



Practical Code List Implementation (Using Controlled Vocabularies in XML Documents)

Crane Softwrights Ltd.
<http://www.CraneSoftwrights.com>





Practical Code List Implementation (Using Controlled Vocabularies in XML Documents)

Crane Softwrights Ltd.

<http://www.CraneSoftwrights.com>

Copyrights

- Other original material herein is copyright (C) 1998-2009 Crane Softwrights Ltd. This is commercial material and may not be copied or distributed by any means whatsoever without the expressed permission of Crane Softwrights Ltd.

Disclaimer

- By purchasing and/or using any product from Crane Softwrights Ltd. ("Crane"), the product user ("reader") understands that this product may contain errors and/or other inaccuracies that may result in a failure to use the product itself or other software claiming to utilize any proposed or finalized standards or recommendations referenced therein. Consequently, it is provided "AS IS" and Crane disclaims any warranty, conditions, or liability obligations to the reader of any kind. The reader understands and agrees that Crane does not make any express, implied, or statutory warranty or condition of any kind for the product including, but not limited to, any warranty or condition with regard to satisfactory quality, merchantable quality, merchantability or fitness for any particular purpose, or such arising by law, statute, usage of trade, course of dealing or otherwise. In no event will Crane be liable for (a) punitive or aggravated damages; (b) any direct or indirect damages, including any lost profits, lost savings, damaged data or other commercial or economic loss, or any other incidental or consequential damages even if Crane or any of its representatives have been advised of the possibility of such damages or they are foreseeable; or (c) for any claim of any kind by any other party. Reader acknowledges and agrees that they bear the entire risk as to the quality of the product.

OASIS Copyrights

- Some information included in this publication is from copyrighted material from the Organization for the Advancement of Structured Information Standards (OASIS) <http://www.oasis-open.org>. Files containing the copyrighted material include the following, which applies only to original OASIS documents and not to this commercial material created by Crane: please go to original OASIS documents to obtain any publicly-available content:
 - Portions copyright (C) OASIS Open 2001-2009. All Rights Reserved.
 - <http://www.oasis-open.org/who/intellectualproperty.php>

Practical Code List Implementation (Prelude) (cont.)



Preface

The main content of this book is in an unconventional style primarily in bulleted form

- derivations of the book are used for instructor-led training, requiring the succinct presentation
 - note the exercises included in instructor-led training sessions are not included in the book
- derivations of the book can be licensed and branded for customer use in delivering training
- the objective of this style is to convey the essence and details desired in a compact, easily perused form, thereby reducing the search for key words and phrases in lengthy paragraphs
- each chapter of the book corresponds to a module of the training
- each page of the book corresponds to a frame presented in the training
- a summary of subsections and their pages is at the back of the book

Much of the content is hyperlinked both internally and externally to the book in the 1-up full-page sized electronic renditions:

- (note the Acrobat Reader "back" keystroke sequence is "Ctrl-Left")
- page references (e.g.: Chapter 3 - Declaring controlled vocabularies (page 32))
- external references (e.g.: <http://docs.oasis-open.org/codelist/genericcode>)
- chapter references in book summary
- section references in chapter summary
- subsection references in table of contents at the back of the book
- hyperlinks are not present in the cut, stacked, half-page, or 2-up renditions of the material

Diagram legend:

- triangle: a structured XML/SGML/HTML document or resource
- parallelogram: a non-structured document or resource in an arbitrary format
- box: a process in the work flow
- diamond: a decision in the workflow

Sample code fragments:

- included with the book purchase is a ZIP file of sample code fragments
- directory names referenced in the book are referencing subdirectories in the unpacked ZIP files
- the `readme.txt` file in the ZIP package documents the running of sample batch files and shell scripts

Practical Code List Implementation (Prelude) (cont.)



Important caveat regarding the information in this publication

- while the author, G. Ken Holman, is the chairman of the OASIS Code List Representation Technical Committee, not all of the material in this publication is necessarily accepted by all of the TC members as some of it is just proposed for acceptance
- all of the content in this book is written from the opinion of G. Ken Holman and Crane Softwrights Ltd. and does not necessarily represent official or agreed-upon content from the perspective of the CLRTC
- this content is not to be construed as legal advice of any kind, nor is it recommending that any particular methodology or process or tool be implemented, it only documents methodologies and technologies available to be considered

Entire chapters of this publication will undergo revision

- the CLRTC is debating the CVA specification
- some software being developed by Crane Softwrights Ltd. is being made freely available for anyone to download and use
- some software being developed by Crane Softwrights Ltd. will be made available only to customers of this publication
 - to supplement the software that is made freely available

The purchase of this publication is protected by the no-charge availability of all future editions

- all of the content in this publication is subject to revision and update and editions will get out of date
- early editions are expected to be created frequently and be short-lived as the community experience with the code list file formats and processes reveals various practices and experiences that will influence how to consider working with this standard

The purchase of this publication grants the legitimate owner no-charge access to accompanying software written by Crane Softwrights Ltd.

- there are no warranties expressed or implied regarding the use of the software; more details are found in the documentation for the software
- access details to download the software are found by registered users on <http://www.CraneSoftwrights.com/sales/pcli/>
- while the software is free of charge, the software is not to be copied for or distributed to or used by anyone who is not a legitimate customer of this book, unless permission has been granted in writing

The author welcomes any and all suggestions for improvements and additional content

- please do not hesitate to contribute ideas for improving on this publication
- all submissions to info@CraneSoftwrights.com will be acknowledged (though not necessarily accepted!)
 - please note that aggressive spam filters may make our email delivery difficult

Practical Code List Implementation



- Introduction - Practical Code List Implementation
- Chapter 1 - Controlled vocabularies
- Chapter 2 - Defining and using controlled vocabularies
- Chapter 3 - Declaring controlled vocabularies
- Chapter 4 - Controlled vocabulary representation detail
- Chapter 5 - Associating controlled vocabularies in XML documents
- Chapter 6 - Controlled vocabulary association detail
- Chapter 7 - Your own business document controlled vocabulary
- Annex A - OpenOffice 3 genericcode and CVA filters
- Conclusion - Where to go from here?

Series: Practical Code List Implementation

Reference: Electronic commerce

Outcome

- detailed review of the concepts, implementation and deployment of code lists

Practical Code List Implementation

Introduction - Practical Code List Implementation



This book is oriented to the system architect and system developer interested in deploying OASIS specifications for code lists in XML

- how do external code lists differ from embedded code lists and to what benefit?
- what information design facets support flexible code list deployment?
- what responsibilities face a community needing to support a legacy and a future of code lists?
- what tools are available to be adapted for use in validating code lists?

This book covers that subset of OASIS genericcode 1.0 suitable for expressing a single code list, and all of the OASIS context/value association draft specification

- <http://docs.oasis-open.org/codelist/genericcode>
- http://www.oasis-open.org/committees/document.php?document_id=29990

The first two chapters overview controlled vocabularies in general

- what do they represent?
- how are they used?
- how, historically, have they been deployed?
- what new approaches are available for deployment?

The third and fourth chapters overview the specification of code lists in XML

- the use of the OASIS genericcode specification
- the detail of the OASIS genericcode vocabulary

The fifth and sixth chapters overview the specification of context/value association files

- the use of OASIS CVA files
- the detail of the OASIS CVA vocabulary

The last chapter outlines how to adapt validation stylesheets to use genericcode and CVA files for an arbitrary vocabulary

The only annex introduces Crane Softwrights Ltd.'s OpenOffice 3 XML filter package

Crane-gc2ods

- opening and saving OASIS genericcode files
- opening and saving OASIS CVA files

Enjoy!

Chapter 1 - Controlled vocabularies



- Introduction - XML document interchange
- Section 1 - Facets of controlled vocabularies

Outcomes

- understand the role of enumerated values in business documents

XML document interchange

Introduction - Chapter 1 - Controlled vocabularies



An XML document describes a hierarchy of information items

- XML is only responsible for representation of information and not the meaning of information
 - how information is labeled allows it to be identified, not interpreted
 - up to applications to interpret the meaning of the labels and information so-labeled
- each item is labeled using the document's XML vocabulary
 - the item's value is expressed in an attribute's specification or an element's text value
- document constraints describe limitations on the contents of the XML documents
 - what is allowed to be used as item labels and where
 - what is allowed to be used as item values and where
 - a document isn't XML unless it is well-formed
 - rules govern the proper labeling of the information in the hierarchy
 - labels can be comprised of namespace URI strings to be globally unique
 - the metaphor for "labels in a namespace" is "words in a dictionary"
 - different document constraint languages provide different validation features
 - directives of the language engage validation semantics

Business documents have many information items whose values are controlled

- code lists have been used for hundreds of years
 - show up in historical documents, business records, passenger manifests, etc.
- codes, identifiers, any information item with a predetermined value set
 - like a label, a code represents the semantic, it doesn't "mean" the semantic
 - nothing in the value conveys understanding, only representation
 - still up to an application recognizing the code to be pre-programmed to interpret the semantic associated with that code
 - sender and receiver need to agree on the understanding of the value
- the information's value is limited to one or more of a set of fixed values
- item values do not impact on the structure of the document

Two distinct kinds of "vocabularies" for interchange

- the XML vocabulary of element and attribute labels
- a controlled vocabulary of code or identifier values
- in document interchange, the vocabularies represent the concepts and information for commonly-understood semantics
- in applications, internal representations of both may be very much richer than the interchange representation

Controlled vocabulary semantics

Introduction - Chapter 1 - Controlled vocabularies



A controlled vocabulary is the set of agreed-upon values for a concept

- e.g. the list of country code abbreviations
 - e.g. "CA" for Canada, "US" for United States
- e.g. the list of currency code abbreviations
 - e.g. "CAD" for Canadian dollars, "USD" for United States dollars
- e.g. the list of transaction payment means
 - e.g. "10" for cash, "20" for cheque
- e.g. the list of units of measure
 - e.g. "KGM" for kilogram, "MTQ" for cubic meter
- e.g. identifiers for different kinds of dimensions
 - e.g. representing gross weight and net weight
- e.g. a company's private list of product and service identifiers
 - e.g. catalogue part numbers

Each value in a controlled vocabulary represents a particular semantic

- for obvious enumerated concepts, no semantic need be published authoritatively
 - e.g. the directions of latitude are either "North" or "South" of the equator
 - e.g. currency conversion operators are either "Multiply" or "Divide"
- for public vocabularies, the associated semantic is a published concept
 - managed by a public authority recognized as the trusted custodian of values
 - e.g. the International Organization for Standardization (ISO) list of country codes, currency codes, container sizes, etc.
 - e.g. the United Nations Economic Commission for Europe (UN/ECE) list of port codes, types of payment means, etc.
 - e.g. the Canadian Post Office list of Canadian province and territory abbreviations
- between trading partners, the associated semantic is an agreed-upon concept
 - e.g. the list of identifiers representing product and service offerings of a vendor
 - e.g. the document status codes accepted by a particular work flow specification

All parties implicitly agree to interpret the concepts in the same way in their independent applications

- by constraining the expression of the possible values to an agreed-upon set, both parties set expectations for interchange
- a formal expression of the constraints can form part of a business contract agreeing to limit values used to only the agreed-upon set
- traditional approaches to using W3C schema conflate the document constraints with the value constraints
 - new approaches are needed to layer value constraints on document constraints
- an important caveat: "obvious" values may not be obvious to all

Controlled vocabulary semantics (cont.)

Introduction - Chapter 1 - Controlled vocabularies



A controlled value is necessarily unique in a single given list

- if a given string value represented more than one concept it would be ambiguous and there would be no way to distinguish which concept was desired
- list meta data for a value distinguishes ambiguous values in combined lists
 - the values may overlap when meta data is used to identify which list's distinct value is being used
- the unique value is analogous to a relational database table key
 - used as a lookup value
- another use of the "words in a dictionary" metaphor
 - the meta data of the list defines which dictionary the words are from
 - the meta data may distinguish different versions of the same dictionary

Each controlled value may have many associated values

- value meta data may be simple strings or compound values
 - compound information can be expressed in rich markup
- analogous to relational database table columns
- display string(s)
- non-normative synonyms
- language translations
- supporting detail and nuance
- meta data
 - derivation method
 - source of information

ISO parlance has been in use a long time for code lists

- "Code" refers to a value's unique key value within its list
- "Name" refers to that value's description with which meaning is intended to be expressed
- for some concepts, far more information needs to be associated with values

Controlled vocabulary semantics (cont.)

Introduction - Chapter 1 - Controlled vocabularies



Controlled vocabularies are used in documents, databases, applications, messages

- by controlling the representation of a concept, a specified value can unambiguously identify the associated semantic
 - provided all users of the value understand the concept in the same manner
 - the burden is on the trusted custodian of the values to maintain the documentation of the list
- abbreviated values (codes) may provide a savings of effort or space when otherwise the expression of the concept is long-winded or wordy
 - the abbreviation is consistent
 - mnemonic values are typically biased to a particular language
 - e.g. "USD" mnemonic for "United States Dollars"
 - e.g. "ES" mnemonic for "Spain" ("España" is Spanish for "Spain")
 - non-mnemonic numeric values are often used as representations of abstract concepts without language bias
 - e.g. "42" non-mnemonic for "Payment to bank account" payment means
 - the mnemonic or non-mnemonic abbreviation is typically short
 - e.g. "51" non-mnemonic for "norme 6 97-Telereglement CFONB (French Organisation for Banking Standards) - Option A" payment means
- commonly used for centuries in messages to keep messages succinct

Promotes consistent interpretation of the value

- all applications can follow the published or agreed-upon semantics
- opportunity for misinterpretation through neglect or accident
- if each trading partner came up with their own abbreviations independently, it would be impossible to know that two different values represent the same concept
- removes language dependencies when abbreviating the same concept in two languages
 - though some codes are mnemonically derived from a native language, the rule governing that prevents the code from being derived differently in another language
- meta data columns can include various translations
 - promotes common interpretation

Codes and identifiers

Chapter 1 - Controlled vocabularies

Section 1 - Facets of controlled vocabularies



As a general rule of thumb (but not definitively), a controlled vocabulary information item is typically either a code or an identifier

- these are very symmetrically-defined constructs that are distinguished by arbitrary decisions of construction and use
 - guidelines and distinctions are not black and white
 - whether the values are characteristic or lookup can be twisted one way or the other
- these concepts are not always consistently applied
 - e.g. in UBL some identifiers could easily be codes and vice versa
- a code typically represents a unique concept, group or type using a *characteristic* value
 - e.g. a currency code for an account value - "GBP" (British pounds)
 - e.g. a unit of measure for a measurement - "MTR" (meters)
 - e.g. a shipping container's dimensions
 - this is an example of a set of coded values created by the application of a scheme on component parts of the value describing the container's height, width, depth and features
 - e.g. a method of transport
 - e.g. a document's type
- an identifier typically represents a unique thing or singleton from a group using a *lookup* value
 - may be synthesized by applying an algorithmic scheme
 - the range of identifier values may, however, be enumerated as members of a list
 - e.g. a particular account's identifier - "travel" or "supplies" or "ABC0001"
 - e.g. a particular dimension - "gross width" or "net width"
 - e.g. a particular aircraft's identifier
 - e.g. a particular catalogue item's identifier

Trading partners may wish to constrain either codes or identifiers or any other information item as a controlled vocabulary

- codes typically taken from a set representing known semantic concepts
- constraining an identifier would be from a fixed list of identifiers
 - e.g. a set of account identifiers
- open-ended identifiers would typically not be constrained
 - used to identify things that are being created

Code list registration authorities

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



The custodians of abstract code lists are typically the authorities with governance

- users of a list have a level of assurance regarding the maintenance of the sets of values
- stability is implied by the authority's governance of the concepts and expressions of the list members

The publishing of the list is different from the definition of the list

- the authority selects and defines codes in the abstract based on semantics (meaning)
- the list of codes is published hopefully with sufficient information to convey the semantics they represent so that all users interpret the codes as meaning the same concepts

The authorities can publish their code lists in many possible formats

- prose lists and descriptions
 - text files
 - word processing files
 - web site pages (HTML files)
 - algorithmic descriptions (e.g. ISBN checksum)
 - value assemblies (e.g. container height, width, depth and feature values)
- W3C schemas with annotations
- Comma Separated Values (CSV) files
- database tables
- colloquial XML expressions
 - using a bespoke document model invented by a community of users
 - an XML vocabulary not standardized outside of the community or users
- standardized XML expressions
 - using a published document model created by a committee effort
 - openness of process and access to results are important in assessing protection against private interests or encumbrances

Alternative expressions of lists may be made available in the absence of bona fide expressions

- a stop-gap measure to make up for the authority not having published the information in a useful form
- e.g. UBL has expressed a number of published abstract code lists using XML syntax until such time as the official custodians publish their own artefacts for public use

Identifying controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Some controlled vocabularies are already officially maintained

- custodians are typically international standards organizations
- e.g. currency codes by ISO (ISO 4217)
- e.g. country codes by ISO (ISO 3166-1)
- e.g. payment means codes by UN/ECE (UN/ECE 4461)

Projects must establish which codes are applicable to their work

- community responsibility
 - manage expectations of individuals and trading partners
 - guide community in common understanding of concepts and representations
- a subset of codes from established lists
 - don't re-invent the wheel
- new codes for use where an established list is deficient
 - are extensions needed for the community to use?
- new lists of codes where there are no established lists
 - are entire new lists required for a set of community semantics?

Where new codes are needed,

- what do each of them mean?
 - what meta data might be associated with each?
- how are they coded?
 - mnemonic? numeric? arbitrary?
- which values are unconstrained?

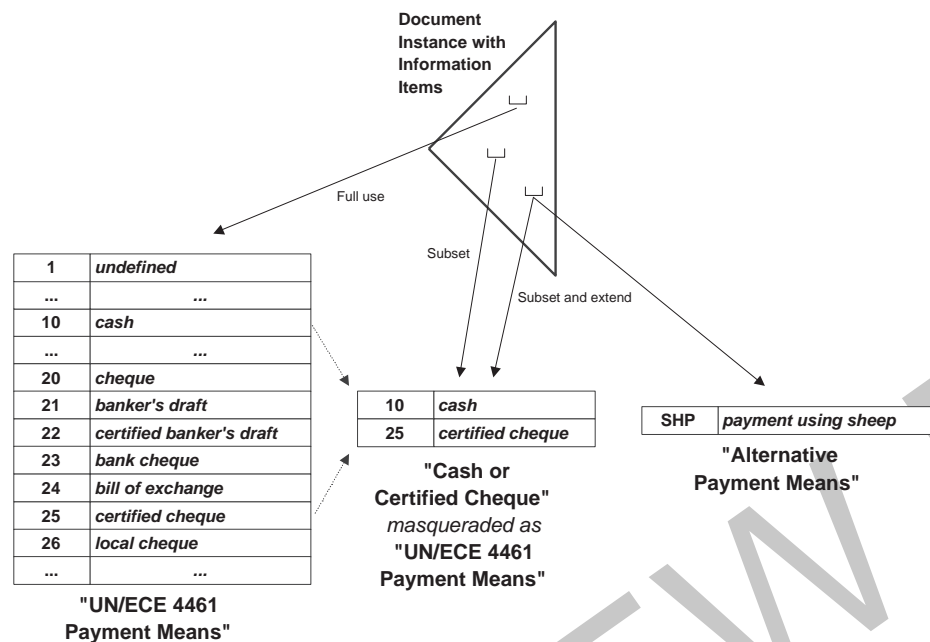
Identifying controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



List-level meta data identifies the list being used

- the needed list is an established list
- the list identification is the official list-level meta data



Identifying controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Value-level meta data qualifies the code with more detail

- code and name are not always sufficient to identify information
- value-level meta data can be used to distinguish facets of the code

Code	Name	Country	Subdivision	Port, rail, road, air, post, multi-modal, fixed, border	IATA
ADALV	Andorra la Vella	Andorra		3,4,6	ALV
...
USCB8	Columbus	United States	MT	2,3	CB8
USCBW	Columbus	United States	WI	3	CBW
USCLU	Columbus	United States	IN	3,4	CLU
USCMH	Columbus	United States	OH	4	CMH
USCSG	Columbus	United States	GA	3,4	CSG
USCUS	Columbus	United States	NM	3,4,B	CUS
USCZX	Columbus	United States	NC	3,6	CZX
USOLP	Columbus	United States	MO	3,6	OLP
USOLU	Columbus	United States	NE	3,4	OLU
USUBS	Columbus	United States	MS	3,4	UBS
USUCU	Columbus	United States	KS	2,3	UCU
USVCB	Columbus	United States	TX	3	VCB
USVDA	Columbus	United States	MI	4	VDA
USYBC	Columbus	United States	NJ	2,3,6	YBC
...
USCB8	Columbus	United States		4	WKI

UN/ECE Rec 16 LOCODE

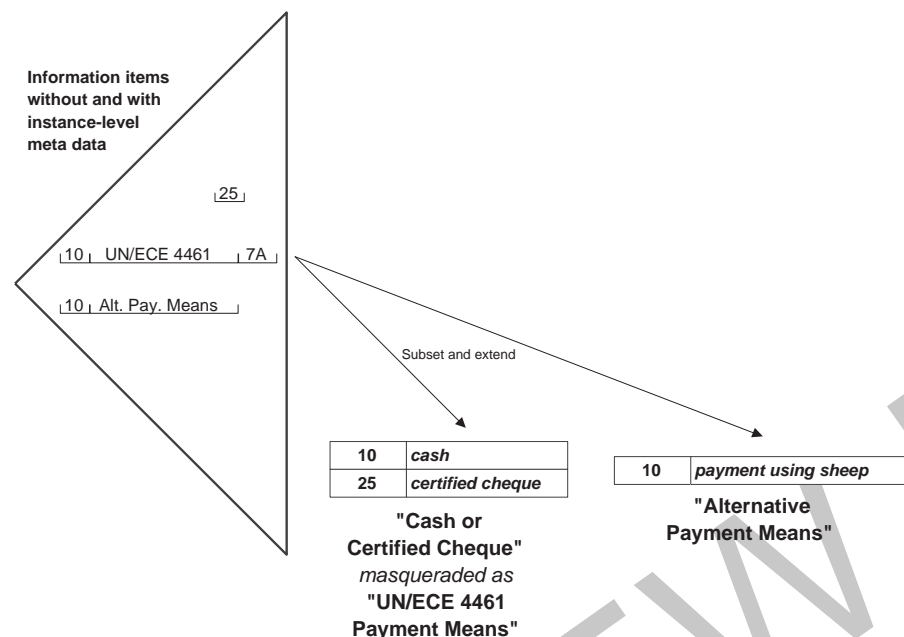
Identifying controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Instance-level meta data qualifies the use of a code

- tells an application the list in which the code is found
- clarifies the meaning



Community responsibility when defining the XML vocabulary

- which instance-level meta data can the user specify?
- how does the user specify instance-level meta data?

Identifying controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Summary of controlled vocabulary meta data

- list-level meta data
 - distinguishes one list of values from another list of values
 - responsibility of the controlled vocabulary custodian
- value-level meta data
 - distinguishes one value from another value within the same list
 - responsibility of the controlled vocabulary custodian
- instance-level meta data
 - distinguishes from which list a value is being used
 - responsibility of the XML vocabulary designer
 - utilized by the XML document writer

Modeling controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



The organization of a set of associated values for all key values is tabular

- one row per semantic concept
- one or more key value columns uniquely identifying the concept
- zero or more meta data value columns defining the concept

The maintenance of list information necessarily needs to be tabular

- such a need distinguishes the available enumeration technologies as to their usefulness
 - i.e. a technology that does not support a tabular arrangement is not as useful as a technology that does support a tabular arrangement of list information

Maintaining an independent expression provides for re-use and change isolation

- the maintenance of a list of values will likely have a different life cycle than the contexts in which the values are used
- revising an external expression of a controlled vocabulary prevents having to change an expression in which a controlled vocabulary is embedded

Human language translations may help as supplemental information

- may reduce problems interpreting what a value represents

Expressing controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Non-standard use of spreadsheets, word processing, comma-separated value (CSV) documents is common

- each custodial organization may have its own way of expressing the representation values and their associated semantics
- applications incorporating the values into their validation processes would need to accommodate ad hoc means with ad hoc measures
- the expression may not be well defined for maintenance

The interchange representation is independent of the internal representation

- though some applications may choose to use the interchange representation as the internal representation
- lookup strategies should be based on application requirements and be independent of interchange requirements

Artefacts for legal contractual agreements may be ambiguous

- using a standardized representation allows both parties to interpret all lists in the same fashion
- prose is often improperly used when meta data may be less ambiguous
 - especially when human language translation is involved

Expressing controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Standards exist with which one enumerates the defined values of a controlled vocabulary

- the choice of expression empowers the use of that expression in different circumstances
- some XML designers fervently believe that document values belong in a document schema
- some XML designers fervently believe that document values must never be in a document schema

W3C Schema enumerations

- designed for use only in validation with W3C Schema semantics
- the formalism only captures the key coded values in a standardized structure
 - the associated meta data may be expressed in a non-standardized structure
 - only one key can ever be used to distinguish the information in the list
- document-wide scope of re-use
 - e.g. every use of the code list incorporates every code of the code list
 - using only a subset of codes requires declaring a separate code list for those contexts where the subset is needed
 - there is a question of what meta data to use for the lists
 - the full list's meta data would be inappropriate for a subset list, yet instances might require the use of the meta data for the full list
- the expressions are intertwined with the expressions of structural constraints
 - to change an enumeration one must "touch" the schema files that participate in structural validation
 - risk of inadvertently modifying the structural constraints, or burden of proving that the structural constraints were not inadvertently modified

OASIS Genericcode 1.0 (2007)

- designed for maintenance of the meta data of an enumeration and its members
- the formalism captures all information about values in a standardized structure
 - when there are multiple keys, the actual key needed can be chosen by an application
 - specifies standardized list-level meta data
 - all controlled vocabularies can be identified using the same mechanisms
 - specifies mechanisms for arbitrary value-level meta data
 - each controlled vocabulary can satisfy its own requirements for value distinction and definition
- context-free scope of use
 - the definition of the codes is independent of the specification of where the codes are used
- the external XML-based expression is independent of any particular use
 - useful for validation or user-interface definition or any use
- still a role for schema specification of available instance-level meta data

Data entry of controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



By formally associating the use of value lists with document contexts, one can direct valid data entry

- directs a user interface
 - can be written to only allow entry of a value from the associated values
- value-level meta data is helpful
 - could be presented to the user to help them choose the right value to use in the data entry
- instance-level meta data records list-level meta data where needed for disambiguation
 - when one information item can have a value from two lists, and a code is needed that exists with the same value in each list, the list-level meta data distinguishes the code and needs to be recorded as instance-level meta data

Application development supporting controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Controlled vocabularies are declarative while application program code is imperative

- easier (therefore cheaper?) to change an outboard declared list of allowed values than to change the inboard program logic
- non-programmer resources can be tasked with changing the declared vocabularies

An application can blindly support all values in a controlled vocabulary

- the application can support all possible allowed values and presume that pre-validation has rejected those instances where a supported value is not allowed
- the flexibility is in the filtering of allowed instances by dynamic application of value constraints during validation, without changing the programming in the application

The trading partner relationship constrains which values are allowed in a given transmission

- message filtering ensures only the messages with the allowed values for a given trading partner are passed for processing

Reduces application development to support new trading partners

- no need to change the program for every trading partner or new trading partners

Flexible to changing trading partner relationships

- as a relationship with a partner matures or changes, only the message filtering need change, not the application code

Validating controlled vocabularies

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Having an expression of valid values enables the validation of specified values

- validation can reject an instance before engaging an application to act on the instance
- off-loads the validation responsibility (and possible error) from applications
- ensures consistent loading of database values

Values outside of the allowed set are considered invalid

- trading partners would not necessarily know what semantic an unexpected value represents
- legal agreements could not be entered into where the parties have arbitrary values possibly representing concepts outside of the agreement

Methodologies are published with which one confirms the proper selection of values in an XML information item

- traditional use of grammar- or type-based document schemas
 - e.g. XML DTD - grammar-based schema language
 - e.g. ISO/IEC 19757-2 RELAX-NG - grammar-based schema language
 - e.g. W3C Schema - type-based schema language
- alternative schema expressions
 - e.g. ISO/IEC 19757-3 Schematron - assertion-based schema language

Traditional approaches validate values at the same time as validating structure

- conflates structural validation with value validation
- inflexible to dynamically changing business requirements
 - no need to change the structural validation just because business relationships change
- business agreements impact on values but do not impact on document structures

Validating controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies

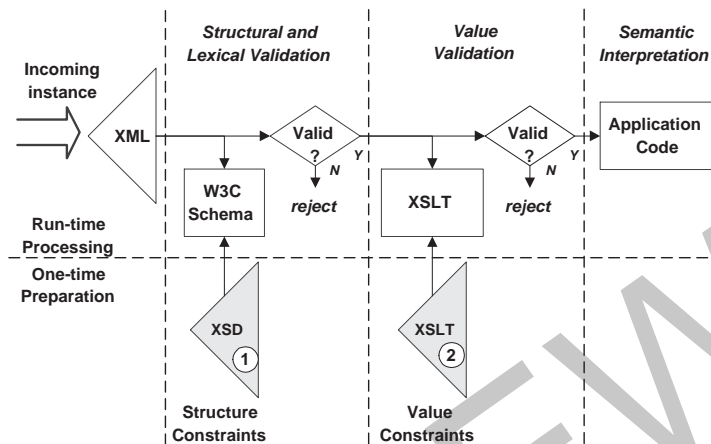


UBL 2.0 separates UBL conformance from code list conformance

- to which version of UBL schemas do the structures conform?
 - e.g. UBL-Invoice-2.0.xsd for an invoice
- to which code lists do the values conform?
 - e.g. defaultCodeList.xsl for a suite of typical code lists

Layering value constraints on top of structural and lexical constraints

- can be used for any XML document structure, not only UBL
- can be applied to any information item with an enumerated set of allowed values
- not restricted to only codes or identifiers
- can be built on top of ISO/IEC 19757-3 Schematron
- separates structural/lexical validation from value validation

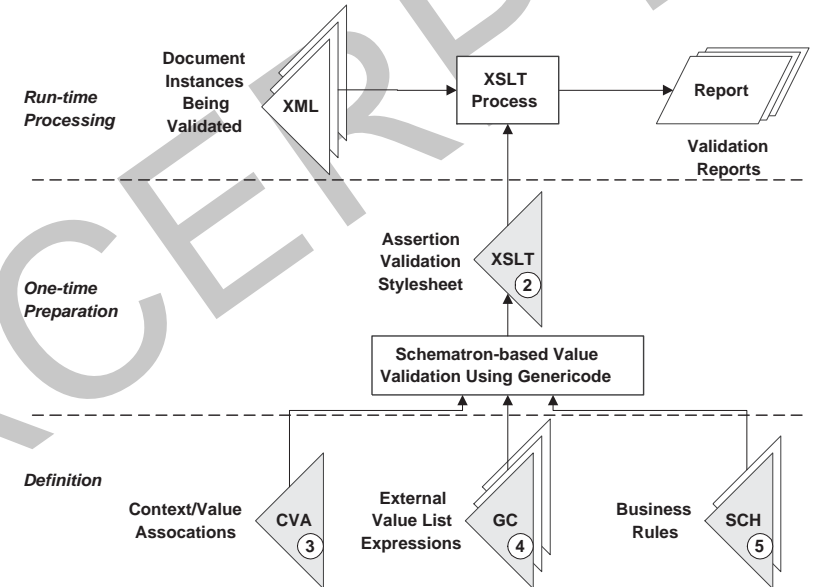


Validating controlled vocabularies (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Opportunity to incorporate many kinds of value constraints



Context/value associations establish which code lists apply where in the document

- gives flexibility to specify different codes for the same conceptual value used in different document contexts

External value list expressions in genericcode

- the XML documents defining the controlled vocabularies
- includes list-level and value-level meta data

Business rules can express co-occurrence and algorithmic constraints

- more powerful than simple declarative approaches
- use Schematron for arbitrary XPath expressions

Semantic representation by fixed values

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Assigned semantics

- each unique value represents an associated concept, label or longer value
- community of users agree on the association between specified value and represented value
- a value has two aspects of context in order to have some meaning
 - context of use/location
 - where in the document is the information item being used
 - context of definition/meaning
 - from which set of values is the information item value being obtained
 - without explicit context a value may be ambiguous or reliant on informal agreement
- especially important for non-mnemonic codes: e.g. "42" vs. "USD"
 - the meaning of some mnemonic codes might be guessed based on context of use, e.g. "USD" for a currency
 - non-mnemonic codes typically have no basis for guessing the meaning, e.g. "42" for a payment means

Instance-level meta data disambiguates a code when context is insufficient

- used for identification of values and definition of values
- list meta data identifies the collection from which the value is taken
 - information about the collection as a whole
 - gives context to the specified values
- value meta data helps define the semantics or details of the value itself
 - information about the one particular coded value

Changes in time can affect the interpretation/semantics of values

- the collection of values evolves creating a new version of an existing code list
 - identified by associated meta data
- the meaning of individual values evolves
 - described by associated meta data or prose
- migrating data from old to new may require simultaneous support of multiple versions of the same code list
 - requires flexibility not typically associated with traditional schema-based approaches to using codes
- unused and retired codes might get re-used later for new semantic concepts
 - e.g. country code "CS" was Czechoslovakia before 2003 and was reserved in 2006 for Serbia and Montenegro

Trading partners and agreements

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Trading partners need to agree on the structure and content of interchanged documents

- so doing provides interoperability between independent systems acting on the information
- using XML isn't a magic bullet
 - it doesn't make our programs better, it makes our interchange of information more reliable
 - layers interchange constraints on top of implementation foundations
- structuring the information in a standardized fashion ensures the information is communicated
 - where to find information based on how it is labeled and where it is found hierarchically
- agreeing on the semantics behind values in the communicated information ensures the information is understood
 - what specified information values mean and represent in the abstract

Relationships between trading partners change, while standards do not

- trading partner requirements can be layered on top of industry standards
- trading partners can also anticipate future changes to standards

Published interchange specifications cannot pretend to know every value that trading partners need

- there is no standardization of many business concepts, just practical and pragmatic use for those concepts of import to trading partners
- therefore that can be no standardization of a set of values representing business concepts that are particular to trading partners

Industries can state what the semantics are behind values

- trading partners can agree on which values to use

Trading partners and agreements (cont.)

Chapter 1 - Controlled vocabularies
Section 1 - Facets of controlled vocabularies



Codes for some established business practices can be supplemented or subset by trading partners

- e.g. extending document status codes
 - a typical workflow will have typical status values for the progress of a document
 - particular workflow systems used by trading partners may use only some or maybe more document status values for a given document
- e.g. restricting payment means codes
 - payment can only be by certified cheque or credit card
- e.g. restricting and extending transportation status codes
 - the suite of status codes includes a subset of a standard suite in combination with non-standard additional values

Identifiers are especially important as they specify actual business objects and not business concepts

- e.g. account identifiers
 - every business will probably have a different set of accounts than other businesses
 - when engaging in a transaction, a trading partner can publish its accepted list of account identifiers so that a correspondent knows which values can be used
- e.g. measurement identifiers
 - a catalogue item's characteristics need to be identified unambiguously (e.g. "gross weight" is distinct from "net weight")

Additional business rules can be layered on top of value constraints

- e.g. validity of non-coded data values based on trading partner relationship
 - e.g. a maximum value for an order
- e.g. the nature of a product identified by its identifier may restrict the payment means by which it is paid for
 - e.g. no credit cards for certain products

Chapter 2 - Defining and using controlled vocabularies



- Introduction - Controlled value list maintenance and identity

Controlled value list maintenance and identity

Introduction - Chapter 2 - Defining and using controlled vocabularies



A list of values has an identity in the abstract, regardless of how it is maintained

- e.g. ISO 3166-1 country codes
- e.g. UN/ECE 4461 payment means codes
- e.g. UBL 2.0 document status codes

The complete list may be maintained by hand or by a database or by any means

- the management of the values is important to long-term maintenance
- some lists may have tens of thousands of entries (e.g. vehicle model codes)
- a list may be synthesized by an algorithmic process
 - e.g. the 100 ISBN numbers assigned to publisher "978-1-894049"

The concrete expressions of the lists may vary based on purpose or contextual use

- e.g. complete lists
- e.g. restricted subset lists
- each list and list subset expression must necessarily be uniquely identified
 - a subset of a list cannot have the same identity as the complete list otherwise there would be confusion regarding which list is the "true" list
 - identity can be expressed as meta data for the list or list subset
- the concrete expression may take any useful form for the user
 - e.g. simple text
 - e.g. comma-separated values
 - e.g. XML files
 - having a standardized representation of lists would encourage the development of more widely-useful applications

The sender and receiver may have different identities for a list of identical values

- e.g. the sender specifies an ISO country code
 - the meta data for the list is that of the complete list
- e.g. the receiver only accepts a subset of ISO country codes as valid
 - the meta data for the subset list is necessarily different than that of the complete list
 - for validation purposes the subset list must masquerade as the complete list yet reject specified values outside of the subset list

Controlled value specification

Introduction - Chapter 2 - Defining and using controlled vocabularies



A controlled value is, in fact, a multi-faceted value

- the list from which the code value is obtained
 - described by meta data for the list
 - the list identification itself may be multi-faceted
- the key code value itself
 - unique within any given list
- properties (value-level meta data) of the values themselves
 - helpful in understanding the semantics of the key code value

When an information item can be populated with a coded value, it should also be possible to specify the associated value list meta data

- even very stable lists of values will change over time
 - one may need to specify a chronologically-distinct interpretation of a given value
 - e.g. the list of provinces in Canada changed in 1999 when the Northwest Territories was split into two territories: Nunavut and the Northwest Territories
 - the Canadian postal province and territories indication of the Northwest Territories was and remains "NT" even though the definition of the territory changed
 - if the distinction is important to a trading partner, then provision for making the distinction must be made available
- e.g. the list of country codes changes frequently
 - before 2003 "CS" represented Czechoslovakia and since 2006 "CS" is reserved for Serbia and Montenegro
- one information item value may be an item selected from one of a number of lists
 - if all of the values are mutually exclusive in separate lists, there is no risk of confusion other than changes over time for any given list as noted above
 - if the values in the lists are not mutually exclusive, meta data is required to disambiguate an ambiguous specified value

The risk is borne by the party encoding the information that the recipient can properly decode the intent expressed by the information

- list meta data is often optional and is often ignored when coded values are specified
- the more specific a specification is, the less opportunity for improper understanding of the intended meaning

Chapter 3 - Declaring controlled vocabularies



- Introduction - Declaring controlled vocabularies

Outcomes

- consider the different ways of declaring the enumerated values of controlled vocabularies

Declaring controlled vocabularies

Introduction - Chapter 3 - Declaring controlled vocabularies



Standards are in development for the non-schema-based representation of a list of coded values

- trading partners may wish to trim or augment the list of coded values acceptable
- trading partners may wish to use different controlled vocabularies for a given information item found in different document contexts
- the representation of individual coded values includes documentary information and metadata
 - for detailed value description
 - for long-term maintenance and understanding
- OASIS genericcode 1.0
 - <http://docs.oasis-open.org/codelist/genericcode>
 - an XML representation standardized by the OASIS Code List Representation Technical Committee
 - <http://www.oasis-open.org/committees/codelist/>
 - "Defining an XML format for interchange, documentation and management of code lists (a.k.a. controlled vocabularies or coded value enumerations) in any processing context"
 - not obliged to use XML format *inside* the application
 - very common to compile the XML interchange format into an internal processing format
 - e.g. conversion to XSLT
 - e.g. implementation in database stored procedures
 - XML is designed for interchange and is not always conveniently structured for real-time processing

One could use schema enumerations but ...

- too inflexible for globally-defined information items
 - cannot have different sets of values in different document contexts for a globally-defined information item
- modifying the schemas means using non-standardized schema expressions
 - not bad in and of itself but requires extra assurances for compatibility
 - structural and lexical validation is assured if the standardized schema expressions are treated as read-only

Meta-data-only code list are important as placeholders

- effectively an infinite set of all possible codes satisfying the lexical rules
- indicating that a particular information item's value is from a controlled vocabulary but that there is no controlled vocabulary listing a set of codes
- e.g. only 18 of 91 UBL code lists are published with values, 73 uniquely-categorized code lists have only meta data
- users have the option of restricting the infinite list into a finite list

Chapter 4 - Controlled vocabulary representation detail



- Introduction - Controlled vocabulary representation detail

Controlled vocabulary representation detail

Introduction - Chapter 4 - Controlled vocabulary representation detail



Genericode

- an XML vocabulary for the representation of an enumerated set of values
- <http://docs.oasis-open.org/codelist/genericode/xsd/genericode.xsd>
 - base vocabulary for the genericode namespace
- <http://docs.oasis-open.org/codelist/genericode/xsd/xml.xsd>
 - imported vocabulary for the XML namespace in support of `xml:lang=`

Meta data identification of an enumerated set of values

- information regarding identifying the list as a whole
 - names by which the list is known
 - version of the particular list
 - resource identifiers with which to retrieve the list contents
- information regarding identifying the custodian
 - agency
- information regarding each individual value

Many features for supporting sets of code lists or derived code lists

- Schematron-based context/value association stylesheets as delivered support only simple single-key code lists in standalone declarations
 - interpreting multiple keys or a set of sets or a derivation of sets is not supported
- this documentation is limited to the available document types and to the supported constructs only of simple single-key standalone declarations
 - this documentation is not meant to be exhaustive documentation for genericode files
 - please see the OASIS code list representation committee deliverables for more details on genericode files

Chapter 5 - Associating controlled vocabularies in XML documents



- Introduction - Constraining information items using controlled vocabularies

Outcomes

- consider the requirements of associating information items defined by controlled vocabularies

Constraining information items using controlled vocabularies

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Three kinds of constraints to be validated for an XML document

- structural constraints ensure information items are correctly found
- lexical constraints ensure information items are correctly formed
- value constraints ensure information items are correctly understood

Constraining the document structure and lexical patterns is independent of business/value rules

- a community of users can publish an agreed upon schema to validate information items are correctly found and formed

Constraining information item use of controlled vocabularies is very dependent on business/value rules

- business/value rules implied by the nature of the information item
 - e.g. points of a compass will never change
- business/value rules imposed by a community of users
 - e.g. the document status codes for the condition of a document in a transaction
- business/value rules agreed upon between trading partners
 - e.g. identification of account numbers for particular purposes

Typical use of W3C Schema conflates structural and value constraints inflexibly

- one gets more flexibility by separating value constraints from structural constraints
- only structural constraints should be imposed across a community of users
 - standard should constrain how the information is found and how it is formed, not how it is valued
 - very infrequent changes to the structure of information being interchanged
 - changes imply big impacts on applications and processing
- value constraints should be selectively imposed
 - changes in trading partners
 - changes in business practices over time
 - possibly frequent changes to the values allowed by different parties
 - once programs accommodate a given set of values, changing the subsets of values in use doesn't change the applications
- business rules should be selectively added
 - private requirements could never be anticipated by standards committees

Context/value association

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Context/value association files

- http://www.oasis-open.org/committees/document.php?document_id=29990
- an XML vocabulary for associating document contexts with specified values
- suitable for constraining document entry in a user interface
- suitable for document validation before application processing
- techniques for specifying, restricting and extending lists for the purposes of validation

Masquerading meta data when restricting a large list to a subset of values

- the validation needs to match an instance's use of large list meta data to a declaration of a subset list using subset list list-level meta data
- the subset list list-level meta data necessarily is different than the list-level meta data of the list from which it is derived
- the subset list masquerades as the list from which it is derived so that instance-level meta data doesn't use the custom subset list list-level meta data

ISO/IEC 19757-3 Schematron deployment

- as supplied, the methodology reports context/value constraint violations in simple text
- Schematron can alternatively be deployed with different available reporting techniques

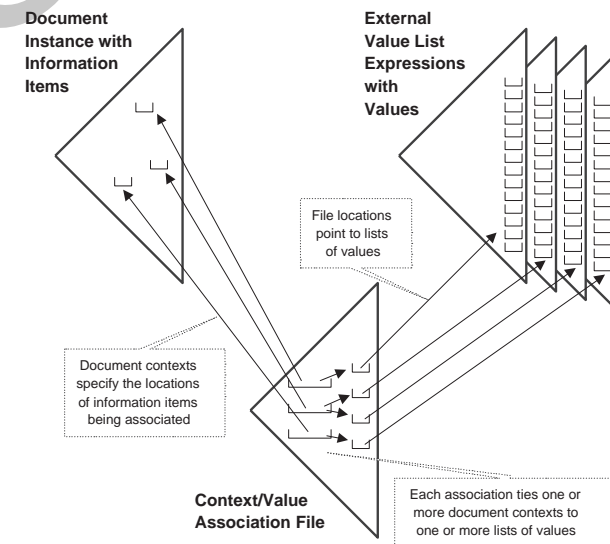
Context/value association (cont.)

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



The principles of context/value association are as follows:

- XML documents have information items that need to be validated
 - the locations (contexts) of those items can be addressed using XPath addresses
- genericcode files have values and list meta data to use for validation
 - the locations of those files can be declared with URL addresses
 - the identity of each list is uniquely specified in order to be referenced multiple times
- an association marries a document context with a set of genericcode files
 - each XPath document context is specified with the identities of the genericcode declarations
- validation checks values found in document contexts against genericcode files linked by the association for the document context
 - any present meta data in the document context is checked with the available genericcode meta data



Context/value association (cont.)

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Appropriate for constraining data entry application user interfaces

- used as a front end to a user preventing the data entry of different values
 - drop-down lists
 - radio buttons
 - check boxes
- the end result of editing an instance is that the values are all from the associated lists
- the value-level meta data can be presented to the user
 - assists the user in choosing which value or values to use
- the options to include instance-level meta data should be offered
 - reflects the list-level meta data for the list from where the values are taken

Appropriate for constraining data validation

- used as a front end to an application that implements the logic for all possible values
- selective association for business scenarios prevents the application from acting on inappropriate values for a given transaction
 - relationships between specific partners may be different
 - different profiles of using documents may constrain particular values

Only the CVA vocabulary is standardized by OASIS, not how it is used

- the file format and the semantics represented by the elements and attributes are being standardized by OASIS
- any implementation is considered out of scope of the committee work

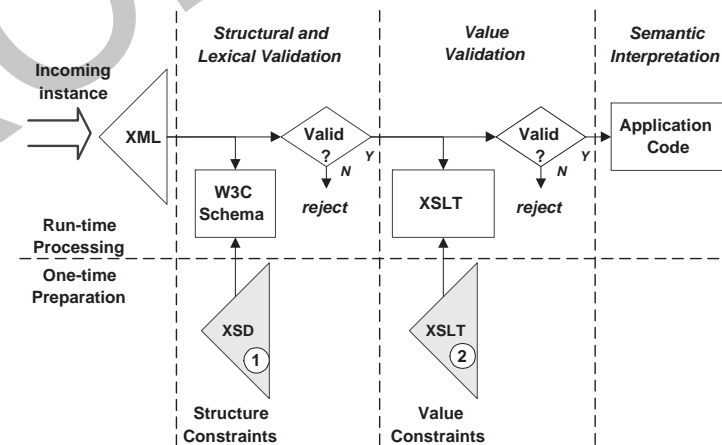
Using context/value association for validation

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Separates structural/lexical validation from value validation

- an XML document is checked using a two-step process
- the first pass for structural and lexical validation passes
- the second pass reports that a coded value used for a currency is unexpected
- the document structure and lexical content can be constrained by standardization
 - e.g. the UBL technical committee publishes normative W3C schemas
- the document controlled-value content is constrained by business requirements between trading partners
 - e.g. the UBL committee publishes default coded value checks
 - defaultCodeList.xsl
- trading partners can use this value validation methodology to create their own value checking second-pass process



Document arrives at application unchanged

- validation only confirms the use of structure and content, without modifying it

Second pass results meaningless without first pass being successful

- the values must be correctly found and correctly formed before checking the actual values produces an accurate result

Using context/value association for validation (cont.)

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Crane-CVA2sch package from Crane Softwrights Ltd. web site

- historically developed in the OASIS UBL Technical Committee
- moved into the OASIS Code List Representation Technical Committee
- moved out of the OASIS Code List Representation Technical Committee
 - the committee decided to focus on file formats and not methodologies
 - intellectual property returned to Crane Softwrights Ltd.
- Crane is donating CVA2sch to an Apache Schematron project

A methodology for code list and value validation based on ISO/IEC 19757-3 Schematron

- an information item is asserted to have one of an allowed set of predetermined values
 - a failed assertion is a value validation error
- assertions are derived from context/value associations

Schematron is usually implemented using the Extensible Stylesheet Language (XSLT)

- the supplied Schematron stylesheet for stylesheets is a copy of the publicly-available reference XSLT implementation
 - <http://www.schematron.com>
- the methodology supplies a wrapper stylesheet for the reference skeleton
- other non-XSLT implementations of Schematron exist
 - e.g. Amara/Scimitar implements ISO Schematron in Python
 - <http://uche.ogbuji.net:8080/uche.ogbuji.net/tech/4Suite/amara/>
 - same architecture as reference XSLT implementation in that Scimitar is a Python program that writes a Python program that performs the validation

The XSLT generated to implement the Schematron assertions is used as the second pass of validation to test XML instances for having correct controlled-vocabulary values

- the testing relies on the first-pass structural validation, having already confirmed the structure and lexical values used in the instance
- without the first pass confirming the accurate presence of information items, the second pass is meaningless

The methodology supports the incorporation of any number of sets of Schematron assertions

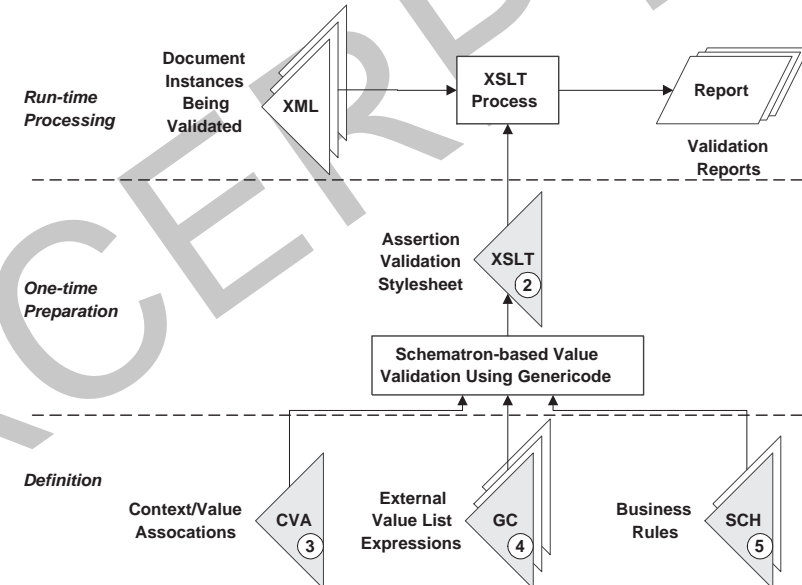
- ISO Schematron supports the inclusion of multiple schema fragments into a single schema expression
- business rules related or unrelated to code lists may be expressed as Schematron assertions
 - the trading partner schema can then include business rules in addition to coded value rules

Using context/value association for validation (cont.)

Introduction - Chapter 5 - Associating controlled vocabularies in XML documents



Overview of the process to prepare the second pass value validation XSLT stylesheet:



- the circled labels in the diagram are indicated by the parenthesized numbers
- the inputs:
 - (3) the specification of contexts uses the context/value association XML vocabulary defined by the OASIS Code List Representation TC
 - (4) the specification of coded values uses the genericcode vocabulary defined by the OASIS Code List Representation TC
 - (5) supplemental business rules are specified using ISO/IEC 19757-3 Schematron
- the output:
 - (2) an XSLT stylesheet (or some other implementation of Schematron assertion checking)

Recall Validating controlled vocabularies (page 24)

- the XSLT created here (2) plugs in to the two-step validation process

Recall Context/value association (page 39)

- all three documents on that diagram are shown here as instances being validated, the context value association files and the external value list expressions

Chapter 6 - Controlled vocabulary association detail



-
- Introduction - Controlled vocabulary association detail

Controlled vocabulary association detail

Introduction - Chapter 6 - Controlled vocabulary association detail



CVA files are validated using an W3C Schema expression of constraints

- /xsd/ContextValueAssociation.xsd
 - includes /xsd/xml.xsd

Development work still in progress

- the work is continuing by the OASIS technical committee
- needs more community input and practice before becoming version 1.0

Chapter 7 - Your own business document controlled vocabulary



- Introduction - Supporting your own business documents with context/value association files

Supporting your own business documents with context/value association files

Introduction - Chapter 7 - Your own business document controlled vocabulary



the CVA2sch validation stylesheets are delivered pre-configured for use with UBL

- genericode for the value list enumerations
- UBL conventions for the information item names
- UN/CEFACT CCTS conventions for the information item instance-level meta data

A "no metadata" configuration of the stylesheets is included with the methodology

- an off-the-shelf implementation of the methodology for business documents without information item meta data
- checks the information items identified in the CVA file without checking any instance-level meta data

Adaptation to other value list enumeration XML vocabularies

- a methodology stylesheet can be replaced with support for an arbitrary outboard representation of sets of enumerated values

Adaptation to other business document XML vocabulary instance-level meta data

- a methodology stylesheet can be replaced with support for arbitrary information item meta data

All methodology stylesheets are written in XSLT 1.0 for portability

- adaptation requires writing new stylesheet fragments to replace existing stylesheet fragments
- the existing stylesheet fragments are likely to be very useful as a model for the replacement
- XSLT 1 information item matching is only based on the namespace-qualified name
- XSLT 2.0 could be used for more nuanced item matching
 - schema-aware XSLT 2.0 could be used for type-based matching

Annex A - OpenOffice 3 genericode and CVA filters



-
- Section 1 - OpenOffice 3 genericode and CVA filters

OpenOffice 3 genericode and CVA filters

Annex A - OpenOffice 3 genericode and CVA filters
Section 1 - OpenOffice 3 genericode and CVA filters



Crane Softwrights Ltd.'s `gc2ods` package is a pair of OpenOffice 3 XML filters

- enables OpenOffice 3 to read and write genericode and context/value association files
- the user interface is presented in a spreadsheet document across multiple sheets

The package is found as a ZIP file linked from the sales page for this book

- <http://www.CraneSoftwrights.com/sales/pcli/>
- your book purchase password is needed to get access to the package

The `readme.html` documentation includes all of the necessary documentation to install, uninstall and use these filters.

OpenOffice 3 genericode and CVA filters (cont.)

Annex A - OpenOffice 3 genericode and CVA filters
Section 1 - OpenOffice 3 genericode and CVA filters



There are four tabs when editing genericode files:

- Identification
 - managing list-level meta data
- Columns
 - managing the kinds of value-level meta data
- Values
 - managing the actual codes and their associated value-level meta data values
- Help
 - context-sensitive help information

There are four tabs when editing context/value association files:

- Identification
 - managing association-wide meta data
- Values
 - managing the external value lists expressed in genericode
- Contexts
 - managing the document context items and their associated values
- Help
 - context-sensitive help information

Where to go from here?

Conclusion - Practical Code List Implementation



The work on genericode and context/value association continues:

- OASIS Genericode 1.0 Committee Specification - December 27, 2007
 - <http://docs.oasis-open.org/codelist/genericode>
- OASIS Context/value Association Specification - 0.5 draft 1
 - http://www.oasis-open.org/committees/document.php?document_id=29990
- focus now shifts to association, support, deployment, awareness and evangelism
- committee mail list - OASIS Code List Representation TC:
 - <http://lists.oasis-open.org/archives/codelist/>
- community mail list - CLR-Dev
 - <http://lists.oasis-open.org/archives/clr-dev/>
 - <http://www.oasis-open.org/mlmanage/>

Colophon

Conclusion - Practical Code List Implementation



These materials were produced using structured information technologies as follows:

- authored source materials
 - content in numerous XML files maintained as external general entities for a complete prose book that can be made into a subset for training
 - specification of applicability of constructs for each configuration
 - 45- and 90-minute lecture, half-, full-, two- and three-day lecture and hands-on instruction, and book (prose) configurations
 - an XSLT transformation creates the subset of effective constructs from applying applicability to the complete file
 - content from other presentations/tutorials included semantically (not syntactically) during construct assembly
 - customized appearance engaged with marked sections and both parameter and general entities
 - different host company logos and venue and date marginalia
 - changing a single external parameter entity to a key file includes suite of files for given appearance
- accessible rendition in HTML
 - an XSLT stylesheet produces a collection of HTML files using Saxon for multiple file output
 - mono-spaced fonts and list-depth notation conventions assist the comprehension of the material when using screen-reader software
- printed handout deliverables
 - an XSLT stylesheet produces an instance of XSL formatting objects (XSL-FO) for rendering
 - XPDF <http://www.foolabs.com/xpdf> extracts raw text from PDF files for the back-of-the-book index methodology published as a free resource by Crane Softwrights Ltd.
 - XEP by RenderX <http://www.renderx.com> produces PostScript from XSL-FO
 - GhostScript <http://www.ghostscript.com> produces PDF from PostScript
 - the iText <http://itext.sf.net> PDF manipulation library for Java is used for page imposition by a custom Python <http://www.python.org> program running under the Jython <http://www.jython.org> environment

Obtaining a copy of this material

Conclusion - Practical Code List Implementation



This comprehensive tutorial on code lists in XML is available for subscription purchase and free preview download:

- "Practical Code List Implementation" First Edition - 2009-02-09 - ISBN 978-1-894049-22-1
 - the free download preview excerpt of the publication includes the complete text of the first chapter and the introductory text of all of the other chapters
- the cost of purchase includes all future updates to the materials with email notification
 - the materials are updated after new content developed
 - more frequent in earlier editions than later editions
 - the materials are updated after incorporating comments gleaned during presentations and from feedback from customers
- available in PDF
 - formatted as 1-up or 2-up book pages per imaged page
 - dimensions in either US-letter or A4 page sizes
 - available as either single sided or double sided
- accessible rendition available for use with screen readers
- site-wide and world-wide staff licenses (one-time fee) are available

See <http://www.CraneSoftwrights.com/links/trn-20090209.htm> for more details.

Software available to customers

- accompanying software is available at no charge to customers of this book
- the license for this free software does not allow for free distribution of the software to others
- free download from <http://www.CraneSoftwrights.com/sales/pcli/> to registered users

Feedback

- the unorthodox style has been well-accepted by customers as an efficient learning presentation
- feedback from customers is important to improve or repair the content for future editions
- please send suggestions or comments (positive or negative) to info@CraneSoftwrights.com



Practical Code List Implementation (Using Controlled Vocabularies in XML Documents)

Crane Softwrights Ltd.
<http://www.CraneSoftwrights.com>



Table of contents

Indexed by slide number

1	[Prelude] Practical Code List Implementation (2) (3)
4	[Overview] Practical Code List Implementation
5	[Introduction I-1] Practical Code List Implementation
6	[1] Controlled vocabularies
7	[Introduction 1-I-1] XML document interchange
8	[Introduction 1-I-2] Controlled vocabulary semantics (9) (10)
11	[1-1-1] Codes and identifiers
12	[1-1-2] Code list registration authorities
13	[1-1-3] Identifying controlled vocabularies (14) (15) (16) (17)
18	[1-1-4] Modeling controlled vocabularies
19	[1-1-5] Expressing controlled vocabularies (20)
21	[1-1-6] Data entry of controlled vocabularies
22	[1-1-7] Application development supporting controlled vocabularies
23	[1-1-8] Validating controlled vocabularies (24) (25)
26	[1-1-9] Semantic representation by fixed values
27	[1-1-10] Trading partners and agreements (28)
29	[2] Defining and using controlled vocabularies
30	[Introduction 2-I-1] Controlled value list maintenance and identity
31	[Introduction 2-I-2] Controlled value specification
32	[3] Declaring controlled vocabularies
33	[Introduction 3-I-1] Declaring controlled vocabularies
34	[4] Controlled vocabulary representation detail
35	[Introduction 4-I-1] Controlled vocabulary representation detail
36	[5] Associating controlled vocabularies in XML documents
37	[Introduction 5-I-1] Constraining information items using controlled vocabularies
38	[Introduction 5-I-2] Context/value association (39) (40)
41	[Introduction 5-I-3] Using context/value association for validation (42) (43)
44	[6] Controlled vocabulary association detail
45	[Introduction 6-I-1] Controlled vocabulary association detail
46	[7] Your own business document controlled vocabulary
47	[Introduction 7-I-1] Supporting your own business documents with context/value association files
48	[A] OpenOffice 3 genericcode and CVA filters
49	[A-1-1] OpenOffice 3 genericcode and CVA filters (50)
51	[Conclusion C-1] Where to go from here?
52	[Conclusion C-2] Colophon
53	[Conclusion C-3] Obtaining a copy of this material
54	[Postlude] Practical Code List Implementation

Index



A
 account identifiers 28
 algorithmic 11
 application 8, 22, 33, 41

B
 business rules 25, 28, 37, 43

C
 characteristic 11
 code 9, 10, 29
 colloquial XML vocabulary 12
 comma separated values (CSV) 12, 19, 30
 community 16
 constraints 24
 structural constraints 37
 value constraints 37
 container size code list 11, 12
 context (document) 26
 context/value association 25, 36, 44
 controlled vocabulary 11
 Core Component Technical Specification (CCTS) 47
 country codes 8, 10, 13, 30
 Crane-gc2ods 3, 5, 48
 currency codes 8, 10, 13, 26
 custodian 12

D
 data entry 21, 40
 database 9, 10, 12, 23, 30, 33
 defaultCodeList.xsl 24, 41
 document status codes 28, 30
 Document Type Definition (DTD) 23

E
 Extensible Stylesheet Language Transformations (XSLT) 33, 42, 43, 47

F

G
 genericode 20, 25, 33, 34, 39

H
 hierarchies in an XML document 7

I
 identifier 29

interchange 19, 27
 International Organization for Standardization (ISO) 8, 9
 International Standard Book Numbering (ISBN) 12, 30

J

K
 key code 9, 18, 20, 31

L
 language 10
 legend 2
 lookup value 11

M
 masquerade 38
 measurement identifiers 28
 meta data
 instance-level 16, 17, 20, 21, 26, 40, 47
 list-level 9, 14, 17, 20, 26, 35, 40
 meta-data-only code list 33
 value-level 9, 10, 15, 17, 20, 21, 26, 31, 35, 40

N
 name 9

O
 OpenOffice 3 48

P
 payment means codes 8, 10, 13, 26, 28, 30
 Python 42

Q

R
 RELAX-NG (Regular Language for XML) 23

S
 sample code fragments 2
 Schematron 23, 24, 25, 35, 38, 42, 43
 scope 20
 document-wide scope 20
 semantics (meaning) 10, 13, 18, 26, 27, 31

T
 trading partners 8, 27, 28, 37
 agreements/contracts 8, 11, 19
 relationship 22, 40
 translations 10, 18
 transportation status codes 28
 typographical conventions 2

U
 UN/CEFACT 8
 unconstrained code lists 13
 union
 of code lists 21, 31
 uniqueness 9, 31
 unit of measure codes 8
 Universal Business Language (UBL) 11, 12, 24, 41, 47

user interface 21

V
 validation 23, 24, 25, 39, 40, 41, 42, 43
 vocabulary, controlled 7
 semantics 8, 9, 10
 vocabulary, XML 7

W
 W3C Schema 8, 12, 20, 23, 35, 37, 41, 45

X
 XML Path Language (XPath) 25, 39

Y

Z