 a)	DateTime	String	A particular point in the progression	formatCode	The format of the	Optional	String	Use UN/EDIFACT

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
			of time together with the relevant supplementary information.		date/time content			data element 2379 code list. Examples of codes to be used: 102 (CCYYMMDD), 304 (CCYYMMDDHHMMSSZZZ) and 602 (CCYY)
5 a)	Identifier	String	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects in the same scheme together with relevant supplementary information.	schemeID	The identification of the identification scheme.	Optional	String	
b)				schemeName	The name of the identification scheme.	Optional	String	
c)				schemeAgencyID	The identification of the agency that maintains	Optional	String	Use UN/EDIFACT data element 3055

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
					the identification scheme.			code list.
d)				schemeAgencyName	The name of the agency that maintains the identification scheme.	Optional	String	
e)				schemeVersionID	The version of the identification scheme.	Optional	String	
f)				schemeDataURI	The Uniform Resource Identifier that identifies where the identification scheme data is located.	Optional	String	
g)				schemeURI	The Uniform Resource Identifier that identifies where the identification scheme is located.	Optional	String	
6 a)	Indicator	String	A list of two mutually exclusive					

Annex

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
			Boolean values that express the only possible states of a property.					
7 a)	Measure	Decimal	A numeric value determined by measuring an object along with the specified unit of measure.	unitCode	The type of unit of measure.	Mandatory	String	Use UN/ECE Rec.20 code list
8 a)	Numeric	Decimal	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.					
9 a)	Percent	Decimal	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.					
10 a)	Quantity	Decimal	A counted number of non-monetary units possibly including fractions.	unitCode	The unit of the quantity	Mandatory	String	Use UN/ECE Rec.20 code list

Annex

List of Attributes of Core Component Types used in WCO Data Model

Ref.	Core Component Type Name	Data Type of Core Component Type	Core Component Type Description	Attribute Name	Attribute Description	Mandatory / Optional	Attribute Data Type	Remarks
12)	Rate	Decimal	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.					
13)	Text	String	A character string (i.e. a finite set of characters) generally in the form of words of a language.	languageID	The identifier of the language used in the content component.	Optional	String	