Practical Universal Business Language Deployment

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

+//ISBN 1-894049::CSL::Presentation::UBL//DOCUMENT Practical Universal Business Language Deployment 2009-02-12 13:50UTC//EN
Third Edition - 2009-02-12
ISBN 978-1-894049-23-8
http://www.CraneSoftwrights.com
Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment

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

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`

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.

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment (Prelude) (cont.)

Preface

This PDF publication is a living document

- the purchase of any edition entitles the purchaser to all future editions at no charge
- the plan is to publish new editions often
    - practices evolve and mature when working with new specifications
    - training and community mail lists will reveal the need to explain sections of the material differently or more thoroughly
    - customers of the book request new or expounded information in a new edition
- the paper rights for this publication are still available for an interested publisher
- this publishing model was honed on two earlier Crane books for XSLT and XSL-FO and has been well received by customers of both PDF and paper editions

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

- derivatives of the book are used for instructor-led training, requiring the succinct presentation
    - note the exercises used in instructor-led training sessions are not included here
- *this PDF book is accompanied by a ZIP file* with supplementary files
    - the number of supplementary files will grow with future editions
- derivatives 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 projected during the training

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment (Prelude) (cont.)

Much of the content is *hyperlinked both internally and externally* to the book in the 1-up full-page sized electronic renditions (not in the half-page renditions):

- note when using Acrobat Reader, the history "back" keystroke sequence is "Ctrl-Left"
- page references, e.g.: Chapter 5 - Documents and document models (page 96)
    - click on the title or page number above to test!
    - the back-of the book index is hyperlinked to the body of the book
    - the letter references at the bottom of each page are hyperlinked to the index
- references to sections of the specification are in parentheses, e.g.: (4.2.1.)
    - the section number linked in full-sized pages to the online specification
    - click on the "4.2.1." above to test!
- external references are in monospaced text (click on the following to test!)
    - e.g.: `http://docs.oasis-open.org/ubl/os-UBL-2.0/`
- chapter references in book summary
- section references in chapter summary
- subsection references in table of contents at the back of the book
- no hyperlinks are 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

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment (Prelude) (cont.)

Important caveat regarding the information in this publication

- while the author, G. Ken Holman, is a co-editor of the UBL 2.0 specification, not all of the material in this publication is necessarily accepted by all UBL Technical Committee members
- 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 UBL TC
- 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 UBL TC is debating the use of procedures, processes and data files in support of UBL
- 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 UBL 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/publd/`
- 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

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment

Series: Practical Universal Business Language Deployment

Reference: Electronic commerce

Outcome

- detailed review of the components of the Universal Business Language deliverables and supplementary packages

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment

Introduction - Practical Universal Business Language Deployment

The first and oldest documents created by mankind were business documents:

- early Bronze Age Sumerians invented writing from commercial inscriptions (3300BC)
  - transactions, inventories, etc.
- commerce and trade developed to fund the war machine to overcome invading empires

Paper business documents are the norm in commerce today

- most are ad-hoc presentations of purchase orders, invoices, etc. created by companies and software providers without regard to any standard layout or presentation
- paper layouts of business documents have been internationally-standardized
  - (cited from UNCTAD Trust Fund for Trade Facilitation Negotiations Technical Note No.13)
    `http://r0.unctad.org/ttl/technical-notes/TN13_Document%20Simplification.pdf`
  - the UN Layout Key (UNLK) was first adopted in 1963
    - became UNECE recommendation No 1 in 1978
  - a master layout design from which other trade documents can be derived
  - organizes coded information (address, buyer, seller, documentation requirements for certain products, etc) in a box format in fixed locations on a document

Electronic business documents are available in commerce today

- most are ad-hoc representations of purchase orders, invoices, etc. created by companies and software providers without regard to any standard structure or content
- the Electronic Data Interchange (EDI) standard has long been in use by large companies
  - complex to deploy
    - cryptic machine-based representations of the information suitable for computers but not people
    - stovepipe implementations of EDI by different industries inhibits interoperability and message compatibility
  - expensive to operate and out of the reach of small- and medium-sized enterprises
    - EDI service providers often require a minimum utilization that far exceeds the needs of small business
- the OASIS Universal Business Language (UBL) has been internationally-standardized
  - the first royalty-free specification of XML-based business documents and business object library
  - a mature version 2.0 specification based on real-world experience with version 1.0
  - easy-to-digest XML format with human-legible labels on hierarchically-organized structures of information
  - well-defined approaches to customization and augmentation to meet tailored requirements for communities of users

Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment (cont.)

This book takes the reader through the UBL 2.0 specification deliverables and artefacts

- the only normative component of UBL 2.0 is a set of W3C Schema XSD expressions of constraints on classes of XML documents
- the delivery package includes a number of supporting materials based upon the XSD normative document models

Not everything in this book has yet been adopted or agreed to by the UBL Technical Committee

- this is a book by and from the perspective of Crane Softwrights Ltd.
- G. Ken Holman is an active member participating on the UBL Technical Committee and a co-editor of the UBL 2.0 specification
- some of the materials described in this book are Ken's member submissions to the TC for consideration by the membership

Emerging methodologies and deployment approaches are described

- UBL 2.0 offers far more features than UBL 1.0 to address real-world requirements discovered in the deployment of UBL 1.0
- Crane and other vendors and volunteers are making resources available to the user community
    - Crane has a number of UBL-related resources available in the "Free resources" section of the web site linked from the right-hand marginalia of:
        - `http://www.CraneSoftwrights.com/links/trn-20090212.htm`

Important caveat regarding the objectives of the material

- this is a book describing the available UBL 2.0 materials and technical implementation issues for deploying the specification
- this is *not* a book with which to interpret specific UBL information items in a business context or to assess the adaptation of financial systems to the semantics of UBL
- this book *cannot* be interpreted as providing financial or legal advice regarding the application or suitability of UBL to any particular scenario or purpose

This book will evolve through multiple future editions

- based on feedback from students taking training derived from this material
- based on UBL TC developments and decisions and publications
- based on Crane's own deployment experience of using UBL
- based on Crane's consulting customers' experience of using UBL
- based on publicly-available case studies of the successful use of UBL 2.0
- in particular the chapters on customization will be embellished
- Crane's other electronic books began with many early editions before quieting down to less-frequent free updates as the content matured

Enjoy!

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 1 - OASIS Universal Business Language (UBL)

- Section 1 - OASIS Universal Business Language (UBL)
- Section 2 - ebXML context
- Section 3 - UBL applicability
- Section 4 - UBL 2.0 specification contents

Outcomes

- gain an overview of the UBL specification
- be aware of the standardization committee structure
- understand the role of UBL in the ebXML context
- understand the expectations of applicability for UBL and what it tries to accomplish
- be aware of some existing deployments of UBL and thoughts for the future of UBL

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL)

UBL is an international effort to develop open royalty-free standards for the machine-processing of business information

- the UBL committee has worldwide committee membership
  - vendor members
  - consultant and trainer members
  - business expert and user members
- there are worldwide deployments active or in the works
- the intellectual property is owned by the Organization for the Advancement of Structured Information System (OASIS)
  - `http://www.oasis-open.org/who/intellectualproperty.php`
  - UBL is licensed under "RF on Limited Terms"
    - limiting the contributors' IP obligations, not the users' ability to use
    - as a requirement of membership all developers and participants in the technical committees transfer the intellectual property of their contributions to the organization
  - `http://lists.oasis-open.org/archives/ubl/200610/msg00047.html`
  - the intellectual property is not tied up by any one vendor or user

Entry point for e-commerce for small- and medium-sized businesses

- inevitably to be used by large businesses as well due to mandated requirements by large users and governments

Supplants the need to use existing or develop one's own proprietary electronic format

- proprietary software is under the control of a vendor or the software developer
- proprietary formats have limited (if any) interoperability with systems created by other vendors
- no leverage of available implementations for re-use

Opportunity to move quickly from business process modeling to active messaging

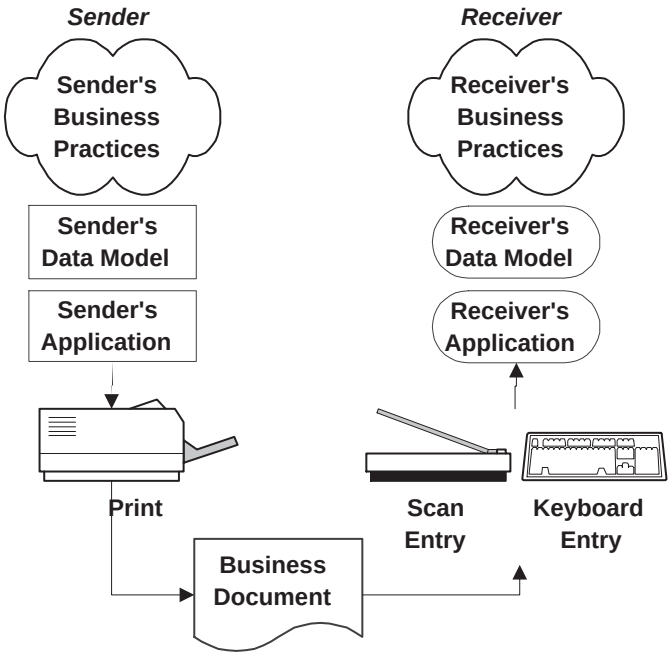
- use your own methods to determine your own business processes
- when done map your information model components to off-the-shelf UBL business objects
- utilize off-the-shelf UBL document models for information interchange in the modeled business process
- customization approaches provide for unique requirements

Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL) (cont.)

UBL is designed to eliminate re-keying of data

- supplant existing fax- and paper-based supply chains



The sender prepares a business document for sending:

- the information components of the sender's business practices are expressed in a data model supporting the specific business practices of the sender
- the sender's application accesses the information in the data model and prints off the business document in hard copy

The receiver receives a business document:

- the document information components are scanned or manually entered into the receiver's application, running a risk of a possible scanning or entry error
- the receiver's application stores the information components in the receiver's data model
- the receiver's business practices act on the presence of expected information found in the data model

The sender's and receiver's business practices are probably very similar, though they do not have to be

- the data models could be quite different in structure, but probably have similar components
- the applications could be very different, on different platforms and obtained from different vendors

Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL) (cont.)

The basis of UBL is the use of the Extensible Markup Language (XML)

- `http://www.w3.org/TR/xml`
- web standard for the representation of hierarchically-structured text-based information
- platform-, application- and vendor-independent standard based on the use of international Unicode characters
- widespread and growing support in development tools and developer skills

Contrast the XML format with the commonly-known CSV format

- Comma separated values (CSV) is a commonly-used expression of information
  - no declaration of the character set encoding of the characters in the file
  - information maintained in a set of flat comma-delimited text records
  - no concepts of data types for the declarative lexical validation of values
  - labeling of position-oriented fields optional based on presence of first line of labels
  - all labels are unique without the use of context to distinguish similar items
- XML is a richly-featured specification of an expression of information
  - provides for specification of a character encoding of Unicode characters
  - information is well organized in a strict hierarchy under a single "top" element (called the "document element") of an upside down tree structure
  - W3C Schema XSD 1.0 layers a type-based hierarchy of data types on top of XML text strings in order to govern lexical (character-level) validation of the information
    - `http://www.w3.org/XML/Schema`
  - all information at every branch of the tree is unambiguously labeled
    - hierarchical document context allows the same label to be used in different branches of the tree
  - validating tools can confirm the appropriate use of labels and the lexical structure of text beneath the labels according to the data types
- programs and applications can identify the information in an arbitrary XML document with more precision and without the burden of the validation provided for by outboard validation tools

Just using XML is not a panacea

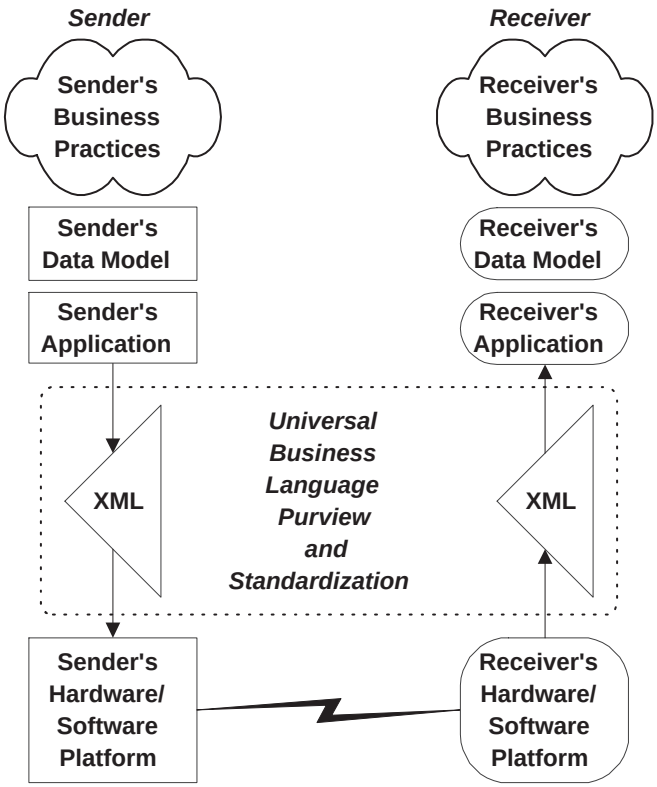
- using markup to label information in and of itself is platform independent
- labeling the information in markup using a particular vocabulary (set of labels) enables applications to access the information so labeled
- without an agreed-upon vocabulary, applications do not know under which labels particular information items can be identified
- an arbitrary, vendor-defined vocabulary is as proprietary as a non-markup-based system for labeling information

Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL) (cont.)

Objective: enable interoperability between dissimilar systems using open standards

- using XML for all of the benefits of platform, vendor and application independence
- using an agreed-upon vocabulary ensures that all users of the XML can identify the same information items using the agreed-upon labels
- the XML document of a known vocabulary can then be understood by different applications on different platforms
- free tools are available to work with the documents
    - document creation, vocabulary validation, web and print formatting, etc.



UBL does not attempt to redefine anyone's business practices

- UBL is only addressing the representation of business information in a standardized format for the purposes of interchange
- businesses may wish to modify their expectations for information transfer in order to take advantage of UBL formats

UBL does not attempt to redefine anyone's back-end data models

- the interchange of information is distinct from anyone's internal storage representation
- UBL only describes the document model of the actual interchange artefact

Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL) (cont.)

UBL is an extensible royalty-free library of standard electronic XML business documents

- e.g. documents for the transaction of business: purchase order, invoice, etc.
- customization model provides for a community creating one's own profile of existing UBL document models
- extension model provides for a community extending UBL document models with additional information items not standardized by UBL
- a common library of objects enables a community to build additional document models not standardized by UBL

XML instances express all of the information of the business document

- all calculations represented in the numbers in the XML instance
    - each community of users may have their own calculation models
    - recipient knows that all of the information found in the document already has all of the calculation models applied
- no stylesheet or processing required by the receiving application to "complete" unspecified values
    - a receiving application may choose to do completeness and consistency checking

The UBL specification was developed under an open and accountable process

- governed by OASIS Technical Committee procedures
    - `http://www.oasis-open.org/committees/process.php`
    - includes use of Roberts Rules of Order for committee process
- all committee mail list archives and both intermediate and final work products are publicly available
    - `http://lists.oasis-open.org/archives/ubl/`

Vendors can compete on product differentiation around a standardized interchange

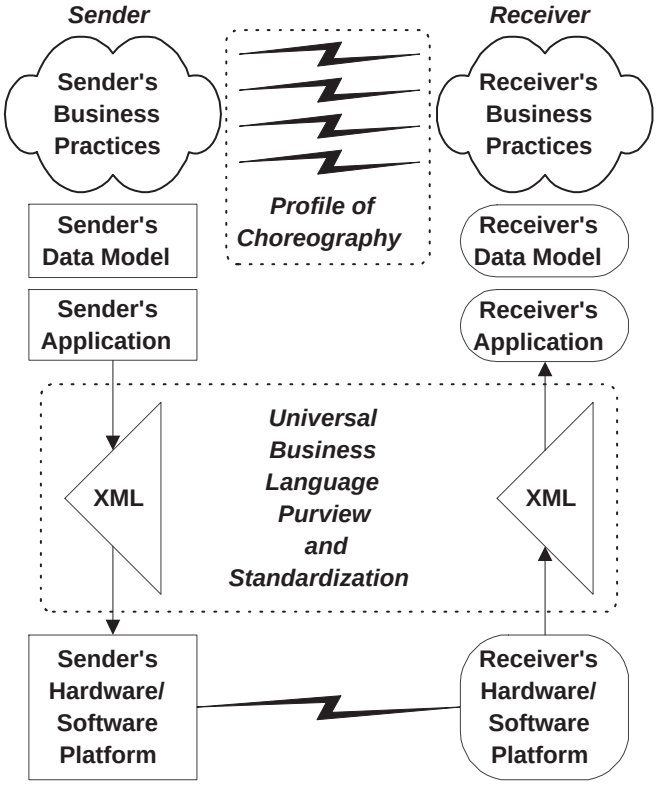
- standardization opens up a market for vendors to compete with their own innovations
    - features of implementation
    - e.g. ease of use, documentation, etc.
- it can be considered undesirable to be innovative with the document format
    - e.g. no need to reinvent an invoice that already works sufficiently well even if it is not perfect

Copyright © Crane Softwrights Ltd.

OASIS Universal Business Language (UBL) (cont.)

Communities of users have the opportunity to define a common use of UBL

- includes profiles of choreography of documents to meet arbitrary business processes
- includes modifications to the XML specifications of the UBL documents as appropriate to the needs of the community



Furthermore, individuals in the community can layer content constraints

- within any community of users, two trading partners can agree to utilize the community definition of UBL with an individual's requirements for code list values, identifier values and business rules
- note this is not depicted in the diagram but discussed later on in the material

Copyright © Crane Softwrights Ltd.

UBL history

Long history of development:

- distinguishes UBL from other standards by not starting from scratch with only a set of requirements
- 1997 - Veo Systems builds Common Business Language (CBL) 1.0
  - funding from NIST
  - first release in public domain (no licensing ownership)
- 1998 - Commerce One acquired Veo and builds CBL 2.0
  - for Commerce One electronic marketplaces
- 2000 - Commerce One partners with SAP to build xCBL 3.0
  - based on review of EDIFACT and X.12 to support EDI-type functionality
- April 2001 - UBL started with contribution of xCBL 3.0
- January 2003 - UBL 0.7 released for public review
  - 7 document types for procurement
  - utilized by some early-adopters
- September/November 2004 - UBL 1.0 released as Committee Specification and OASIS standard
  - 8 document types for procurement
- October 12, 2006 - UBL 2.0 finalized as a Committee Specification
  - 31 document types for procurement and for transport
- December 12, 2006 - UBL 2.0 standardized as an OASIS Standard
  - `http://docs.oasis-open.org/ubl/os-UBL-2.0/`
  - zero technical differences from the committee specification of the materials
    - the only difference is the wording on the cover page
  - any instances and implementations based on the committee specification need not change to support the final standard
- May 26, 2008 - UBL 2.0 Update published as errata
  - `http://docs.oasis-open.org/ubl/os-UBL-2.0-update/`
  - zero changes to document constraints
  - republished code lists to latest version of genericode
  - repaired definitions and meta data

Copyright © Crane Softwrights Ltd.

UBL FAQ

Chapter 1 - OASIS Universal Business Language (UBL)
Section 1 - OASIS Universal Business Language (UBL)

`http://www.oasis-open.org/committees/ubl/faq.php`

Highlights:

- "provide the world with standards for the electronic versions of traditional business documents"
- "recognizes established commercial and legal practices"
- "working toward the establishment of an international ecommerce infrastructure"
- "some similarities between UBL and other XML business data initiatives, but taken together, UBL's attributes make it unique"
- "UBL does not seek to compete with any existing XML business vocabularies but rather to meet a set of needs that are not being adequately met by any of them."

The committee attempts to keep the FAQ up to date with the latest information

Copyright © Crane Softwrights Ltd.

Committee structure

UBL Technical Committee

- `http://www.oasis-open.org/committees/ubl/`
- international membership (active members in Asia, Australia, Europe, North America)
    - most members have a business perspective on the use of business documents
    - few members have a technical perspective on the technology of XML and markup
    - the combination worked well to keep the focus on the business need and ensuring the role of the technical members was to support the business members
- company and individual memberships
- the intellectual property of member contributions is transferred to OASIS through the membership agreement
    - this restricts ad hoc contributions from outside the committee, thus assuring the provenance of the information that comprises the specifications

UBL Subcommittees (alphabetical)

- OASIS UBL Adoption SC
    - comprises the editorial board for `http://ubl.xml.org`
- OASIS UBL Human Interface SC
    - developing formatted output specifications
    - developing input form specifications
    - the subcommittee is not creating implementations, only detailed specifications sufficient for developers to implement
        - thought to promote the broad development of a number of available implementations
- OASIS UBL Procurement SC
    - developing, documenting and maintaining the conceptual models for the UBL document types related to the procurement process
- OASIS UBL Small Business SC
    - determining a formalized subset of UBL suitable for use by small businesses not needing the full functionality available in the entire UBL package
    - developing expressions suitable for machine-processing of the subset
- OASIS UBL Transportation SC
    - developing, documenting and maintaining the conceptual models for the UBL document types related to transportation

Copyright © Crane Softwrights Ltd.

Committee structure (cont.)

Localization subcommittees develop UBL-related projects with translation objectives

- semantic descriptions
- user interfaces
- OASIS UBL Chinese Localization Subcommittee
- OASIS UBL Danish Localization Subcommittee
- OASIS UBL German Localization Subcommittee
- OASIS UBL Italian Localization Subcommittee
- OASIS UBL Japanese Localization Subcommittee
- OASIS UBL Korean Localization Subcommittee
- OASIS UBL Spanish Localization Subcommittee
- OASIS UBL Turkish Localization Subcommittee

Community driven collaboration for supporting new localizations

- for any language of interest, publicly-edited spreadsheets are created for a community to build with localized definitions of UBL constructs
- `http://ubl.xml.org/forums/ubl-international-data-dictionary-idd-contributions`
- at any point a localization committee can then be formed to take ownership of the definitions and submit them for inclusion in the UBL International Data Dictionary (IDD)
- see Localization spreadsheets (page 127) for more details

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Modular suite of specifications of infrastructure that enables electronic commerce

- `http://www.ebxml.org`
- suitable for implementation by enterprises of any size
- suitable for use by enterprises in any geographical location
- designed for use over the Internet
    - "builds on the experience and strengths of existing EDI knowledge"

Globally-developed and open XML-based standards

- provides for plug-and-play shrink-wrapped solutions
- implementations of ebXML components are freely available

Collaborative global project

- OASIS
    - Organization for the Advancement of Structured Information Standards
- United Nations/ECE agency CEFACT
    - ECE - Economic Commission for Europe
    - CEFACT - Centre for Trade Facilitation and Electronic Business

Copyright © Crane Softwrights Ltd.

ebXML standards that are also ISO/IEC standards

ebXML CPPA V 2.0 - ISO/IEC 15000-1

- ebXML Collaboration Protocol Profile and Agreement
- definitions of the sets of information used in business collaboration
- business partner definitions of capabilities
- profile - data regarding technical abilities to engage in electronic business collaboration
- agreement - data agreed to configure the public, shared aspects of protocols
    - aligned with ebBPSS and ebMS

ebXML MS V 2.0 - ISO/IEC 15000-2

- ebXML Message Service
- develop and recommend technology for the transport, routing and packaging of business transactions using standard Internet technologies
- communication-protocol neutral method for exchanging electronic business messages
- enveloping constructs supporting reliable, secure delivery
- contain payloads of any format type
- XML framework that leverages common Internet standards
- message consumption outside of the scope of ebMS

ebXML RIM V 2.0 - ISO/IEC 15000-3

- ebXML Registry Information Model
- information model for the ebXML Registry for definition of content and meta data
- stable store of information in a federated architecture
- facilitate ebXML-based partnerships and transactions
- e.g. schemata, documents, process descriptions, context descriptions, UBL models, parties, software components, etc.
    - identified by URN identifiers
- could be viewed simply as an online distributed data base store

ebXML RS V 2.0 - ISO/IEC 15000-4

- ebXML Registry Services
- services specification for the ebXML Registry for access to content
- interaction protocols, message definitions, message schemata, etc.
- enable sharing of information between interested parties to enable business process integration
- could be viewed simply as the methods and functions of online access to the repository

Copyright © Crane Softwrights Ltd.

ebXML - Electronic business using XML (cont.)

ebXML standards that are also ISO/IEC standards (cont.)

ebXML CCTS V2.01 (under aegis of UN/CEFACT TMG) - ISO/IEC 15000-5

- ebXML Core Components Technical Specification
- syntax-neutral information about real-world business concepts
    - basing two syntactic on the same semantic concept leads to interoperability
- a methodology for defining messages
    - defines what you should call and how you should structure your components in your documents
- based on legacy EDI perspective of business concepts
- basis for implementing interoperable XML business standards
    - or EDIFACT, or X12, or whatever actual syntax
- human-readable and machine-processed representations
- work led by UN/CEFACT Techniques and Methodology Group (TMG)

Other ebXML specifications

ebXML BPSS

- technical business process specification
- standard language with which business systems can be configured to support business collaboration between trading partners
- for peer-to-peer interchange that crosses domains of control

Free implementations of some components of ebXML are already available

- `http://www.freebxml.org/` from Hong Kong

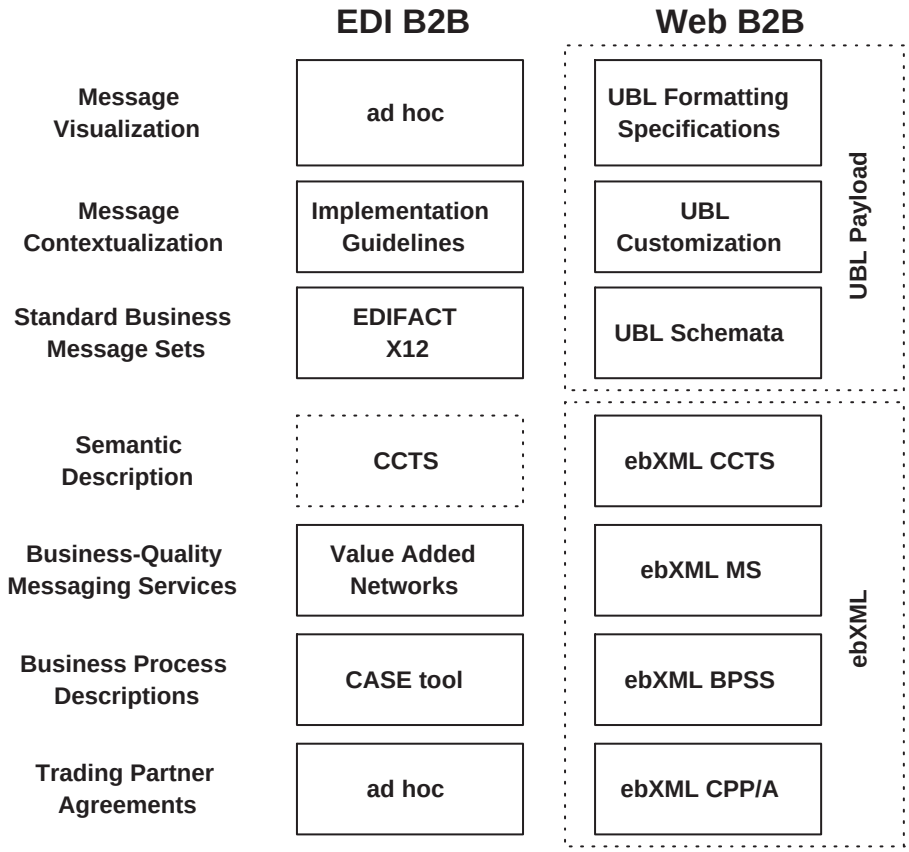
No actual message payloads being standardized in ebXML

- the "payload" is the meat of the message with the actual information being conveyed by the message infrastructure
- decision in May 2000 to not include payload syntax from initial set of deliverables
    - lead to the formation of the Universal Business Language Technical Committee
- UBL documents are suitable as payloads for ebXML messages

Copyright © Crane Softwrights Ltd.

ebXML - Electronic business using XML (cont.)

Comparing the EDI and UBL standards stacks:



Opportunity for semantic interoperability using CCTS

- UN/CEFACT Core Component Technical Specification used to define a core component library of semantic constructs
- the information items in the EDI messages will be based on the same core component library as the information items in the UBL schemata
- active harmonization effort underway in UN/CEFACT TBG-17 Harmonization group to find and catalogue all UBL semantics in the core component library

Copyright © Crane Softwrights Ltd.

UBL does not try to address every electronic business interchange problem

- business needs are as unique as the entities doing business
- a wide range of candidate uses of UBL documents is described in the specification

UBL does not try to define or constrain any business process

- business processes are ad hoc or developed using process modeling methodologies
- UBL document models can plug into a business process to satisfy interchange requirements
- an opportunity to move quickly to a production working interchange environment from an abstract business process definition
    - map information model to the UBL business objects and document types

UBL is only defining the vocabulary and structure of electronic business documents for interchange

- normative deliverables:
    - spreadsheet specification of the vocabulary
        - ISO/IEC 11179 dictionary entry names
        - UBL element names
        - cardinalities (how many of the elements may or must be present)
        - English language definitions
    - W3C Schema (XSD) expressions of the document constraints
        - defines a set of XML interchange document models and vocabulary
        - constrains the nesting of information in XML constructs
        - also includes constrains not expressible in W3C Schema
- UBL does not force any constraints on back-end processing, data models, database schemata, calculation models, etc.
- customization approaches available to address tailored interchange requirements
    - restricting and extending UBL document models

UBL employs a framework for communities of users and for trading partners to declare conformance to a particular suite of controlled vocabularies (e.g. code lists, identifiers)

- uses a supplemental methodology and approach to layer code list and value validation on top of XSD schema validation
- a community of users of UBL specifies which controlled vocabularies are in use
    - orthogonal to UBL specification

Copyright © Crane Softwrights Ltd.

The role of UBL in e-commerce (cont.)

UBL attempts to address 80%-90% of requirements for information interchange

- basic requirements based on business experience of committee members
- UBL 2 addresses 80%-90% of more documents than UBL 1
    - UBL 2 does not attempt to flesh out the remaining 10-20% of business information not implemented in UBL 1

Customization

- provides for specifying how UBL is the basis for building only that which is needed by communities of users or by two trading partners
- different profiles of business practices may impose the use or non-use of available UBL constructs
- can specify custom document models based on UBL information models
    - referred to in committee as "compatible customization"
    - focus is on the information model
- can specify custom subsets of UBL document models
    - referred to in committee as "conformant customization"
    - focus is on the XML syntax

Extensibility

- provides for addressing those parts of the missing components needed by communities of users or by two trading partners
- custom extensions are allowed to be added to standardized UBL document models
    - developed using compatible customization methods
- communities can build their own documents utilizing the UBL library of business objects

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

The role of UBL in e-commerce (cont.)

UBL scenarios and business processes are only documentary

- used to support the decisions made by committee members when choosing document structures
- there is no obligation to match the business document flows or workflows described by UBL
- a plausible scenario was required by the committee with which to frame the directions undertaken and the decisions made

Users need only use those documents of UBL that they require

- no obligation to implement all UBL document types

Users need only use those optional portions of UBL documents that they require

- no obligation to implement all UBL information items

Communities of users can define a "UBL Customization" for their collective use

- profiles of different scenarios for exchanging business documents
- specifications of document types in each profile
    - a single UBL document may have different specifications of use in two profiles
- choice of standardized information items in each document type
- specifications of extension information items in each document type
- documentation of calculation models to populate the information items

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

The role of UBL in e-commerce (cont.)

UBL provides a key economic benefit to the end user through standardization

- realized through availability of off-the-shelf inexpensive business software to handle the functions of dealing with the information transmissions
- the risk of investment by developers is mitigated by having a wider community of users adopting the technology

Possible stimulus for a major economic shift in the area of electronic commerce software

- the same shift as has happened in the past in other areas of information processing
    - initial "cottage industry" of vendors creating expensive custom systems for specific applications by users who can afford to pay
    - through standardization it becomes possible to address most (though not all) requirements
        - sufficient functionality that people will adopt the solution because the implementation is inexpensive to use and experiment with
    - sufficient customer pull creates a demand for a wide variety of software vendors to meet the need of a growing number of users of the low-cost or free technology
        - vendors can still compete on aspects of differentiation (usability, performance, features, integration, etc.)
- operating systems (1960's)
    - migration from bespoke systems to portable machine-independent systems
- programming languages (1970's)
    - migration from specific-purpose languages to general-purpose languages
- publishing systems (1980's)
    - migration from typesetting and typography to desktop publishing and generalized markup
- hypertext systems (1990's)
    - migration from custom applications (e.g. Ted Nelson's Xanadu, Bill Atkinson's Hypercard, etc.) to all-purpose web browsers (Tim Berners-Lee's HTML)

Standardization creates a marketplace for inexpensive products and support services

- potential big savings for small companies
- software, training, books can all be developed for a larger customer base than with a proprietary technology

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Where is UBL going?

UBL 2.x development within OASIS

- base delivery 2.0 plus extension releases for "dot versions" (e.g. 2.1, 2.2, etc.) if the committee decides they are important enough to issue
- new document types
  - support a wider range of messages and scenarios
- new common library components
  - support a wider collection of information objects to use in messages
- new code lists
  - support a wider set of code lists and newer versions of code lists
- UBL 2.1 planned for 2009/2010 time frame
  - the technical committee is currently accepting use cases and contributions
  - UBL 2.1 will include new business objects and new document types

UBL next generation development will be within UN/CEFACT

- agreed in March 2006 - UN/CEFACT and UBL collaboration
  - "UN/CEFACT recognizes UBL 2 as appropriate first-generation XML documents for eBusiness"
  - future UN/CEFACT deliverables constitute the upgrade path for UBL
  - in the expectation that UN/CEFACT will produce its own integrated set of XML schemas within a period of three years, OASIS will produce no further major versions of UBL past UBL 2
  - OASIS will grant UN/CEFACT a perpetual, irrevocable license to create derivative works based on UBL, provided significant work progresses within the three-years

Adopting UBL is not contravening UN/CEFACT strategic direction

- UBL is recognized by the memorandum of understanding on eBusiness standards
  - memorandum between International Electrotechnical Commission (IEC), International Organization for Standardization (ISO), International Telecommunication Union (ITU) and United Nations Economic Commission for Europe (UN/ECE, UN/CEFACT)
- "UBL is the useable stepping stone to a unified UN/CEFACT standard"
  - from Draft Business plan for a CEN/ISSS Workshop on "Interoperability in the Implementation of electronic public procurement in Europe" (CEN/ISSS WS/ePPE - renamed "Business Interoperability Initiative")

UBL business objects are being aligned with UN/CEFACT semantics

- working with TBG-17 harmonization group to catalogue every UBL 2.1-defined semantic in the UN/CEFACT core component library

Copyright © Crane Softwrights Ltd.

Formal analysis by KPMG on behalf of the Danish Ministry of Finance October 2003

- `http://www.idealliance.org/proceedings/xtech05/papers/03-05-02/`
- estimated 18 million invoices sent to public authorities each year
- conservative estimate of eliminating 10 minutes of handling for each invoice would save €94m
- conservative estimate of eliminating 17 minutes of handling and matching for each combination of order and invoice would save €160m

Legislated use of derivative of UBL 0.7 Invoice - OIOXML

- `http://en.wikipedia.org/wiki/OIOXML`
- 1.2 million invoices per month since February 2005
- data entry houses created for those without UBL support
    - free to companies with a turnover less than €2m
    - cost of €1 each invoice to other companies and public institutions
        - implemented by paying for a stamp
        - stamp affixed to the invoice as evidence of payment
- hundreds of millions of dollars/euros in savings realized just for the implementation of a single document type

Online Schematron validation

- an online service is made available to check the validity of candidate UBL submission before sending the document to the government
    - useful as a diagnostic tool for developers
    - useful as a confirmation tool for invoice submitters
- proposed initial implementation of the Danish UBL 2.0 Customization will only have Schematron validation and not any customized schema validation

Anticipated legislated use of UBL 2 may include up to 15 document types - OIOUBL

- `http://www.oioubl.info`
- "Offentlig Information Online" ("Public Information Online")
- catalogues
- ordering
- statements

OIOSI project defining and implementing interchange protocols

- `http://www.softwareborsen.dk/projekter/softwarecenter/serviceorienteret-infrastruktur`
- OIO Service-oriented Infrastructure
- open-source availability of service-oriented architecture components
- defined use of an OIOSI toolkit and UUID registry

Copyright © Crane Softwrights Ltd.

Government procurement

European e-Government: Ministers make a unanimous declaration on 2010 targets

- agreement reached in 2005
- http://www.publictechnology.net/modules.php?op=modload&name=News&file=article&sid=4060
- By 2010 all public administrations across Europe will have the capability of carrying out 100% of their procurement electronically, where legally permissible, thus creating a fairer and more transparent market for all companies independent of a company's size or location within the single market.
- By 2010 at least 50% of public procurement above the EU public procurement threshold will be carried out electronically.

NES - North European Subset

- http://www.nesubl.eu
- targeted for public procurement for northern European governments
- led by the Danes and built on the Danish experience

BII - Business Interoperability Interfaces

- http://www.en.ds.dk/bii
- targeted for public procurement for all European governments
- led by the Danes and combines requirements from CODICE Spain (for pre-award documents) and NES (for post-award documents)
- objective only to specify requirements

PEPPOL pilot - Pan-European Public eProcurement On-Line

- http://www.peppol.eu
- led by the Danes to deploy the BII requirements as a pilot project
- development of document models and open-source document transportation facilities
- available to European members to adopt and deploy in production
- The vision of the PEPPOL project is that any company and in particular SMEs in the EU can communicate electronically with any European governmental institution for the entire procurement process. The project will set up integrated pilot solutions across borders

Copyright © Crane Softwrights Ltd.

Other projects seen on the UBL radar

So many UBL projects are underway that the committee is unaware of or only just heard of without knowing details of unpublished projects

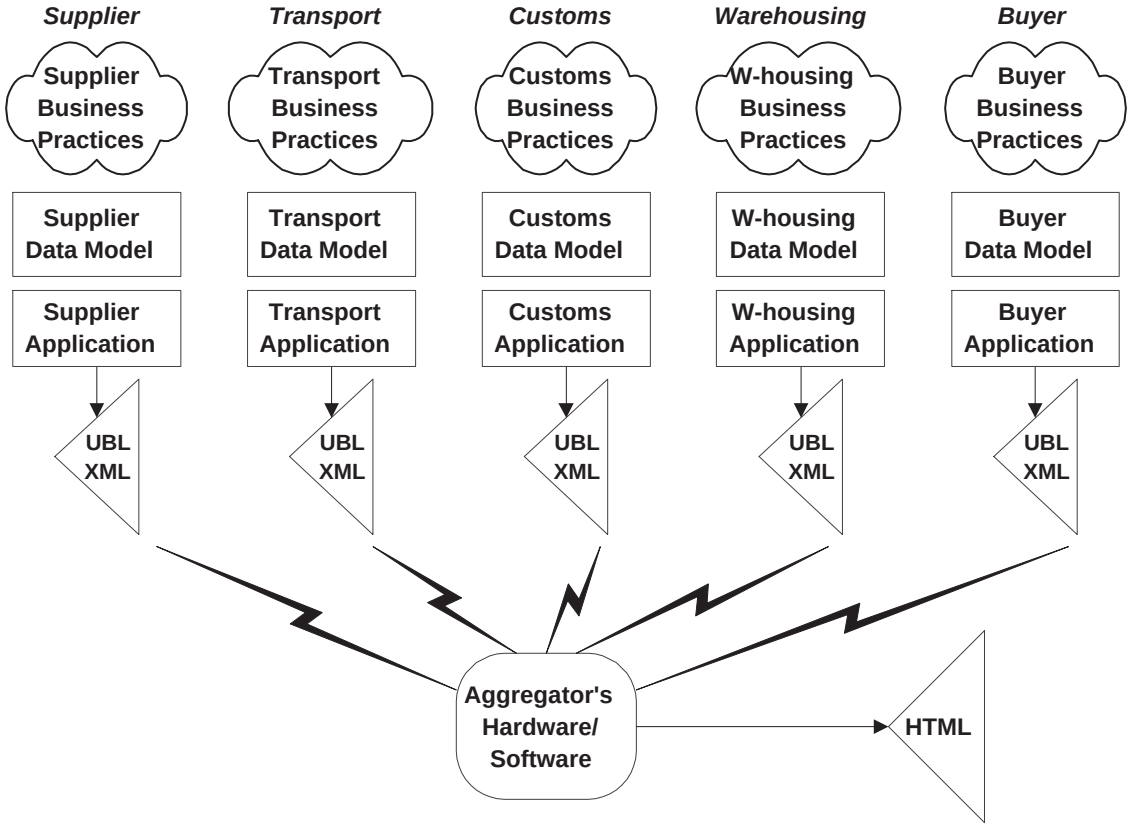
- please let us know at `info@CraneSoftwrights.com` of any others you are aware of in order for us to update these materials.
- Hong Kong
  - DTTN
  - `http://www.unece.org/cefact/single_window/sw_cases/hongkong.htm`
- Italy
  - UBL 1.0 and SBS for SME-driven business scenarios
  - `http://services.txt.it/abilities/project.html`
  - project member work in Lithuania, Slovakia, Turkey, Romania and Hungary
- Korea
  - Korean Customs Service
- Panama
  - La Cámara de Comercio de Panama
- Singapore
  - New TradeNet
- South America
  - various projects underway based on Spanish projects
- Spain
  - UBL Invoice government wide for Balearic Islands
  - Ministry of the Economy project
  - Spanish Tax Agency and the banking system
- Sweden
  - October 2005 - "Swed-Invoice" recommended for all government use by the Swedish Financial Management Authority
    - anticipated savings of SEK 4b (US$500m) in the first five years
- United Kingdom
  - February 2006 - Zanzibar marketplace, created by the UK Office of Government Buying Solutions

Copyright © Crane Softwrights Ltd.

Other projects seen on the UBL radar (cont.)

Projects (cont.)

- United States (Department of Transport)
  - Electronic Freight Management (EFM) project undergoing tests with UBL Despatch Advice, Receipt Advice, Bill of Lading and Transportation Status
    - example pilot projects with Limited Brands (China) shipping Victoria Secret clothing to Columbus, Ohio involving 14 participating companies
      - each participant has their own business practices, data models for those business practices and applications for those data models
    - not a central database, but a distributed Service-Oriented Architecture (SOA) where responses to messages are standard UBL instances
    - requesting a status report triggers a federated query to all participants in the shipping process, each returning a UBL Transportation Status message
    - the query aggregates the result into a single HTML page
    - a non-responding participant does not forestall the report
    - `http://www.tfhrc.gov/pubrds/06may/06.htm`
    - `http://www.ops.fhwa.dot.gov/freight/intermodal/efmi/`
    - `http://www.metrans.org/nuf/2007/documents/Onder.pdf`



Copyright © Crane Softwrights Ltd.

Document standardization business areas for UBL

Groups of UBL documents are being defined for two major areas of electronic business

Sourcing-to-payment procurement cycle (post-award) - in UBL 2.0

- cataloguing, ordering, invoicing, payment

Transportation - in UBL 2.0

- fulfillment, shipping

Tendering (pre-award) - coming in UBL 2.1

- notices, tenders, invitations, notifications

Sales order and reporting - coming in UBL 2.1

- requirements from ENEA in Italy

Collaborative planning, forecasting and replenishment documents - coming in UBL 2.1

- requirements from iSURF in Turkey

Some overlapping roles between scenarios

- the actual roles of parties in UBL transactions are dependent on the context of use
- e.g. the despatch party and delivery party as applied to the procurement process may differ in the transportation process
    - i.e. whether the consignor in the transportation process is actually equal to the despatch party or seller party in the procurement exchange depends on different business cases

Copyright © Crane Softwrights Ltd.

UBL 2.0 specification contents

`http://docs.oasis-open.org/ubl/os-UBL-2.0-update/`

- public home of UBL specification and files
- replaces old installation of UBL 2.0 initial release:
    - `http://docs.oasis-open.org/ubl/os-UBL-2.0/`

The UBL 2.0 updated specification two parts unpack to 190Mb of content:

- `http://docs.oasis-open.org/ubl/os-UBL-2.0.zip`
    - OASIS standard - December 12, 2006
- `http://docs.oasis-open.org/ubl/os-UBL-2.0-update-delta.zip`
    - Errata "delta" package - May 26, 2008
    - these files are copied on top of the UBL 2.0 files, replacing the ones that now have new content
    - be careful if you are overlaying these files on a modified UBL installation as some modifications may be lost
- `art/` - referenced artwork in the documentation files (page 34)
- `asn/` - ASN.1 expressions of the document models (page 99)
- `cl/` - code list expressions in genericode
- `css/` - CSS stylesheets for HTML rendering of the documentation files (page 34)
- `db/` - DocBook support files (page 34)
- `doc/` - NDR documentation file (page 92)
- `etc/` - re-use cross reference file (page 74)
- `mod/` - document model spreadsheets (page 76)
- `uml/` - UML expressions of the document models (page 81)
- `val/` - out-of-the-box two-step validation demonstration (page 119)
- `xml/` - sample instances (page 105)
- `xsd/` - document model schemata with documentation (page 113)
    - the files in this directory are the only normative files for UBL 2.0
- `xsdrt/` - document model schemata without documentation (page 113)

Copyright © Crane Softwrights Ltd.

UBL 2.0 specification contents (cont.)

The normative documentation is in XML with an HTML projection for reading in a browser

- `UBL-2.0.xml` - UBL specification hypertext document in XML
  - the hypertext document links to the components of the specification
  - `art\` includes all artwork
  - `UBL-2.0.html` - documentation rendering in HTML format (page 38)
    - this is where people can browse the content of the hypertext document
    - `db\` used for transformation from DocBook XML source to HTML target
    - `css\` used for rendering HTML
  - `UBL-index-2.0.pdf` - documentation rendering in PDF format
    - this is a committee process artefact that serves no useful purpose
    - included only to satisfy archaic process requirements that are under OASIS review in consideration of removing its mandated presence
- note that the directories listed above without page number citations are related to the documentation files in support of the `UBL-2.0.xml` hypertext XML document and its HTML and PDF renderings
- `UBL-2.0-update.html` - errata-only documentation
  - the main UBL documentation in the hypertext document did not change with the update
  - this errata document only indicates which files have changed

`http://www.oasis-open.org/committees/ubl`

- public home of UBL committee working pages
- committee membership
- committee documents and files
- committee email archive
- public comments submission and archive
  - "Send A Comment" button
  - do not use this list for general comments or questions, only for comments to be formally reviewed by the technical committee

`http://docs.oasis-open.org/ubl/`

- public home of UBL committee repository files
- milestone deliverables
- committee documents and files too large for the committee working pages

Copyright © Crane Softwrights Ltd.

UBL.xml.org and UBL-Dev

`http://ubl.xml.org`

- community support site replacing the former Support Subcommittee site
  - `http://www.oasis-open.org/committees/download.php/28723/support.htm`
- resources created by the UBL committee and by the UBL community
  - committee documents
  - contributions from academic research projects
  - vendors and users
  - model trading partner agreement templates
- publicly-available customizations
- news
- events
- products
- services
- forums
- blogs

`http://www.oasis-open.org/archives/ubl-dev/`

- un-moderated publicly-subscribed developer list
- `http://www.oasis-open.org/mlmanage/`

Copyright © Crane Softwrights Ltd.

Chapter 2 - Parties, document types and profiles

- Introduction - Participants and document flows
- Section 1 - UBL parties
- Section 2 - UBL document types
- Section 3 - Customizations and profiles of UBL

Outcomes

- understand which business parties might be interested in UBL documents
- overview the set of UBL documents
- distinguish between customizations and profiles for UBL

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Participants and document flows

Typically two or more trading partners agree to engage in a business transaction

- the initial UBL 2 scope is in two areas
  - procurement of goods or services
  - transport of goods

A given business transaction may involve a number of participants (parties)

- individual, a group, or a body having a role in a business function
- a single party may play a number of different roles in a given business transaction
- all of the roles in the UBL scenarios are representative and need not actually be realized as real people or parties

Contexts defined for 21 roles involved in 31 document types being exchanged

- most document exchanges involve two parties
- some document exchanges involve multiple parties
- some roles are not involved in sending and receiving documents

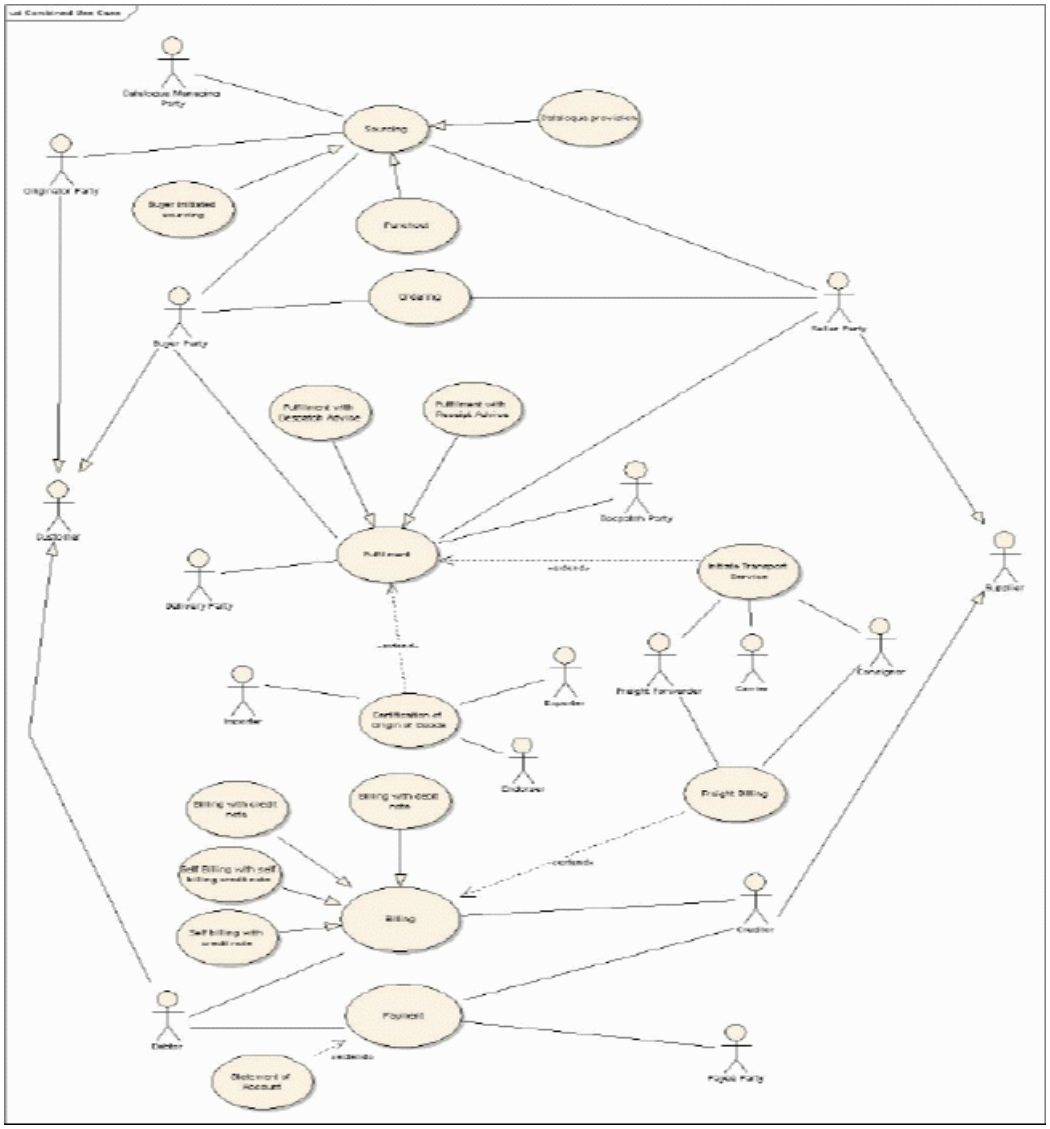
Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Participants and document flows (cont.)

Documentary scenario use case diagram outlines the roles of the many participants:

- the following image is a compressed version of the image found in the specification
  - under "UBL 2.0 Context of Use" (4.); view it directly to see the detail



Copyright © Crane Softwrights Ltd.

Participants and document flows (cont.)

A given business transaction may involve a number of document exchanges

- exchanges are choreographed in a UBL scenario as only an example exchange of documents for documentary purposes to illustrate their use
- UBL does not require trading partners to engage in any particular exchange of documents
- communities and users of UBL can create their own scenarios using documents of the various document types
- communicating and negotiating the requirements for document exchanges are not part of UBL
    - e.g. could be implemented by ebCPP/A partner agreements

Communities of users are creating "profiles" of choreography of documents for different scenarios

- a scenario describes the objective of the interchange of documents
- the profile fulfills the scenario by describing a particular choreography of a subset of documents
- the same document can be used differently in different profiles
- the community could also customize a UBL document in different ways for each of the profiles they define
    - different constructs have applicability in different scenarios
- a hierarchy of profile identity:
    - `UBLVersionID` - version of UBL
        - e.g. "2.0", "2.1", etc.
    - `CustomizationID` - customization of a particular UBL version
        - e.g. community
    - `ProfileID` - profile within a particular customization
        - e.g. business process
    - each specific document type is customized for the profile
        - there may be different document type customizations for different profiles within the same customization

Benefits can be realized by implementing only a single document

- e.g. Denmark realized millions of euros in savings implementing (and legislating) only Invoice and no other document types
    - the Danish government already moving to support (and legislate) other document types

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

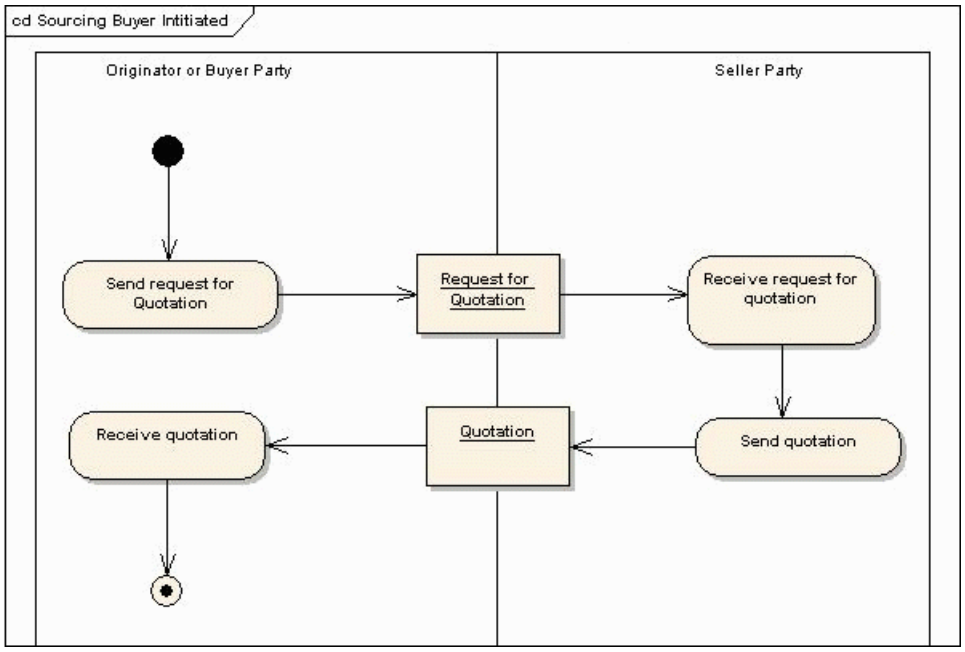
Participants and document flows (cont.)

Each document exchange is documented in the specification with sample workflows using activity diagrams:

- the vertical "columns" of activity diagrams are colloquially referred to as "swim lanes"
- document types are depicted in square-cornered boxes with underscored labels
- actions are depicted in round-cornered boxes
- the start of the process is the solid-filled circle
- the end of the process is the hollow-filled circle

Example workflow activity diagram for quotations excerpted from UBL documentation:

- excerpted from the specification section 4.3.2.



Find all diagrams in the UBL main documentation file under section 4.

Copyright © Crane Softwrights Ltd.

Parties in sourcing-to-payment procurement cycle

The UBL committee characterizes a number of different parties each with possible roles:

- documented in the specification in section 4.2.1.
- customer party
  - originator role
    - initiating the need for the procurement transaction
    - contact point for queries
    - e.g. the person needing a box of pencils
  - buyer role
    - purchases the goods or services on behalf of the originator
    - e.g. the person buying a box of pencils
  - delivery role
    - to whom the goods or services should be delivered
    - e.g. the person receiving the delivered box of pencils
  - debtor / accounting customer role
    - responsible for payment and billing issue resolution
    - e.g. the person paying for the box of pencils
- supplier party
  - seller role
    - responsible for handling originator and buyer services
    - legally responsible for providing the goods or services
    - e.g. the person selling boxes of pencils
  - despatch role
    - from whom goods or services are to be collected
    - e.g. the person delivering the box of pencils
  - creditor / accounting supplier role
    - responsible for claims payment, billing issue resolution and settlement arrangement
    - e.g. the person receiving the payment for the box of pencils
  - payee role
    - to whom the invoice is paid
    - e.g. the person receiving the money for the box of pencils

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Parties in sourcing-to-payment procurement cycle (cont.)

Parties (cont.)

- other party
  - catalogue managing role
    - party receiving a catalogue (not the originator or buyer if nothing ordered)
    - e.g. a person considering buying a box of pencils
  - information content owner role
    - responsible for integrity of information provided about an item
    - e.g. the person responsible for a catalogue containing the box of pencils
  - receiver role
    - party receiving a document
    - e.g. the software receiving an application response
  - sender role
    - party sending a document
    - e.g. the software sending an application response
  - consignor role
    - (procurement) party where goods are to be collected from
    - (transport) receiver of invoice and payer of transportation service
  - consignee role
    - receiver of goods as stipulated in the transport contract
  - freight forwarder role
    - (procurement) arranger of the carriage of goods on behalf of consignor or consignee
    - (transport) creator of invoice who bills consignor for a transportation service
  - carrier role
    - provider of physical transport services
  - exporter role
    - supplier of goods through an international purchase
  - endorser role
    - government appointee with right to certify a certificate of origin
  - importer role
    - receiver of goods through international purchase

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types in UBL

A "document type" represents a class of documents with the same logical structure

- actual documents are instances of the document type
- UBL identifies each document type by a contraction
  - e.g. `CatalogueItemSpecificationUpdate`

Each document exchange involves a transmission and reception of an XML instance

- the instance validates against the formalism declaring the structure of the class of documents defined by the document type
- the sender of the instance generates the XML instance according to the document type using the information that needs to be transmitted
- the recipient of the instance extracts the received information from within the XML instance that has been validated to be properly structured
  - this may require other validation such as code list value validation and negotiated business rule validation

UBL does not define how the transmission and reception of an XML instance happens

- could be by any electronic means whatsoever
- e.g. could be implemented by formal ebMS message services
- e.g. could be implemented by informal services: email, FTP, HTTP REST (Representational State Transfer), etc.
- responsibility of lower layers (see page 12)

Separate document types for different business actions

- e.g. for ordering there are separate document types order, order change, order cancel, etc.
- e.g. for catalogue there are separate document types request, create, update, etc.
- this is unlike "function codes" in an EDI message where the code indicates the action of the message

Summary of UBL 2.0 Document Types (section 4.11.)

- the specification summarizes a description of each document, the processes involved and the roles involved

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Catalogue provision

- a catalogue is a document that describes items and prices
- a seller sends information regarding items available for purchase to a potential customer
- role that receives a catalogue is a catalogue managing party because of the potential (not guarantee) of a subsequent purchasing request
- request a catalogue (4.3.1.2., 4.3.1.3., 4.3.1.4.)
    - `CatalogueRequest`
    - "A document to request a Catalogue from a seller. May be either an entire new Catalogue or an update (at the discretion of the Seller)."
    - processes: create catalogue, update item specification, update pricing
    - submitter role: contracting party
    - receiver role: seller
- create a catalogue (4.3.1.2.)
    - `Catalogue`
    - "A document produced by a party in the procurement chain that describes items and prices. The document typically enables the transmission of information regarding pricing and catalogue details for goods and services offered by a seller to a buyer."
    - process: create catalogue
    - submitter role: seller
    - receiver role: contracting party
- update a catalogue item specification (4.3.1.3.)
    - `CatalogueItemSpecificationUpdate`
    - "A document to update information about Items in an existing Catalogue."
    - process: update catalogue item specification
    - submitter role: seller
    - receiver role: contracting party
- update catalogue pricing (4.3.1.4.)
    - `CataloguePricingUpdate`
    - "A document to update information about Prices in an existing Catalogue."
    - process: update catalogue pricing
    - submitter role: seller
    - receiver role: contracting party
- delete catalogue (4.3.1.5.)
    - `CatalogueDeletion`
    - "A document to cancel an entire Catalogue. All previous Catalogue information becomes obsolete."
    - process: delete catalogue
    - submitter role: seller
    - receiver role: contracting party

Copyright © Crane Softwrights Ltd.

Document types for sourcing (cont.)

Customer-initiated sourcing and punchout

- the originator may explicitly request a quotation (customer-initiated) or may directly accesses a seller's application by some mechanism outside the scope of UBL (punchout), both returning a quotation
- request a quotation (4.3.2.)
  - `RequestForQuotation`
  - "A document to request pricing and availability information about goods or services. The document may requesting a quote on specified goods or services."
  - process: customer initiated sourcing
  - submitter role: originator
  - receiver role: seller
- deliver a quotation (4.3.2., 4.3.3.)
  - `Quotation`
  - "A document to specify pricing and availability information about goods or services. The document which, with a view to concluding a contract, sets out the conditions under which the goods are offered."
  - processes: customer initiated sourcing, sourcing punchout
  - submitter role: seller
  - receiver role: originator

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types for ordering

Ordering

- buyer places an order to the seller (4.4.)
  - `Order`
  - " A document that contains information directly relating to the economic event of ordering products. The document by means of which a customer initiates a transaction with a supplier for the supply of goods or services as specified, according to conditions set out in an offer, or otherwise known to the customer."
  - process: ordering
  - submitter role: buyer
  - receiver role: seller
- seller confirms receipt with either commitment to fulfill or with rejection (4.4.)
  - `OrderResponseSimple`
  - "A document responding to the customer to indicate simple acceptance or rejection of an entire order. The document acknowledging an undertaking to fulfill an order and confirming conditions or acceptance of conditions."
  - process: ordering
  - submitter role: seller
  - receiver role: buyer
- seller proposes changes to an order (4.4.)
  - `OrderResponse`
  - "A document responding to the customer to indicate detailed responses against a single order already received."
  - process: ordering
  - submitter role: seller
  - receiver role: buyer
- buyer initiates a change to an established order (4.4., 4.5.2.)
  - `OrderChange`
  - "A document that contains information directly relating to the economic event of changing an order already sent."
  - processes: ordering, fulfillment with receipt advice
  - submitter role: buyer
  - receiver role: seller
- buyer cancels an established order (4.4., 4.5.2.)
  - `OrderCancellation`
  - "A document that advises either party of the cancellation of an Order."
  - processes: ordering, , fulfillment with receipt advice
  - submitter role: buyer
  - receiver role: seller

Copyright © Crane Softwrights Ltd.

Document types for fulfillment

Fulfillment

- despatch party confirms to delivery party the shipment of items (4.5.)
  - `DespatchAdvice`
  - "A document that describes the content of goods shipped. Document/message by means of which the seller or consignor informs the consignee about the despatch of goods."
  - process: fulfillment
  - submitter role: despatch
  - receiver role: delivery
- delivery party confirms to despatch party the receipt of items (4.5., 4.5.2.)
  - `ReceiptAdvice`
  - "A document that advises the goods received and accepted by the buyer. The document acknowledges the receipt of goods and in addition may indicate receiving conditions."
  - processes: fulfillment, fulfillment with receipt advice
  - submitter role: delivery
  - receiver role: despatch

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types for billing

Traditional billing

- raise an invoice
- supplier (creditor) invoices customer (debtor) when goods or services are delivered or provided, (4.6.2.1., 4.6.2.2.)
    - `Invoice`
    - "A document claiming payment for goods or services supplied under conditions agreed between the supplier and the customer. In most cases this document describes the actual financial commitment of goods or services ordered from the supplier."
    - processes: billing using credit notes, billing using debit notes
    - submitter role: supplier accounting party
    - receiver role: customer accounting party

Credit note

- issue a credit note
- seller (creditor) indicates to customer (debtor) that an invoice is incorrect (4.6.2.1., 4.6.3.1.)
    - `CreditNote`
    - "A document for a supplier to specify a reduced payment. The document for providing credit information to the relevant party."
    - processes: billing using credit notes, self billing using debit notes
    - submitter role: supplier accounting party
    - receiver role: customer accounting party

Debit note

- issue a debit note
- customer (debtor) indicates to seller (creditor) that an invoice is incorrect(4.6.2.2.)
    - `DebitNote`
    - "A document for a customer to specify a reduced payment. The document for providing debit information to the relevant party."
    - process: billing using debit notes
    - submitter role: customer accounting party
    - receiver role: supplier accounting party

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types for billing (cont.)

Traditional billing (cont.)

Self-billing (a.k.a. "billing on receipt")

- no change in roles, only in the initiation of the invoice document itself
- customer (debtor) raises an invoice in the name and on behalf of the supplier (creditor) (4.6.3.1., 4.6.3.2.)
  - `SelfBilledInvoice`
  - "A document provided by a customer, in the name and on behalf of the supplier, describing the claim for payment for goods or services supplied under conditions agreed between the supplier and the customer."
  - processes: self billing using credit notes, self billing using self billed credit notes
  - submitter role: customer accounting party
  - receiver role: supplier accounting party
- customer (debtor) raises an credit note in the name and on behalf of the supplier (creditor) (4.6.3.2.)
  - `SelfBilledCreditNote`
  - "A document for a customer to specify a reduced payment in a Self Billing environment. The document indicates that the customer is claiming credit in a self billing environment."
  - process: self billing using self billed credit notes
  - submitter role: customer accounting party
  - receiver role: supplier accounting
- the process may instead involve a regular credit note rather than a self-billed credit note

Freight billing

- freight forwarder initiates the billing of the consignor for logistic services (4.6.4.)
  - `FreightInvoice`
  - "A document issued by a transport operation specifying freight costs and charges incurred for a transport operation and stating conditions of payment."
  - process: freight billing
  - submitter role: freight forwarder
  - receiver role: consignor or consignee

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Account Payment (remittance)

- notify a remittance (funds transfer)
- payee (usually creditor) is notified of funds transferred against the account of the debtor (4.7.)
    - `RemittanceAdvice`
    - "A document to specify that funds have been transferred from the customer to the supplier. The document advising of the remittance of payment."
    - process: payment
    - submitter role: customer accounting party and/or payee
    - receiver role: supplier accounting party and/or payee

Account Statement

- notify the account status
- debtor is notified of the status of billing (4.7.1.)
    - `Statement`
    - "A document to list the financial transactions between customer and supplier and notify of their status. This is a Statement of Account and not intended as a summary Invoice."
    - process: report state of accounts
    - submitter role: supplier accounting party
    - receiver role: customer accounting party

Account Reminder

- notify the continued need for a payment
- debtor is reminded of the need for a payment (4.6.5.)
    - `Reminder`
    - "A document used to request payment."
    - process: reminder for payment
    - submitter role: supplier accounting party and/or payee
    - receiver role: customer accounting party and/or payee

Copyright © Crane Softwrights Ltd.

Document types for transport services

Transport

- ordering of logistical services for international trade

Forwarding

- instructions for the transportation services
- consignor (shipper) requests services of forwarder, carrier, shipping agent, etc. (4.8.)
  - `ForwardingInstruction`
  - "The document used by any party who gives instructions for the transportation services required for a consignment of goods to any party who is contracted to provide the transportation services. The parties who issue this document are commonly referred to as the shipper or consignor while the parties who receive this document are forwarders, carriers, shipping agents, etc. Note that this document may also be issued by a forwarder or shipping agent in their capacity as a Transport Service Buyer. This document may be used to arrange for the transportation (1) of different types of goods or cargoes; (2) whether containerized or non-containerized; (3) through different modes of transport including multi-modal, and (4) from any origin to any destination. The document issued to a freight forwarder, giving instructions regarding the action to be taken by the forwarder for the forwarding of goods described therein."
  - process: initiate transport services
  - submitter role: consignor (or consignee), freight forwarder
  - receiver role: freight forwarder, carrier

Status

- status information for the transportation services
- forwarder, carrier, shipping agent, etc. indicates status of transportation (4.8., 4.10.)
  - `TransportationStatus`
  - "A message to report the transport status and/or change in the transportation status (i.e. event) between agreed parties."
  - processes: initiate transport services, report status of goods
  - submitter role: freight forwarder
  - receiver role: consignee, consignor

Copyright © Crane Softwrights Ltd.

Document types for transport services (cont.)

Transport (cont.)

Bill of lading

- details of transportation, charges, terms and conditions of service
- physical transportation provider (carrier) instructs services requestor (shipper, consignor, etc) (4.8.)
  - `BillOfLading`
  - "A document issued by the party who acts as an agent for the carrier or other agents, to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided. The party issuing this document does not necessarily provide the physical transportation service. It corresponds to the information on the Forwarding Instructions. It is used for any mode of transport. A Bill of Lading may serve as a contractual document between the parties for the transportation service. The document evidences a contract of carriage by sea and the acceptance of responsibility for the goods by the carrier, and by which the carrier undertakes to deliver the goods against surrender of the document. A provision in the document that the goods are to be delivered to the order of a named person, or to order, or to bearer, constitutes such an undertaking. A negotiable document that evidences a contract of carriage by sea and the taking over or loading of goods by carrier, and by which carrier undertakes to deliver goods against surrender of the document."
  - process: initiate transport services
  - submitter role: freight forwarder, carrier
  - receiver role: consignor (or consignee), freight forwarder
- almost identical to waybill but the bill of lading has legal status
  - a bill of lading carries title and travels with the goods
  - whoever holds the bills owns the goods

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types for transport services (cont.)

Transport (cont.)

Waybill

- non-negotiable and un-assignable details of transportation, charges, terms and conditions used as a cargo receipt
  - not required to be surrendered at destination in order to pick up the cargo
- physical transportation provider (carrier) informs services requestor (shipper, consignor, etc) (4.8.)
  - `Waybill`
  - "A document issued by the party who acts as an agent for the carrier or other agents to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided. The party issuing this document may not provide the physical transportation service. It corresponds to the information on the Forwarding Instructions. It is used for all modes of transport. It may serve as a contractual document between the parties for the transportation service. A Waybill is a non-negotiable document evidencing the contract for the transport of cargo. It provides information similar to Bill of Lading but is not negotiable and cannot be assigned to a third party."
  - process: initiate transport services
  - submitter role: freight forwarder, carrier
  - receiver role: consignor (or consignee), freight forwarder
- almost identical to bill of lading but the waybill is just information and has no legal status
  - waybill documents arrive well ahead of the goods and carry no title

Packing list

- state the distribution of goods in individual packages (4.8.)
  - `PackingList`
  - "A document stating the detail of how goods are packed. The document specifies the distribution of goods in individual packages (in trade environment the despatch advice message is used for the packing list)."
  - process: initiate transport service
  - submitter role: consignor
  - receiver role: freight forwarder

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types for origin certification

Certification of origin of goods

- a Certificate of Origin (CoO) is required by governments declaring that goods in a particular international shipment are of a certain origin
- exporter signs CoO and applies it to authority
- endorser (issuer of CoO) verifies claims of origin
- importer received CoO
- certify the origin of goods (4.9.)
    - `CertificateOfOrigin`
    - "A document required by governments, declaring that goods in a particular international shipment are of a certain origin. Customs offices will use this document to determine whether or not a preferential duty rate applies on the products being imported and whether a shipment may be legally imported during a specific quota period. The document identifies which authority or body authorized to issue it certifies expressly that the goods to which the certificate relates originate in a specific country. The word 'country' may include a group of countries, a region, or a part of a country. This certificate may also include a declaration by the manufacturer, producer, supplier, exporter, or other competent person."
    - processes: certification of origin of goods
    - submitter role: exporter, issuer
    - receiver role: issuer, importer

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Inquiry detail

- wrap an arbitrary attachment in a UBL document while referencing the UBL document to which the attachment applies
  - `AttachedDocument`
  - "In effect a 'wrapper' UBL envelope that may contain anything. This allows a referenced document to be included in the package of documents being exchanged."
  - processes: all
  - submitter role: sender
  - receiver role: receiver

Response detail

- indicate an application's response to a transaction
  - `ApplicationResponse`
  - "A document to indicate the application's response to a transaction at the business application level concerning the processing of a document."
  - processes: all
  - submitter role: sender
  - receiver role: receiver

Copyright © Crane Softwrights Ltd.

Customizations and profiles of UBL

A customization of UBL for a community is a collection of profiles

- a hierarchy of profile identity:
  - `UBLVersionID` - version of UBL
    - e.g. "2.0", "2.1", etc.
  - `CustomizationID` - customization of a particular UBL version
    - e.g. community
  - `ProfileID` - profile within a particular customization
    - e.g. business process
  - each specific document type is customized for the profile
    - there may be different document type customizations for different profiles within the same customization

Profile typically includes a prescribed message exchange choreography

- which documents are sent in which order in response to other documents

Should include prescribed code list conformance criteria

- an itemized list of all codes and identifiers agreed upon by the community
- may be restricted by trading partners to stay within the community definition
- may be extended by trading partners to stay

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Customizations and profiles of UBL (cont.)

As small as a single document type

- a single company can create a single profile for use by any of their customers to submit to the company

As big as a government initiative

- e.g. the OIOUBL legislation in Denmark for public procurement
- recall The Danish UBL experience (page 28)
- example of a migration from simple to complex requirements
    - OIOXML implements only a single document type (invoice)
    - OIOUBL implements 15-20 document types

As major as a concerted pan-European effort for government procurement

- e.g. the PEPPOL initiative to prototype BII requirements to meet European 2010 requirements for government electronic commerce
- recall Government procurement (page 29)

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

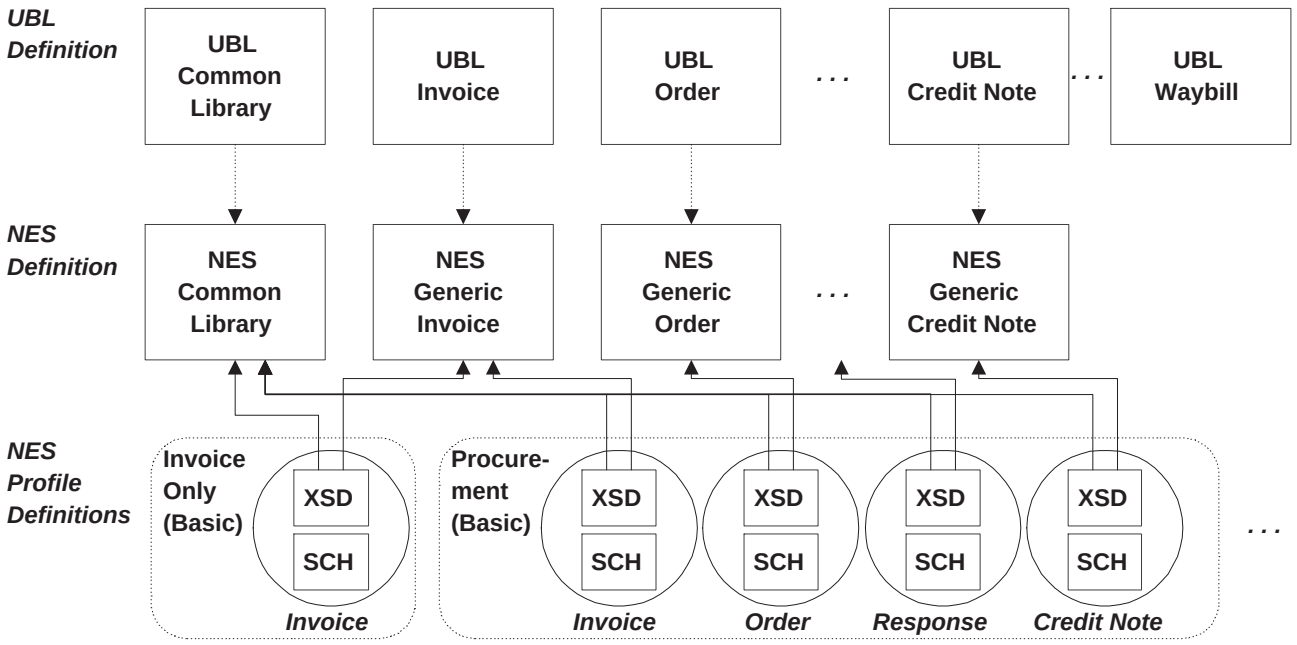
Customizations and profiles of UBL (cont.)

Layered approach to the library specification

- not all UBL document types are included in the NES specification
- the NES common library is a derivation of the UBL common library
- the NES document libraries are derivations of the UBL document libraries
- an NES document type has constructs from the NES common library and from the NES document library

Profiles of different definitions of the same document types

- different profiles of transactions, each with business rules and a set of document types
  - e.g. "invoice only" contains only the invoice document type
  - e.g. "basic billing" contains both invoice and credit note document types
  - the two invoice document types are different in each profile because of the demands of information exchange in the business flows
- ISO/IEC 19757-3 Schematron is layered on top of schemas for contextually limiting the document contents based on the profile
  - the Schematron schema asserts that for a given profile, an NES construct allowed in the document in other profiles is disallowed in the given profile's version of the document



Copyright © Crane Softwrights Ltd.

North European Subset (NES)

- `http://www.nesubl.eu/`
- governments of Denmark, Finland, Sweden, Norway, Iceland and England
- work products stopped being produced mid-2007
- all NES work is being migrated to the BII project

Profile 1: Catalogue Only

- Catalogue
- Application Response

Profile 2: Catalogue with Updates

- Catalogue
- Catalogue Item Specification Update
- Catalogue Pricing Update
- Application Response

Profile 3: Basic Order Only

- Order (basic)

Profile 4: Basic Invoice Only

- Invoice (basic)

Profile 5: Basic Billing

- Invoice (basic)
- Credit Note (basic) to match the transaction.

Profile 6: Basic Procurement

- Order (basic)
- Order Response Simple
- Invoice (basic)
- Credit Note (basic)

Profile 7: Simple Procurement

- Order (library level)
- Order Response Simple
- Invoice (library level)
- Credit Note (library level)
- Application Response

Profile 8: Basic Billing with Dispute Response

- Invoice (basic)
- Credit Note (basic)
- Application Response

Copyright © Crane Softwrights Ltd.

NES/BII customizations and profiles (cont.)

Business Interoperability Interfaces on public procurement in Europe (WS/BII)

- `http://www.en.ds.dk/bii`
- includes work from Spain CODICE project to cover pre-award processes and documents
- includes work of North European Subset to cover post-award processes and documents

Defined profiles with document types:

- each profile specifies the message flow (choreography)
- each profile specifies the requirements for the message content

Pre-award profiles

- Tender Notification Basic (BII14)
- Tender Notification Advanced (BII10)
- Pre-qualification and Tender Invitation (BII11)
- Tendering Simple (BII12)

Post-award profiles

- Catalogue only (BII01)
- Catalogue with update (BII02)
- Basic Order only (BII03)
- Basic Invoice only (BII04)
- Basic billing (BII05)
- Basic procurement (BII06)
- Simple procurement (BII07)
- Basic billing with dispute (BII08)
- Advanced Procurement with dispatch (BII13)
- Customs Bill (BII09)
- Scanned Invoice (BII15)
- Catalogue Deletion (BII16)
- Multi-Party Catalogue (BII17)
- Punch-out (BII18)
- Advanced Procurement (BII19)
- Quote request (BII20)
- Statement (BII21)

The above list is excerpted from the CEN WS-BII-WG1 phase 2 report - Report 1, Version 2.0 (undated).

Copyright © Crane Softwrights Ltd.

Chapter 3 - Information items

- Introduction - Information found in UBL documents
- Section 1 - Information definitions
- Section 2 - Information maintenance
- Section 3 - Crane's UBL information model reports

Outcomes

- understand the determination and documentation (meta data) of UBL information items and documents
- overview the components of the library of information and document models

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Information found in UBL documents

Committee members are responsible for information modeling of UBL document types

- the XML document modeling is not the direct responsibility of members
- XML document models are synthesized from the collaborative information models
- office software spreadsheets used for collaboratively developing the information models
- UML information models used for confirming the spreadsheet models
    - some members start with the UML and add to the spreadsheets from their work

UBL development started with the xCBL 3.0 information items

- along the way refined the concepts based on reviews and feedback from the public and from committee members
- applied the principles of syntax-neutral CCTS to the definition of an XML vocabulary
    - UBL is the first publicly-available royalty-free library of XML components defined using CCTS 2.01

UBL keeps the innovative concept of a library of common shared information items

- document types are not developed in isolation
- individual document types are built from components in a single library
- ensures semantic equivalence of constructs between document types because the constructs are the same (not just similar)

UBL uses CCTS distinction between basic (atomic) and aggregate (molecular) constructs

- XML structures are syntactic representations of semantic building blocks of CCTS

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

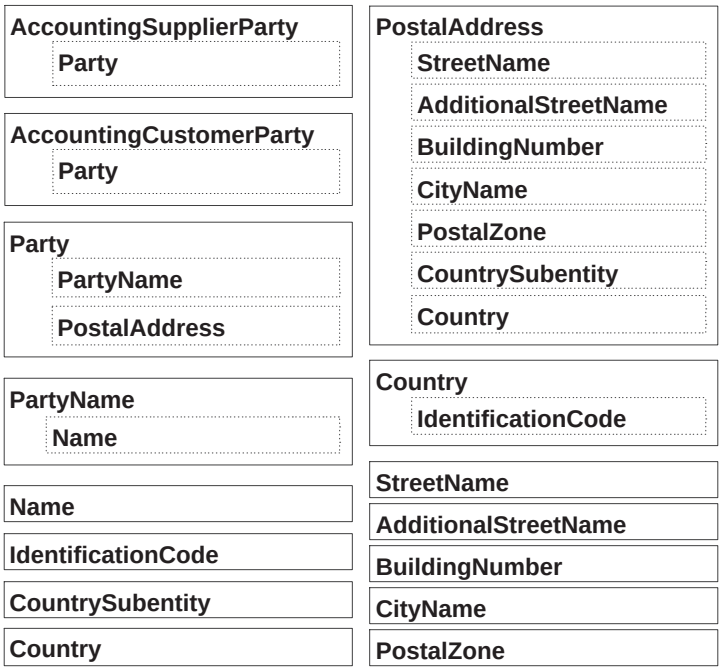
Copyright © Crane Softwrights Ltd.

Individual information entities are declared with their constituent components

- a declaration describes a parent/children or item/text relationship
  - a declaration of an item with children constrain the order and quantity of children
  - a declaration of an item without children constrains the characters used in the string of text
- one declaration per information item
  - a very flat set of declarations (no "grandchildren" are ever declared)

A view of the nesting of declarations using a simplified model illustrating only a few of the child constructs:

- a number of optional child constructs are omitted in this diagram, but the ones shown are those used in the associated markup example that follows

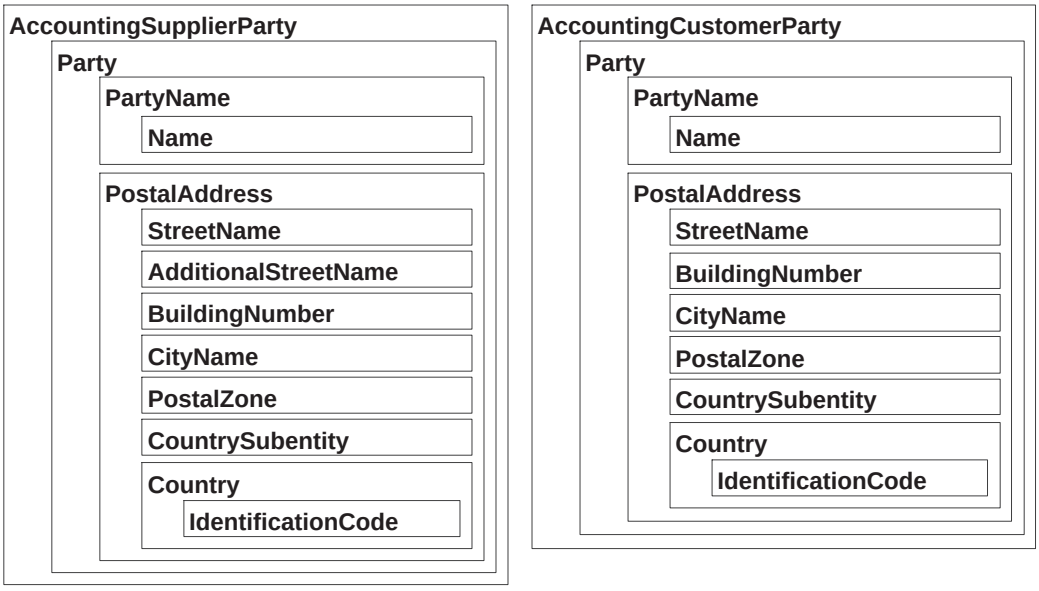


Copyright © Crane Softwrights Ltd.

Information organization (cont.)

Structured information is maintained in a tree organization of repeating parents

- the tree can be considered as upside down with the root above the top-most branch
  - branches and leaves reach down the diagram
  - sometimes the tree is drawn from side to side
    - the triangles in the diagrams of this material are meant to evoke the branching of structured information from left to right, with the root at the left and the leaves at the right
- branches of the tree are containers of containers
  - structural constraints on the order and quantity of child constructs
- leaves of the tree are containers of text
  - lexical constraints on the characters used in the string of text



Information items are used as many times as needed

- still only one declaration per item regardless of how often the item is used
  - e.g. the `StreetName` item exists twice in the collection above
    - once nested three levels deep under `AccountingSupplierParty`
    - once nested three levels deep under `AccountingCustomerParty`
- an instance is not obliged to have optional information items
  - e.g. the `AdditionalStreetName` is not a part of the address for the `AccountingCustomerParty`

Copyright © Crane Softwrights Ltd.

Information organization (cont.)

Each of the uses of an object describes a document context

- any given information item has a single document context from the outermost container through all of the nested containers to the item
- different use of the noun "context" for "document context" than for "business context"
  - structured information uses the term "document context" for the differing locations of information items
  - business process description uses the term "business context" for the differing applications of rules

The following is a subset of document contexts excerpted from an XPath text file

- defined formally in Chapter 7 - XPath enumerations (page 129)

```
01 AccountingSupplierParty
02 AccountingSupplierParty/Party
03 AccountingSupplierParty/Party/PartyName
04 AccountingSupplierParty/Party/PartyName/Name
05 AccountingSupplierParty/Party/PostalAddress
06 AccountingSupplierParty/Party/PostalAddress/StreetName
07 AccountingSupplierParty/Party/PostalAddress/AdditionalStreetName
08 AccountingSupplierParty/Party/PostalAddress/BuildingNumber
09 AccountingSupplierParty/Party/PostalAddress/CityName
10 AccountingSupplierParty/Party/PostalAddress/PostalZone
11 AccountingSupplierParty/Party/PostalAddress/CountrySubentity
12 AccountingSupplierParty/Party/PostalAddress/Country
13 AccountingSupplierParty/Party/PostalAddress/Country/IdentificationCode
14 AccountingCustomerParty
15 AccountingCustomerParty/Party
16 AccountingCustomerParty/Party/PartyName
17 AccountingCustomerParty/Party/PartyName/Name
18 AccountingCustomerParty/Party/PostalAddress
19 AccountingCustomerParty/Party/PostalAddress/StreetName
20 AccountingCustomerParty/Party/PostalAddress/BuildingNumber
21 AccountingCustomerParty/Party/PostalAddress/CityName
22 AccountingCustomerParty/Party/PostalAddress/PostalZone
23 AccountingCustomerParty/Party/PostalAddress/CountrySubentity
24 AccountingCustomerParty/Party/PostalAddress/Country
25 AccountingCustomerParty/Party/PostalAddress/Country/IdentificationCode
```


Copyright © Crane Softwrights Ltd.

Information organization (cont.)

The XML markup reflects the nesting of containers

- corresponding to the uses of the information items in each document context
- note that the indentation is irrelevant

```
<cac:AccountingSupplierParty>
 <cac:Party>
  <cac:PartyName>
   <cbc:Name>Consortial</cbc:Name>
  </cac:PartyName>
  <cac:PostalAddress>
   <cbc:StreetName>Busy Street</cbc:StreetName>
   <cbc:AdditionalStreetName>East of Main</cbc:AdditionalStreetName>
   <cbc:BuildingNumber>56A</cbc:BuildingNumber>
   <cbc:CityName>Farthing</cbc:CityName>
   <cbc:PostalZone>AA99 1BB</cbc:PostalZone>
   <cbc:CountrySubentity>Heremouthshire</cbc:CountrySubentity>
   <cac:Country>
    <cbc:IdentificationCode>GB</cbc:IdentificationCode>
   </cac:Country>
  </cac:PostalAddress>
 </cac:Party>
</cac:AccountingSupplierParty>
<cac:AccountingCustomerParty>
 <cac:Party>
  <cac:PartyName>
   <cbc:Name>IYT Corporation</cbc:Name>
  </cac:PartyName>
  <cac:PostalAddress>
   <cbc:StreetName>Avon Way</cbc:StreetName>
   <cbc:BuildingNumber>56A</cbc:BuildingNumber>
   <cbc:CityName>Bridgtow</cbc:CityName>
   <cbc:PostalZone>ZZ99 1ZZ</cbc:PostalZone>
   <cbc:CountrySubentity>Avon</cbc:CountrySubentity>
   <cac:Country>
    <cbc:IdentificationCode>GB</cbc:IdentificationCode>
   </cac:Country>
  </cac:PostalAddress>
 </cac:Party>
</cac:AccountingCustomerParty>
```


Copyright © Crane Softwrights Ltd.

UBL is the first published set of publicly-available XML schemas to express in XML syntax instantiations of the syntax-neutral ebXML core components (CCTS v2.01)

- core component types
  - the set of information types from which all information types are derived
    - e.g. currency amount, text, numeric, date, etc.
  - "supplementary components" are the meta data of the value in the instance
    - e.g. currency of the currency amount
- core components
  - building blocks for semantic description of a specific concept
  - aggregate core component
    - collection of related pieces of business information
    - representation of an object class
  - basic core component
    - singular business characteristic of an aggregate core component
  - association core component
    - complex business characteristic of an aggregate core component
    - represents the use of an aggregate core component
- business information entities
  - building blocks for business data
  - aggregate business information entity (ABIE)
    - collection of related pieces of business information
    - representation of an object class
  - basic business information entity (BBIE)
    - singular business characteristic of an aggregate business information entity
  - association business information entity (ASBIE)
    - complex business characteristic of an aggregate business information entity
    - represents the use of an aggregate business information entity

The association construct is the mechanism by which an aggregate construct is referenced as part of another aggregate construct

- a distinction is made in modeling so as not to confuse the role of an aggregate definition
- the aggregate entry in the model is the definition of the aggregate
- the association entry in the model is a reference to the aggregate

Declarations exist for only ABIE and BBIE entities

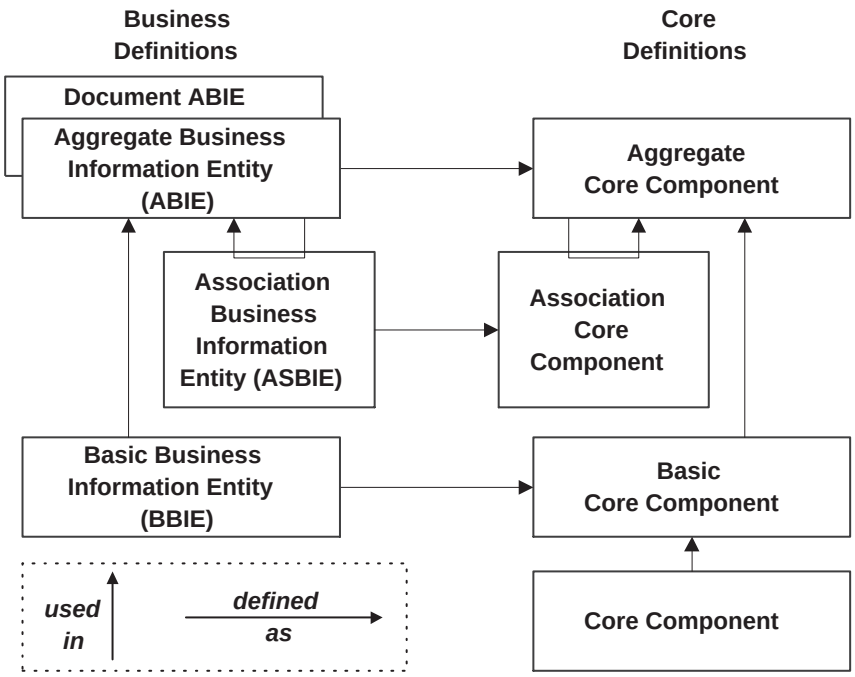
- ASBIE entities in ABIE entities reference other ABIE entities

Copyright © Crane Softwrights Ltd.

Basis of interoperability (cont.)

Derivation of business information entities from core components:

- note that the "Document ABIE" is the outermost ABIE that is not referenced as an ASBIE from any other ABIE
  - e.g. "Invoice"



Examples of UBL business information entities

- `<cac:Party>` defined by the ABIE named "`Party. Details`"
  - contains `<cac:PostalAddress>` defined by the ASBIE named "`Party. Postal_ Address. Address`" which references the ABIE named "`Address. Details`"
- `<cac:PostalAddress>` defined by the ABIE named "`Address. Details`"
  - contains `<cbc:AdditionalStreetName>` defined by the BBIE named "`Address. Additional_ Street Name. Name`"
- recall the markup in Information organization (page 66) used to represent these items

Copyright © Crane Softwrights Ltd.

- version 2.01 of CCTS in play when UBL 2.0 development began

CCTS prescribes the description of a component to have the following information:

- without including anything about syntax
    - CCTS abstract concepts are designed to be realized in a concrete syntax
- dictionary entry name ("DEN") conventions based on ISO/IEC 11179 Part 5
    - Naming and identification Principles for data elements
        - name built on three components
            - usually: object, function, type
            - also: activity, characteristic, form
    - `Qual_ Object Class. Qual_ Property Term. Representation Term`
    - e.g. "`Address. Additional_ Street Name. Name`"
    - name structure:
        - usually: (Qualifier)_ (Object). (Qualifier)_ (Function). (Type)
        - also: (Qualifier)_ (Activity). (Qualifier)_ (Characteristic). (Form)
    - three name major components:
        - object class/activity: logical data grouping or aggregation
        - function/characteristic: distinguishing property of the object class
        - type/form: describes the shape of the information at a lexical text-string level
        - some components may be qualified
    - dictionary entries in English using the Oxford English Dictionary
- definition
    - unique business semantic of the core component
- cardinality
    - mandatory/optional
    - singleton/repeatable
    - four combinations represented symbolically by postfix Kleene operators
        - mandatory and not repeatable - `1` - (no Kleene postfix operator)
        - optional and not repeatable - `0..1` - "`?`"
        - mandatory and repeatable - `1..n` - "`+`"
        - optional and repeatable - `0..n` - "`*`"
    - note that while a cardinality may be explicitly repeatable - e.g. `0..7` for up to 7 uses of the item - this is not used in UBL
        - regarded as a facet of business rule checking to check arbitrary conditions
            - e.g. arbitrary cardinality
            - e.g. field length
- business terms (optional)
    - commonly-known synonym term used in business
    - there may be several terms or synonyms for a single core component

Copyright © Crane Softwrights Ltd.

UBL definitions

UBL component definition (derived from CCTS definitions (page 69)):

- UBL name (e.g. "`AdditionalStreetName`")
  - calculated from the dictionary entry name
  - object class and punctuation are dropped
  - some abbreviations (e.g. "`ID`" for "Identifier")
  - rules remove duplicate parts of a complete name
    - e.g. when the representation term is the same as the end of the property term
- definition in prose
  - written in English
  - see Chapter 6 - Model semantics (page 122) regarding translations
- component type
  - BBIE, ASBIE or ABIE
- dictionary entry name (e.g. "`Address. Additional_ Street Name. Name`")
  - object class portion
    - class qualifier (optional)
      - in the end this was never used by the UBL TC
    - class name (e.g. "`Address.`")
  - property term portion (e.g. "`Additional_ Street Name.`")
    - property term qualifier (optional) (e.g. "`Additional_`")
    - property term (e.g. "`Street Name.`")
      - property term possessive noun (e.g. "`Street`")
      - property term primary noun (e.g. "`Name.`")
  - representation term portion (e.g. "`Name`")
    - i.e. this information item "is represented by an amount value"
    - e.g. amount (currency), identifier, text, name, etc.
    - the representation term isn't supposed to convey any semantics or meaning
    - aggregate items use the representation term "`Details`"
- qualified data type
  - data type qualifier (optional)
  - data type (e.g. "`Name. Type`")
  - describes the set of valid values for a BBIE
  - uses the representation term
- cardinality (e.g. "`0..1`")
  - how often the information item occurs
- business terms (e.g. "thoroughfare")
- example value (e.g. "Corner of Aberdeen Road")

Copyright © Crane Softwrights Ltd.

UBL definitions (cont.)

Object class + Property term = sufficient recognition of an item

- doesn't replace the definition but should be distinctly recognizable for its purpose
    - the associated definition is the formal normative definition for the item
- Representation term only identifies the structure of the item
    - only used when distinguishing the property term as being a particular structure of a different name
    - an absent representation term infers that the representation term is identical to the property term

Important rule of thumb when developing compatible customizations

- when creating new elements that are not elements in UBL
- dictionary entry naming techniques should follow practices followed by the UBL committee
    - see Dictionary entry naming (page 202) for more detail
- UBL naming is automated from the dictionary entry name

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL information item constraints

Aggregate items (ABIE) are defined as a sequence of child items each with cardinality

- BBIE children have simple text values
    - text values are of a particular core component data type
- ASBIE children are ABIE items with item children
    - no mixed content in any UBL information item
        - i.e. no mixing of items and sibling text
        - e.g. no "bold" or "italic" constructs for highlighting content

BBIE data types restrict the text expression of the value

- amount (currency)
- binary object type
- code
- date
- identifier
- indicator
- measure
- name
- numeric
- percent
- quantity
- rate
- text
- time
- others defined in the CCTS standardized set of unqualified core component data types are not used in UBL 2.0

No cultural preferences are allowed in numeric values

- decimal separator is always "."
- no thousands punctuation
- no other punctuation
    - currency is indicated using a supplementary component and is expressed using an attribute

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL information item constraints (cont.)

Coded-value item and identifier-value item values are mnemonic or representative of semantics or lookup values

- information items defined by values from code lists or identifier lists are popular in many business documents
- a single value is expressed as a sequence of characters possibly with embedded single spaces
- also known as "controlled vocabularies"

A coded-value item's data type is either qualified (when a code list has been suggested by the UBL TC) or unqualified (when a code list has not been suggested by the UBL TC)

- initial values are provided by UBL TC for qualified coded value data types
    - not a normative aspect of the UBL definition
- a predefined set of values available only to be restricted
    - e.g. `Currency_ Code. Type`
        - initially defined set of values to be all currencies of the world
- a predefined set of values available to be supplemented or restricted
    - e.g. `Document Status_ Code. Type`
        - values "`Cancelled`", "`Disputed`", "`NoStatus`", and "`Revised`"
    - e.g. `Payment Means_ Code. Type`
        - values "`10`" (cash), "`20`" (cheque), etc.
- a limited set of values unlikely to be changed
    - e.g. `Latitude Direction_ Code. Type`
        - values "`North`" and "`South`"
    - e.g. `Longitude Direction_ Code. Type`
        - values "`East`" and "`West`"
- no defined values but available to be defined between trading partners
    - e.g. `Code. Type` for `Invoice. Invoice_ Type. Code`

None of the identifier types have any initial values suggested by the committee

- e.g. account numbers, customer numbers, etc.

Copyright © Crane Softwrights Ltd.

ABIE re-use report

Chapter 3 - Information items
Section 2 - Information maintenance

`http://docs.oasis-open.org/ubl/os-UBL-2.0/etc/UBL-ABIE-Reuse-Table-2.0.ods`

`http://docs.oasis-open.org/ubl/os-UBL-2.0/etc/UBL-ABIE-Reuse-Table-2.0.xls`

Summary cross-reference report of the use and re-use of aggregate business information entities:

- most aggregate items use other aggregate items through associated proxies
- re-use report summarizes the aggregate items used by other aggregate items and the aggregate items using other aggregate items
- does not summarize the use of basic items

Copyright © Crane Softwrights Ltd.

Library of shared and specific information items

UBL maintains two collections of sets of information items

- the top-level document type ABIE items are separate from the common lower-level BBIE items and the ABIE items used as ASBIE items
- recall the diagram on page 68 distinguishing the document ABIE items

A collection of document type sets of items that make use of the shared libraries

- the collection is referred to as "`maindoc`"
- one set of information item definitions per document type
    - only one top-level document ABIE information item per document type
- the document type ABIE items are not used anywhere as ASBIE items

A collection of shared libraries of items

- the collection is referred to as "`common`"
- one library includes the definition of all BBIE items and ABIE items that are used as ASBIE items
    - the UBL 2.0 common library defines 113 ABIE items used by document types and other library items
- one module includes the declarations of qualified data types
    - meta data for UBL-suggested qualified code lists
    - no code list values

An office software spreadsheet is used to maintain the details of items in a set

- each row of a spreadsheet maintains one ABIE, BBIE or ASBIE information item
- the columns of the spreadsheet describe the properties of each information item
- one of the two normative components of UBL is the spreadsheets used to maintain the definitions
    - the focus of collaboration is on the information being maintained by UBL
    - the spreadsheets are not normative as they only determine and govern the XML structures but they do not, in and of themselves, express the mechanics of the XML structures
        - the focus of UBL standardization is the interchange of information in an XML document
        - the semantics and the collaboration guide the determination of the interchange but do not define the actual interchange syntax
    - all aspects of interchange syntax and XML message construction are synthesized from the collaboration stylesheets
        - the other normative component of UBL is the set of W3C Schema XSD expressions to be described on page 98
- the spreadsheets are input to automated processes that synthesize the formal document model constraint schema syntax
    - not all column values are copied into the schema
    - columns with yellow headers are copied, columns with gray headers are not

Copyright © Crane Softwrights Ltd.

Library of shared and specific information items (cont.)

`http://docs.oasis-open.org/ubl/os-UBL-2.0/mod/maindoc` directory

- one module of information items for each specific document type and document ABIE
- each document type makes reference into the shared library for common components
    - unique components are defined for each document type
- `UBL-ApplicationResponse`
- `UBL-AttachedDocument`
- `UBL-BillOfLading`
- `UBL-Catalogue`
- `UBL-CatalogueDeletion`
- `UBL-CatalogueItemSpecificationUpdate`
- `UBL-CataloguePricingUpdate`
- `UBL-CatalogueRequest`
- `UBL-CertificateOf Origin`
- `UBL-CreditNote`
- `UBL-DebitNote`
- `UBL-DespatchAdvice`
- `UBL-ForwardingInstruction`
- `UBL-FreightInvoice`
- `UBL-Invoice`
- `UBL-Order`
- `UBL-OrderCancellation`
- `UBL-OrderChange`
- `UBL-OrderResponse`
- `UBL-OrderResponseSimple`
- `UBL-PackingList`
- `UBL-Quotation`
- `UBL-ReceiptAdvice`
- `UBL-Reminder`
- `UBL-RemittanceAdvice`
- `UBL-RequestForQuotation`
- `UBL-SelfBilledCreditNote`
- `UBL-SelfBilledInvoice`
- `UBL-Statement`
- `UBL-TransportationStatus`
- `UBL-Waybill`

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Library of shared and specific information items (cont.)

`http://docs.oasis-open.org/ubl/os-UBL-2.0/mod/common` directory

- information items common to all document types
- `UBL-CommonLibrary`
    - parties, contacts, addresses, locations, dimensions, tax information, etc.
- `UBL-qDT`
    - "qualified data types"
    - definitions of values for code list meta data for those code lists for which the UBL TC provides values
        - `AllowanceReasonCode`
        - `ChannelCode`
        - `ChipCode`
        - `ContainerSizeTypeCode`
        - `CountryIdentificationCode`
        - `CurrencyCode`
        - `DocumentStatusCode`
        - `LatitudeDirectionCode`
        - `LineStatusCode`
        - `LongitudeDirectionCode`
        - `OperatorCode`
        - `PackagingTypeCode`
        - `PaymentMeansCode`
        - `PortCode`
        - `SubstitutionStatusCode`
        - `TransportationStatusCode`
        - `TransportEquipmentTypeCode`
        - `TransportModeCode`
        - `UnitOfMeasureCode`
    - note there isn't a qualified data type for `BinaryObjectMimeCode` yet there is such a code list
        - this is an unqualified supplementary component of the `BinaryObjectType` defined by UN/CEFACT

Copyright © Crane Softwrights Ltd.

UBL information model spreadsheets

Chapter 3 - Information items
Section 2 - Information maintenance

Two spreadsheet documents capture information regarding each single shared or specific library collection:

- same information in both spreadsheets, only different file formats for different tools
- useful for collaboration between business experts who are not document or model experts
- `.ods` - Open Document Format (ODF) Spreadsheet
- `.xls` - Microsoft Excel Spreadsheet

Color conventions

- magenta row - ABIE
- white row - BBIE
- cyan row - ASBIE
- yellow-headed column - value copied to embedded schema documentation

Example from `UBL-CommonLibrary` (e.g. row 10 has `AdditionalStreetName`):

	A	B	C	D	E	F	G
1	UBL Name	Dictionary Entry Name	Object Class Qualifier	Object Class	Property Term Qualifier	Property Term Possessive Noun	Property Term Primary Noun
2	Address	Address. Details		Address			
3	ID	Address. Identifier		Address			Identifier
4	AddressTypeCode	Address. Address Type Code. Code		Address		Address Type	Code
5	AddressFormatCode	Address. Address Format Code. Code		Address		Address Format	Code
6	Postbox	Address. Postbox. Text		Address			Postbox
7	Floor	Address. Floor. Text		Address			Floor
8	Room	Address. Room. Text		Address			Room
9	StreetName	Address. Street Name. Name		Address		Street	Name
10	AdditionalStreetName	Address. Additional_ Street Name. Name		Address	Additional	Street	Name
11	BlockName	Address. Block Name. Name		Address		Block	Name
12	BuildingName	Address. Building Name. Name		Address		Building	Name
13	BuildingNumber	Address. Building Number. Text		Address		Building	Number
14	InhouseMail	Address. Inhouse_ Mail. Text		Address	Inhouse		Mail
15	Department	Address. Department. Text		Address			Department
16	MarkAttention	Address. Mark Attention. Text		Address			Mark Attention
17	MarkCare	Address. Mark Care. Text		Address			Mark Care
18	PlotIdentification	Address. Plot Identification. Text		Address		Plot	Identification
19	CitySubdivisionName	Address. City Subdivision Name. Name		Address		City Subdivision	Name
20	CityName	Address. City Name. Name		Address		City	Name
21	PostalZone	Address. Postal_ Zone. Text		Address	Postal		Zone
22	CountrySubentity	Address. Country Subentity. Text		Address		Country	Subentity
23	CountrySubentityCode	Address. Country Subentity Code. Code		Address		Country Subentity	Code
24	Region	Address. Region. Text		Address			Region
25	District	Address. District. Text		Address			District
26	TimezoneOffset	Address. Timezone Offset. Text		Address		Timezone	Offset
27	AddressLine	Address. Address Line		Address			
28	Country	Address. Country		Address			
29	LocationCoordinate	Address. Location Coordinate		Address			
30	AddressLine	Address Line. Details		Address			
31	Line	Address Line. Line. Text		Address Line			Line
32	AirTransport	Air Transport. Details		Air			
33	AircraftID	Air Transport. Aircraft Identifier. Identifier		Air Transport		Aircraft	Identifier
34	AllowanceCharge	Allowance Charge. Details		Allowance			
35	ID	Allowance Charge. Identifier		Allowance Charge			Identifier
36	ChargeIndicator	Allowance Charge. Charge. Indicator		Allowance Charge			Indicator

Copyright © Crane Softwrights Ltd.

UBL information model spreadsheets (cont.)

Rows ordered by object class

- no specific order within object class

Columns of interest:

Informal mnemonic

- UBL Name

Formal name

- Dictionary Entry Name
- Object Class Qualifier
- Object Class
- Property Term Qualifier
- Property Term Possessive Noun
- Property Term Primary Noun
- Property Term
- Representation Term
- Data Type Qualifier
- Data Type
- Associated Object Class Qualifier
- Associated Object Class

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL information model spreadsheets (cont.)

Columns of interest (cont.):

Properties

- Alternative Business Terms
    - other simple names by which the construct is known
- Cardinality
    - how often the item is allowed to occur
- Component Type
    - BBIE, ASBIE or ABIE
- Definition
    - natural-language description
- Examples
    - representative sample values
- Current Version
    - when introduced into UBL use

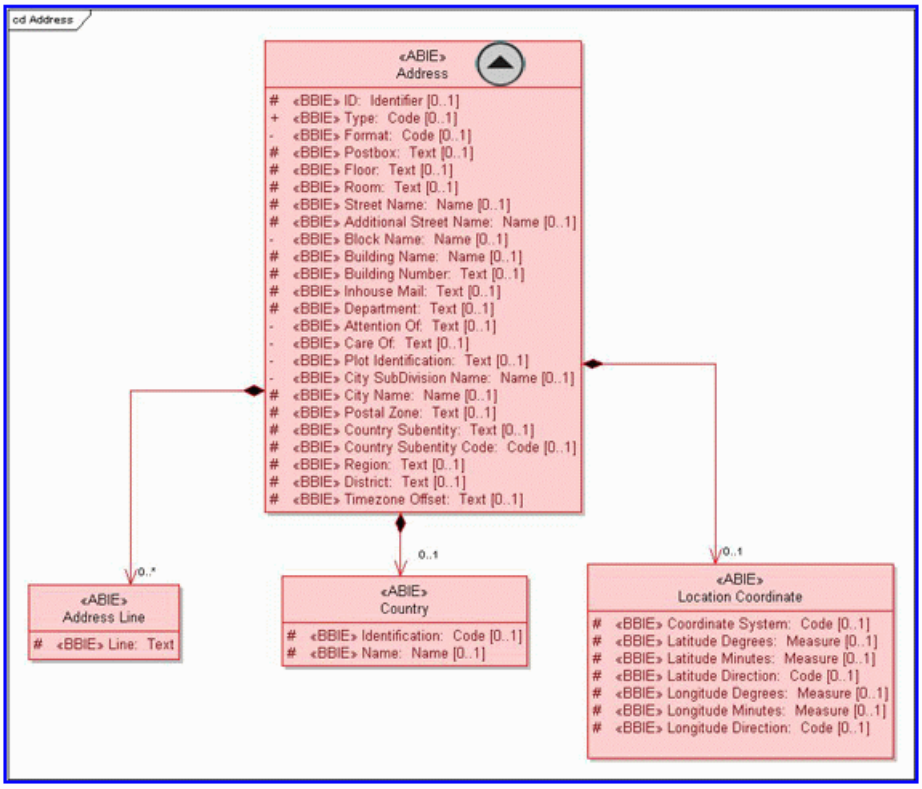
Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL UML information models

Chapter 3 - Information items
Section 2 - Information maintenance

Unified Modeling Language (UML) expressions of the information models are found in `http://docs.oasis-open.org/ubl/os-UBL-2.0/uml/`

- standalone JPEG files illustrating the structure diagrams
- interlinked HTML files that reference the JPG files



Copyright © Crane Softwrights Ltd.

UBL UML information models (cont.)

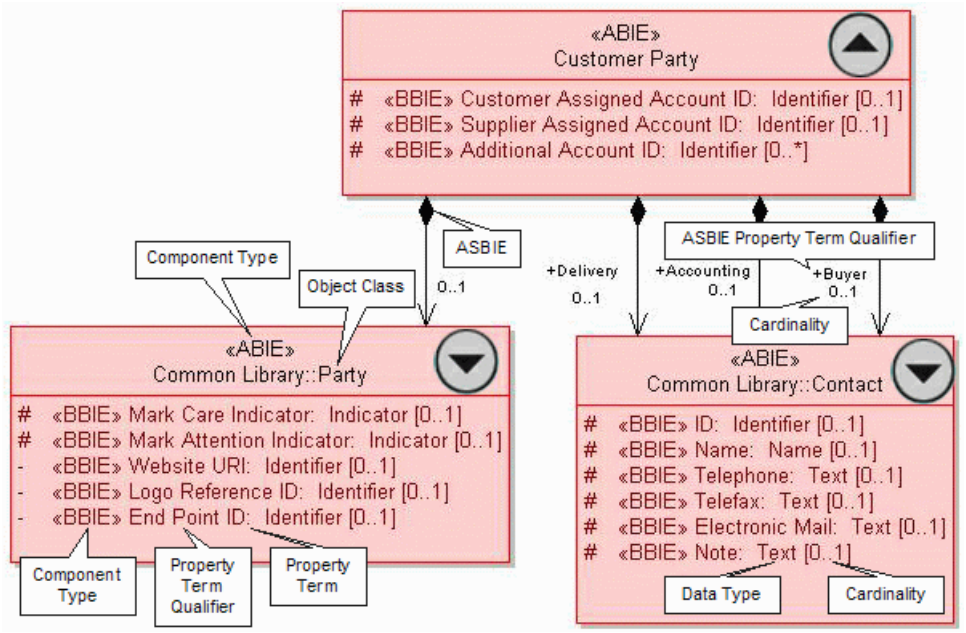
Chapter 3 - Information items
Section 2 - Information maintenance

UML drawing conventions

Compare the rows in the spreadsheet for `CustomerParty`:

	A	E	F	G	H	I	J	K
1	UBL Name	Property Term Qualifier	Property Term Possessive Noun	Property Term Primary Noun	Property Term	Representation Term	Data Type Qualifier	Data Type
262	CustomerParty							
263	CustomerAssignedAccountID	Customer Assigned	Account	Identifier	Account Identifier	Identifier		Identifier. Type
264	SupplierAssignedAccountID	Supplier Assigned	Account	Identifier	Account Identifier	Identifier		Identifier. Type
265	AdditionalAccountID	Additional	Account	Identifier	Account Identifier	Identifier		Identifier. Type
266	Party				Party	Party		
267	DeliveryContact	Delivery			Contact	Contact		
268	AccountingContact	Accounting			Contact	Contact		
269	BuyerContact	Buyer			Contact	Contact		

With the matching UML drawing for `CustomerParty`:



Copyright © Crane Softwrights Ltd.

Crane's UBL information model reports

Crane Softwrights Ltd. has created an aggregated report from all document models into a single file to be viewed from a web browser

- `http://www.CraneSoftwrights.com/resources/ubl/#ubl2modelreport`
- `Crane-UBL2Reports/EN/Crane-UBL2Report-All-EN.html` - 5Mb (complete; all document types in one)
- `Crane-UBL2Reports/EN/Crane-UBL2Report-???????-EN.html` - individual document types
- alphabetically indexed by UBL Name and document type
- hyperlinked to model row reproduction in an HTML table (juggled columns)
- hyperlinked to ABIE for each ASBIE, and row numbers to index definitions
- available in all languages of the IDD

Multilingual versions of the report

- the suffix is the two-character language indicator
- the only columns translated are the definition and common business terms

Report has three main sections

- front matter (indexes, meta data, etc.)
- alphabetized initial two-letters of each component
- alphabetized link to the model collections included in the report

Trimmed common library component rows

- only those common library rows that are used directly or indirectly for the document types are included
- this improves the traversal through the model as unused rows do not distract the reader
    - e.g. the catalogue-related rows are not included directly or indirectly in the invoice report
- in the report for a customized schema, the trimming includes all unused constructs

Rules of thumb for navigation

- hyperlinked row numbers switch between summary and table views
- hyperlinked UBL names stay within the summary and table views

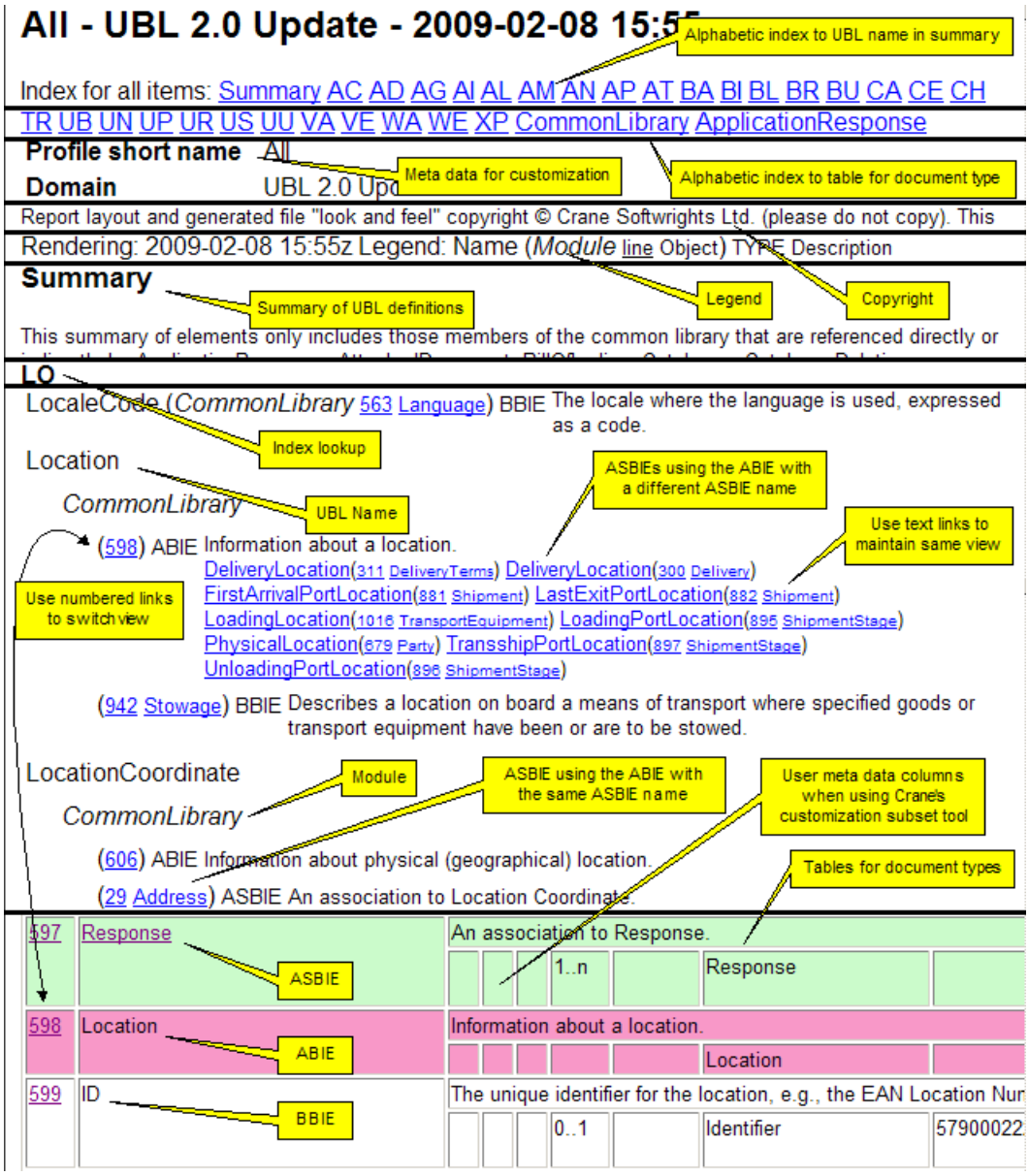
Created by Crane's schema subset configuration tool

- see Annex A - OpenOffice 3 UBL customization environment (page 229)
- the same report is generated for an arbitrary customization of the schemas

Copyright © Crane Softwrights Ltd.

Crane's UBL information model reports (cont.)

Chapter 3 - Information items
Section 3 - Crane's UBL information model reports



Copyright © Crane Softwrights Ltd.

Chapter 4 - Naming and design rules (NDR)

- Introduction - Formal naming and design rules
- Section 1 - Converting the information models to document models

Outcomes

- understanding the role and applicability of the naming and design rules

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL model development is only in the information models, not the document models

- using spreadsheets and UML models as the basis for collaboration
- models for different document types
- model for common library of components

A document model constraints the labels used for information items in XML documents

- a number of XML constraint languages are available to be used
- the UBL TC chose the W3C XML Schema (XSD) constraint language

UBL document models correspond to the document types in the information models

- XML documents are constrained by formal model expressions of constraints
  - a given expression of constraints is a schema
  - a schema is written in a schema language expressing the semantics of a given schema validation process
- information models in spreadsheet form are translated into document models in a schema language

Copyright © Crane Softwrights Ltd.

Formal naming and design rules (cont.)

Naming and Design Rules (NDR) are expressed as rules for XSD schema generation that can be automated

- enables but does not oblige the rules to be applied using automation
- automated tools can be rules-driven in synthesizing the schema fragments

Document extension is an aspect of schema generation, not model design

- the information models say nothing at all about extension mechanisms in the schema fragments
- the `UBLExtensions` element and its descendants are specified as schema facilities for all document types

Downstream customization processes can rely on schema expressions to follow UBL NDR

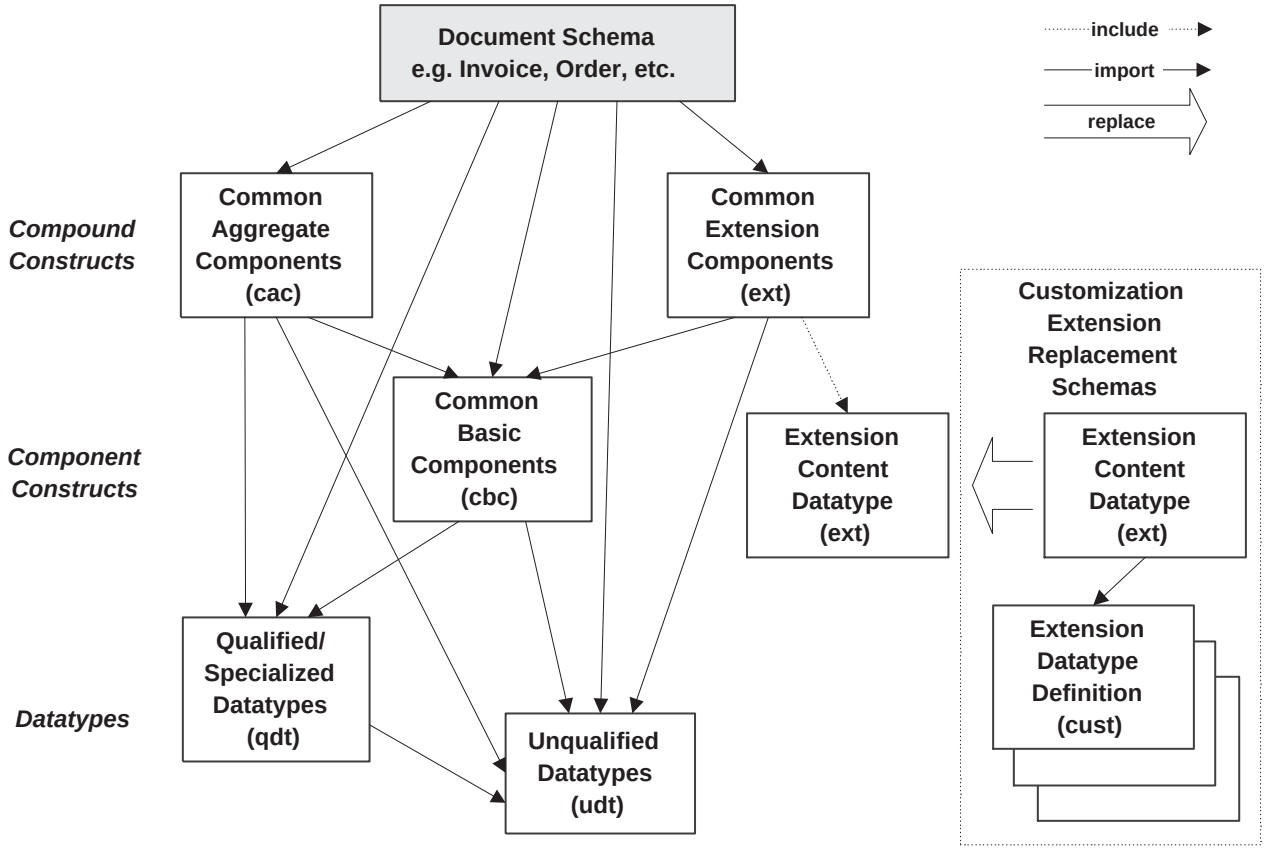
- straightforward processing of expressions can be accomplished because of particular choices of rules
- breaking the rules prevents the downstream processes from being able to extract the required information out of the models

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Formal naming and design rules (cont.)

This illustrates the schema import and include dependencies described by the NDR:

- `<xsd:include>` must be used when the namespace URI string does not change
- `<xsd:import>` must be used when the namespace URI string changes
- the parentheses surround the namespace prefix used by convention by the UBL TC



Copyright © Crane Softwrights Ltd.

NDR document

Chapter 4 - Naming and design rules (NDR)
Section 1 - Converting the information models to document models

`http://docs.oasis-open.org/ubl/prd-UBL-NDR-2.0.doc`

Sections:

- Introduction and Guiding Principles
- Relationship to ebXML Core Components
- Overall Schema Structure
- Naming and Modeling Constraints
- Reusability Scheme
- Extension Scheme
- Namespace Scheme
- Versioning Scheme
- Modularity Strategy
- Annotation and Documentation Requirements
- Naming Rules
- Declarations and Definitions
- Code Lists

Not delivered as part of UBL

- separate specification
- referenced by other organizations for their naming and design rules
    - e.g. US Department of Navy

Copyright © Crane Softwrights Ltd.

NDR rule groupings

Rules are grouped by a prefix indicating the grouping:

- ATD - Attribute Declaration
- ATN - Attribute Naming
- CDL - Code List
- CTD - `ComplexType` Definition
- DOC - Documentation
- ELD - Element Declaration
- ELN - Element Naming
- GNR - General Naming
- GTD - General Type Definition
- GXS - General XML Schema
- IND - Instance Document
- MDC - Modeling Constraints
- NMC - Naming Constraints
- NMS - Namespace
- RED - Root Element Declaration
- SSM - Schema Structure Modularity
- STA - Standards Adherence
- VER - Versioning

Rules are found throughout the different sections of the document

- perform a search on the group prefix and rule number in order to find a given rule

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

NDR rule groupings (cont.)

Some example rules

- all information items use Oxford English (rule GNR1)
- all elements globally defined (rule ELD2)
- complex types are declared for each ABIE (rule CTN1) and BBIE (rule CTN2)
- all non-document-element elements declared in CAC or CBC schema modules (ELD10)
- every ABIE must use sequence groups (rule CTD2)

Important effects of the rules enabling downstream processing

- no anonymous types
- no locally-defined elements
- no choice groups
- though the ease of downstream processing was not necessarily a driving factor in selecting the NDR

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

NDR checklist

`http://docs.oasis-open.org/ubl/os-UBL-2.0/doc/ndr/NDR-checklist.pdf`

Summary report:

- enumeration of the NDR rules as a checklist of adherence
- rules in place at the time the schemata were developed

Copyright © Crane Softwrights Ltd.

Commercial tools for implementing UBL NDR

These tools have come to the attention of the UBL TC that create document models from information models using the UBL NDR

- also useful tools for viewing the components of the information models

GEFEG.FX

- created and marketed by GEFEG mbH in Germany
- `http://www.gefeg.com/index.htm`
    - "Downloads" at left creates a new timed session for obtaining software
    - "UBL Reader" at left refreshes the data entry form for use with the UBL models
    - fill out contact information
    - receive key in electronic mail

UBLer

- created and distributed by Invinet Systems in Spain
- `http://lists.oasis-open.org/archives/ubl/200612/msg00030.html`

UBLish

- created and distributed by SoftML in Singapore
- `http://www.softml.net/jedi/ubl/sw/UBLish/UBLish-1.0/index.html`
- application is free for downloading and requires use of the XPS engine
    - `http://www.softml.net/xps/`

Copyright © Crane Softwrights Ltd.

Document extension mechanism

Document extension is an aspect of schema synthesis, not information modeling

- recall the schema hierarchy shown on page 88
- every document schema imports the definition of the "extension point"
- the extension point is not found in the spreadsheets
    - not part of the information model
- the extension point includes the definition of the extension point content
- the supplied extension point content is unconstrained allowing any information
- any specific definition of the extension point content can replace the supplied extension point content

Every document model can have additional information added under the extension point

- unconstrained from a UBL perspective, so any values allowed
    - except the apex of the extension content must be in a non-UBL namespace
- communities of users defining common extensions can constrain the extension point with a formal set of constraints in the definition of the extension point content

All extension content is encountered in advance of standardized content

- irrelevant to applications accessing the content in random order
- important to applications accessing the content in serial order
    - by encountering all extensions first, there is no need in a serial processing application to hold off acting on standardized content in anticipation of subsequent extended content
    - all extension information can be recognized and accommodated in advance of encountering the standardized content

Developing extension content is an aspect of UBL customization

- see Chapter 13 - Customization extension (page 204) for more information

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Abbreviations

The following abbreviations must be used when creating element and attribute names:

Word sequence	*Abbreviation*
Credit Card Verification Numbering System	CV2
Identifier	ID
Uniform Resource Identifier	URI
United Nations Dangerous Goods	UNDG
Universal Business Language	UBL
Universally Unique Identifier	UUID

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 5 - Documents and document models

- Introduction - Document model formal expressions
- Section 1 - ASN.1 document models
- Section 2 - XML documents and document models

Outcomes

- be introduced to the ASN.1 models
- review how UBL specifies the use of XML documents
- understand the organization of the XML schema files

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document model formal expressions

A UBL document is an XML instance that satisfies constraints chosen by the UBL TC

- structural and lexical constraints are described by a document model
- value constraints are described through supplemental declarative files
    - see Chapter 8 - Controlled vocabulary overview (page 147) for overview
- instance constraints are described in the specification

A document model is a syntax formalism

- for machine processing of the constraints of information model components in a physical expression of a sequence of characters
- two different expressions of the document model based on the form of the documents
    - determined by the kinds of networks across which the documents are transmitted
- compact binary-encoded documents
    - ASN.1 ISO 8825
- marked-up text-based documents
    - W3C Schema XSD
- note that the XML constraints are normative while the ASN.1 constraints are not normative

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Document model formal expressions (cont.)

The document model constrains the interchange of information, not the application data model

- a popular misconception in XML-based system design is to equate the document model with the application data model
- XML provides independence between the two data models of the applications performing an interchange
- an application translates information from its data model to and from the interchange model

One of two normative components of UBL is the W3C Schema XSD expression of XML document constraints

- formal expression of document constraints expressed for automated processing
- two instantiations of the document constraints:
    - one with comments
        - spreadsheet model information copied for reference purposes
    - one without comments
        - streamlined file may be processed more quickly by some applications
        - referenced as the "run time versions" of the schemas
- the other normative component is the definitions in the spreadsheets
    - recall the discussion on page 75

The UBL TC has mandated additional normative instance-oriented constraints that go beyond the formal document model

- this promotes interoperability and blind interchange of XML documents
- some of these constraints are difficult or impossible to be checked using XML-based tools

Copyright © Crane Softwrights Ltd.

ASN.1 documents and document models

`http://docs.oasis-open.org/ubl/os-UBL-2.0/asn/` directory

- ASN.1 expressions of document models
  - ASN.1 specification of the content
  - ASN.1 schema
- one module per document type

Abstract Syntax Notation number One

- internationally-standardized binary encoding of arbitrary stream of nested hierarchical information
  - formalism for the specification of abstract data types
  - `http://asn1.elibel.tm.fr`
- reduces bandwidth use to 20% or even 10% of original size
- very broad use over wireless networks

Packed Encoding Rules (PER)

- well-defined lossless transformation from XML to PER to XML
- compiler can create a C binding to XML through PER

The UBL ASN.1 document models are programmatically derived from the XML document models

Copyright © Crane Softwrights Ltd.

XML documents

Chapter 5 - Documents and document models
Section 2 - XML documents and document models

XML documents are constrained at a syntax level by the XML Recommendation

- `http://www.w3.org/TR/REC-xml`
- well-formedness rules govern the use of markup characters
- a document isn't considered XML unless it is well-formed
- the Document Type Definition (DTD) specification in XML is not used for UBL

XML markup is used to label the document content

- a well-formed document describes a hierarchy of nested elements, attributes and text
  - outer-most element containing all content is called the "document element"
  - all child elements properly nested within their parent
- document annotation constructs can be embedded in content within certain rules
  - comment annotations are for humans to read
  - processing instruction annotations are for programs to read
- the tree's root contains the document element and any annotations before or after the document element as its children
- namespaces are used for richly-defined labels of the information items
  - `http://www.w3.org/TR/xml-names11`
  - provides disambiguation between two items that have the same "local name"
  - label names are a combination of a namespace URI and a simple local name
    - a useful metaphor for a namespace is a dictionary
      - the same word in two languages may represent different meanings as defined in that language's dictionary
      - the same XML local name in two namespaces may represent different meanings as defined in that namespace's semantics
  - e.g. there are two UBL elements named `Location` that are distinguished by their respective namespaces
  - XML-based applications acting on UBL documents must be namespace aware to distinguish the use of constructs

W3C Schema (XSD) Recommendation is used to express constraints on XML labels

- `http://www.w3.org/XML/Schema`
- an XSD file can specify the labels for only a single namespace
- XSD files include other XSD files to get label specifications for the same namespace
- XSD files import other XSD files to get label specifications for other namespaces

Copyright © Crane Softwrights Ltd.

UBL documents

A UBL document is an XML document expressing the final content of information transformed from an application's data model

- all calculations have already been performed according to some calculation model
  - e.g. all tax determinations and calculations are complete
  - e.g. all line item extensions, subtotals and totals express the final values

The committee has specified additional document constraints (Section 6.)

- validation
  - a UBL document ***must*** validate against the corresponding set of document constraints expressed in the W3C Schema expression published by the UBL committee
  - this confirms the use of the labels for information items in the document matches the constraints in the document model
- character encoding
  - `<?xml version="1.0" encoding="UTF-8"?>`
  - a UBL document ***must*** include a declaration of the character encoding in the XML declaration
    - this overrides the XML Recommendation that indicates this is optional when the character encoding is UTF-8
  - the character encoding ***must*** be UTF-8
    - this overrides the XML Recommendation that indicates any character encoding may be specified in the XML declaration
  - this constraint cannot be tested using an XML processor as using any properly-declared character encoding is syntactically correct
- empty and absent elements constrained
  - a UBL element can never be empty or indicated as having a nil value
    - the only exception is the `ExtensionContent` element due to processing restrictions that might remove content from unrecognized namespaces
  - the absence of a UBL element cannot convey meaning to an application
  - this constraint cannot be tested using an XML processor as empty elements are syntactically correct
  - e.g. `CustomerParty`
    - has only optional elements but the rule that an element cannot be empty means that `CustomerParty` can only be used in an instance if at least one of its children are being used in the instance

Copyright © Crane Softwrights Ltd.

UBL documents (cont.)

Not included in the specification but important for interoperability

- no embedded validation schema location of any kind
  - e.g. W3C Schema offers the feature of embedding a linked association between a namespace URI string and a system URI location
    - supposed to be implemented by validation tools as a hint only and not a directive
  - an example (that should not be followed) of embedding a W3C Schema schema association in a UBL Invoice instance:

```
<Invoice
xmlns:cbc="urn:oasis:names:...:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:...:CommonAggregateComponents-2"
xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
urn:oasis:names:specification:ubl:schema:xsd:Invoice-2
u:/cd/artefacts/os-UBL-2.0/xsd/maindoc/UBL-Invoice-2.0.xsd
                    ">
 <cbc:UBLVersionID>2.0</cbc:UBLVersionID>
      ...
```

  - for the trading partner creating the document, this might work fine and validate without error
  - for a trading partner receiving the document, this typically would not work because the system identifiers on the two systems are different
  - some W3C Schema tools interpret this feature as an explicit directive that cannot be turned off
    - the instance will fail validation when the system identifier cannot be resolved

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL documents (cont.)

A community of users must agree on a number of auxiliary specifications for the documents, including (but not limited to):

- a calculation model to use for the arithmetic performed on the information items
    - e.g. Danish OIOUBL project documents their calculation model
        - `http://www.oioubl.info/documents/en/en/Guidelines/OIOUBL_GUIDE_TOTALS.pdf`
- a suite of code lists and value lists and the contexts in which those lists are used
- a set of business rules governing the content
- altogether these form a customization specifications
    - this is described in more detail in Chapter 9 - UBL customization (page 159)

Information semantic integrity and validation is not defined by the UBL specification

- the validity of the content of the document is the responsibility of applications
    - e.g. whether columns add up or taxes are calculated properly
- XML standards only govern the validity of the expression of the content in syntax
- validation unburdens an application from having to check for syntactic accuracy
    - having validated a document, an application can focus on the meaning of the values, not on the lexical constraints of the text used for values
        - an exception would be for patterns for identifiers or other values not checked by validation
    - all applications using the same validation are ensure of consistent checking of those constraints covered by the validation tasks
- an application may perform semantic validation based on business issues
    - e.g. an ordered item is out of stock or has the wrong identification number

Copyright © Crane Softwrights Ltd.

UBL documents (cont.)

The domain and scenario for a UBL document are encoded in the instance itself

- all document types have element information items reserved for identifying the domain
- `cbc:UBLVersionID` - version of UBL document models
  - unspecified value indicates version 2.0
  - a specified value indicates that the instance is posited to validate against the published document models for the given version number
    - e.g. the value "`2.3`" indicates that the instance is intended to be validated with any document model defined by version 2.3 or above
    - the actual items found within are defined by UBL 2.0, 2.1 or 2.2
  - to promote interoperability, this should (but is not required to) reflect the "high-water mark" of the earliest version of UBL needed to support all of the information items in the instance
    - the value may be any version later but cannot be any version earlier
- `cbc:CustomizationID` - domain of definition and use
  - identifies the community of users or the defining collection of profiles and document types
- `cbc:ProfileID` - scenario of definition and use
  - identifies the collection of document types within a customization
  - identifies and informs the business rules and business processes associated with the document
    - a given document type may be used differently and/or configured differently in different profiles
- document element (e.g. `in:Invoice`, `po:Order`, etc) - document type
  - identifies which document within a profile

Copyright © Crane Softwrights Ltd.

Sample UBL instances

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xml/` directory

Excerpt from `xml/UBL-Invoice-2.0-Example.xml`, focusing on line 115:

```
<Invoice xmlns="urn:oasis:...:ubl:schema:xsd:Invoice-2"
  xmlns:cac="urn:oasis:...:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:cbc="urn:oasis:...:ubl:schema:xsd:CommonBasicComponents-2">
  ...
  <cac:PayeeFinancialAccount>
    <cbc:ID>12345678</cbc:ID>
    <cbc:Name>Farthing Purchasing Consortia</cbc:Name>
    <cbc:AccountTypeCode>Current</cbc:AccountTypeCode>
    <cbc:CurrencyCode>GBP</cbc:CurrencyCode>
    <cac:FinancialInstitutionBranch>
      <cbc:ID>10-26-58</cbc:ID>
      <cbc:Name>Open Bank Ltd, Bridgstow Branch </cbc:Name>
      <cac:FinancialInstitution>
        <cbc:ID>10-26-58</cbc:ID>
        <cbc:Name>Open Bank Ltd</cbc:Name>
        <cac:Address>
          <cbc:StreetName>City Road</cbc:StreetName>
          <cbc:BuildingName>Banking House</cbc:BuildingName>
          <cbc:BuildingNumber>12</cbc:BuildingNumber>
          <cbc:CityName>London</cbc:CityName>
          <cbc:PostalZone>AQ1 6TH</cbc:PostalZone>
          <cbc:CountrySubentity>London</cbc:CountrySubentity>
          <cac:AddressLine>
            <cbc:Line>5th Floor</cbc:Line>
          </cac:AddressLine>
          <cac:Country>
            <cbc:IdentificationCode>GB</cbc:IdentificationCode>
          </cac:Country>
        </cac:Address>
  ...
</Invoice>
```

Looking at the `mod/common/UBL-CommonLibrary-2.0` spreadsheet

- look on row 719 to find `PayeeFinancialAccount` ASBIE, and note in column M the associated object class is "`Financial Account`"
- look on row 394 to find "`FinancialAccount`" ABIE

Looking at Crane's model report

- click on "`PA`" index entry, then scroll to "`PayeeFinancialAccount`" and click on row "`719`" to bring up the ASBIE
- then click on the ASBIE name in order to bring up row 394 to find the "`FinancialAccount`" ABIE

Copyright © Crane Softwrights Ltd.

XML Namespaces

Namespaces are used to create globally-unique labels for the information items

- without namespaces then element and attribute names may be ambiguous between different vocabularies
- a namespace prefix is a proxy to a namespace URI string
    - the prefix may be absent for elements
- the processed label of an item is the combination of the namespace URI string and the item's name
    - the prefix used is immaterial
    - no obligation to use the same prefixes used in UBL artefacts

A UBL name is ***not*** unique without considering the namespace

- e.g. `cac:Location` and `cbc:Location` are two different information items
    - the two elements have different semantics (meanings)

Documentary convention for namespace prefixes in UBL documentation

- there is no obligation to use these prefixes in any given UBL document
    - programs ***must not*** assume that these documentary prefixes will be used
- e.g. `harry:Location` and `sally:Location` are the same information item if the two proxies point to the same URI strings
- programs ***must not*** rely on the prefix used and ***must*** rely solely on the fully qualified name as indicated by dereferencing the prefix to its associated URI string

Library namespace prefixes:

```
01 xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
02 xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
03 xmlns:ext="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2"
04 xmlns:sdt="urn:oasis:names:specification:ubl:schema:xsd:SpecializedDatatypes-2"
05 xmlns:udt="urn:un:unece:uncefact:data:specification:UnqualifiedDataTypesSchemaModule:2"
```


Copyright © Crane Softwrights Ltd.

XML Namespaces (cont.)

Chapter 5 - Documents and document models
Section 2 - XML documents and document models

Document type namespace prefixes:

```
01
xmlns:ar="urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2"
02 xmlns:ad="urn:oasis:names:specification:ubl:schema:xsd:AttachedDocument-2"
03 xmlns:bl="urn:oasis:names:specification:ubl:schema:xsd:BillOfLading-2"
04 xmlns:ct="urn:oasis:names:specification:ubl:schema:xsd:Catalogue-2"
05
xmlns:cd="urn:oasis:names:specification:ubl:schema:xsd:CatalogueDeletion-2"
06
xmlns:cu="urn:oasis:names:specification:ubl:schema:xsd:CatalogueItemSpecificationUpdate-2"
07
xmlns:cp="urn:oasis:names:specification:ubl:schema:xsd:CataloguePricingUpdate-2"
08 xmlns:cr="urn:oasis:names:specification:ubl:schema:xsd:CatalogueRequest-2"
09
xmlns:co="urn:oasis:names:specification:ubl:schema:xsd:CertificateOfOrigin-2"
10 xmlns:cn="urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2"
11 xmlns:dn="urn:oasis:names:specification:ubl:schema:xsd:DebitNote-2"
12 xmlns:da="urn:oasis:names:specification:ubl:schema:xsd:DespatchAdvice-2"
13
xmlns:fw="urn:oasis:names:specification:ubl:schema:xsd:ForwardingInstruction-2"
14 xmlns:fr="urn:oasis:names:specification:ubl:schema:xsd:FreightInvoice-2"
15 xmlns:in="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2"
```

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

XML Namespaces (cont.)

Chapter 5 - Documents and document models
Section 2 - XML documents and document models

Document type namespace prefixes (cont.):

```
01 xmlns:po="urn:oasis:names:specification:ubl:schema:xsd:Order-2"
02
xmlns:ox="urn:oasis:names:specification:ubl:schema:xsd:OrderCancellation-2"
03 xmlns:oc="urn:oasis:names:specification:ubl:schema:xsd:OrderChange-2"
04 xmlns:or="urn:oasis:names:specification:ubl:schema:xsd:OrderResponse-2"
05
xmlns:os="urn:oasis:names:specification:ubl:schema:xsd:OrderResponseSimple-2"
06 xmlns:pl="urn:oasis:names:specification:ubl:schema:xsd:PackingList-2"
07 xmlns:qn="urn:oasis:names:specification:ubl:schema:xsd:Quotation-2"
08 xmlns:ra="urn:oasis:names:specification:ubl:schema:xsd:ReceiptAdvice-2"
09 xmlns:rd="urn:oasis:names:specification:ubl:schema:xsd:Reminder-2"
10 xmlns:rt="urn:oasis:names:specification:ubl:schema:xsd:RemittanceAdvice-2"
11
xmlns:rq="urn:oasis:names:specification:ubl:schema:xsd:RequestForQuotation-2"
12
xmlns:sc="urn:oasis:names:specification:ubl:schema:xsd:SelfBilledCreditNote-2"
13
xmlns:si="urn:oasis:names:specification:ubl:schema:xsd:SelfBilledInvoice-2"
14 xmlns:st="urn:oasis:names:specification:ubl:schema:xsd:Statement-2"
15
xmlns:ts="urn:oasis:names:specification:ubl:schema:xsd:TransportationStatus-2"
16 xmlns:wb="urn:oasis:names:specification:ubl:schema:xsd:Waybill-2"
```

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

XML Namespaces (cont.)

Common in UBL instances that the default namespace is used

- the absence of a prefix indicates the use of the default namespace
- attributes without prefixes are in no namespace, not in the default namespace

An excerpt from an invoice instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice xmlns="urn:oasis:...:xsd:Invoice-2"
         xmlns:cac="urn:oasis:...:xsd:CommonAggregateComponents-2"
         xmlns:cbc="urn:oasis:...:xsd:CommonBasicComponents-2">
  <cbc:ID>A00095678</cbc:ID>
  <cbc:CopyIndicator>false</cbc:CopyIndicator>
  <cbc:UUID>849FBBCE-E081-40B4-906C-94C5FF9D1AC3</cbc:UUID>
  <cbc:IssueDate>2005-06-21</cbc:IssueDate>
  <cbc:InvoiceTypeCode>SalesInvoice</cbc:InvoiceTypeCode>
  <cbc:Note>sample</cbc:Note>
  <cbc:TaxPointDate>2005-06-21</cbc:TaxPointDate>
  <cbc:DocumentCurrencyCode>GBP</cbc:DocumentCurrencyCode>
  <cac:OrderReference>
    <cbc:ID>AEG012345</cbc:ID>
    <cbc:SalesOrderID>CON0095678</cbc:SalesOrderID>
    <cbc:UUID>6E09886B-DC6E-439F-82D1-7CCAC7F4E3B1</cbc:UUID>
    <cbc:IssueDate>2005-06-20</cbc:IssueDate>
  </cac:OrderReference>
...
  <cac:AllowanceCharge>
    <cbc:ChargeIndicator>false</cbc:ChargeIndicator>
    <cbc:AllowanceChargeReasonCode>17</cbc:AllowanceChargeReasonCode>
    <cbc:MultiplierFactorNumeric>0.10</cbc:MultiplierFactorNumeric>
    <cbc:Amount currencyID="GBP">10.00</cbc:Amount>
  </cac:AllowanceCharge>
...
```

Note that most values are in elements while some are in attributes:

- e.g. an ABIE element
  - `<Invoice>...</Invoice>`
  - this element is using the default namespace (because there is no name prefix)
  - the content of an ABIE is made up entirely of other elements
- e.g. an ABIE element as an ASBIE:
  - `<cac:OrderReference>...</cac:OrderReference>`
- e.g. a BBIE element:
  - `<cbc:DocumentCurrencyCode>GBP</cbc:DocumentCurrency Code>`
  - the content of a BBIE is made up entirely of text
- e.g. an attribute: `currencyID="GBP"`
  - all attributes without prefixes are not in any namespace

Copyright © Crane Softwrights Ltd.

Unconstrained content

UBL extension point defined for customized extended content

- regarded by the UBL models as unconstrained content as from the UBL model's perspective there are no label constraints on the descendants of the `<ext:ExtensionContent>` element
  - XML syntax rules still apply
- a UBL customization takes advantage of the extension point to add constructs to the document model that are not defined by the UBL information model
  - a customized UBL schema would then impose its own constraints on the extension content
- the apex of the extension content ***should*** be in a non-UBL namespace
  - but this cannot be expressed in W3C Schema constraints
- `<ext:ExtensionContent>` is the only UBL element that is allowed to be empty
  - intermediate processing may discard content in unrecognized namespaces
  - the fact that the element is empty does not convey any information or meaning

Serial processing applications see extensions before seeing any standardized content

- the extensibility point is the first child of the document element of all document models
- extension information can be cached until needed when correlating information found in the standardized constructs
- the extensibility element is not present if the instance has no extensions

Any given UBL instance may include multiple extension fragments

- each extension is specified below a `<ext:UBLExtension>` element
- all `<ext:UBLExtension>` elements are specified below a single `<ext:UBLExtensions>` element
- meta data is available to describe the extension content
- the name and namespace of the apex of extension content should also be distinct for each extension definition

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Unconstrained content (cont.)

An extension may have associated meta data with which to identify its source or use

- all extension meta data is optional, but should be used for clarity
- note that extensions may include the use of standard UBL constructs

```
<Invoice xmlns="urn:oasis:...:xsd:Invoice-2"
         xmlns:cbc="urn:oasis:...:xsd:CommonBasicComponents-2"
         xmlns:cac="urn:oasis:...:xsd:CommonAggregateComponents-2"
         xmlns:ext="urn:oasis:...:xsd:CommonExtensionComponents-2"
         xmlns:demo="urn:x-Demo:Demo">
  <ext:UBLExtensions>
    <ext:UBLExtension>
      <cbc:ID>Demo1</cbc:ID>
      <cbc:Name>Demonstration</cbc:Name>
      <ext:ExtensionAgencyID>CSL</ext:ExtensionAgencyID>
      <ext:ExtensionAgencyName>Crane Softwrights Ltd.
</ext:ExtensionAgencyName>
      <ext:ExtensionVersionID>0.1</ext:ExtensionVersionID>
      <ext:ExtensionAgencyURI>http://www.CraneSoftwrights.com/
links/res-dev.htm</ext:ExtensionAgencyURI>
      <ext:ExtensionURI>urn:x-Demo:Demo:0.1</ext:ExtensionURI>
      <ext:ExtensionReasonCode listURI="urn:x-Demo:Demo:ReasonCodes">1
</ext:ExtensionReasonCode>
      <ext:ExtensionReason>Illustration</ext:ExtensionReason>
      <ext:ExtensionContent>
        <demo:Demo>
          <demo:Thing>This is a test</demo:Thing>
          <cbc:ID>DemoTest</cbc:ID>
          <demo:Total currencyID="GBP">100.00</demo:Total>
        </demo:Demo>
      </ext:ExtensionContent>
    </ext:UBLExtension>
  </ext:UBLExtensions>

  <cbc:ID>A00095678</cbc:ID>
  <cbc:IssueDate>2005-06-21</cbc:IssueDate>
  <cbc:Note>sample</cbc:Note>
  <cac:AccountingSupplierParty>
    <cac:Party>
      <cac:PartyName>
...
```


Copyright © Crane Softwrights Ltd.

XML document models

XML documents are constrained at a label level by an expression of constraints in an XML document model

- constrains the use of elements and their attributes (the only labels for information)
- different schema languages are used to describe document constraints using different approaches
- a grammar-based expression of constraints
  - XML Document Type Definition (DTD)- `http://www.w3.org/TR/REC-xml`
  - ISO/IEC 19757-2 RELAX-NG - `http://www.relax-ng.org`
    - Regular Language for XML
- an assertion-based expression of constraints
  - ISO/IEC 19757-3 Schematron - `http://www.schematron.com`
- a type-hierarchy-based expression of constraints
  - W3C Schema (chosen by the UBL TC for UBL document models)
    - `http://www.w3.org/XML/Schema`
    - `http://www.w3.org/TR/xmlschema-0/` - primer
    - `http://www.w3.org/TR/xmlschema-1/` - structures
    - `http://www.w3.org/TR/xmlschema-2/` - datatypes
- the dispatch of portions of a document tree to different validation tasks
  - ISO/IEC 19757-4 NVDL - `http://www.nvdl.org`
  - Namespace-based Validation Dispatching Language
- a growing family of validation-related standards
  - ISO/IEC 19757 DSDL - `http://www.dsdl.org`
  - Document Schema Definition Languages

XML instance annotations cannot be constrained

- comments and processing instructions can be present or absent anywhere allowed by the XML Recommendation
  - does not interfere with the constraints of the elements, attributes and text
- can be added for explanatory or processing purposes without disturbing the validity of the document
- comments are annotations for a person to read
  - information about the instance
    - e.g. creation meta data (author, generating software, etc.)
    - e.g. source code control strings
- processing instructions are annotations for a program to read
  - information controlling the processing of the instance

Copyright © Crane Softwrights Ltd.

W3C Schema model fragments

XSD fragments are, themselves, XML files

- elements and attributes in the `http://www.w3.org/2001/XMLSchema` namespace
- fragment annotations are included as XML comments
    - copyright statement, creation information, etc.

XSD model annotations are formalisms expressed in elements

- these annotations are not the same as XML annotations
- XSD models are XML documents so XML annotations may be included in addition to XSD annotations
- XSD semantics include formal documentation constructs with which to make annotations richly-defined machine-processed

The reference to `AdditionalStreetName` BBIE in the `Address` ABIE has the following XSD annotations:

```
<xsd:element ref="cbc:AdditionalStreetName"
             minOccurs="0" maxOccurs="1">
 <xsd:annotation>
  <xsd:documentation>
   <ccts:Component>
    <ccts:ComponentType>BBIE</ccts:ComponentType>
    <ccts:DictionaryEntryName>Address. Additional_ Street Name.
                              Name</ccts:DictionaryEntryName>
    <ccts:Version></ccts:Version>
    <ccts:Definition>An additional name of a street used to
                     further specify the Street Name</ccts:Definition>
    <ccts:Cardinality>0..1</ccts:Cardinality>
    <ccts:ObjectClass>Address</ccts:ObjectClass>
    <ccts:PropertyTermQualifier>Additional</ccts:PropertyTermQualifier>
    <ccts:PropertyTerm>Street Name</ccts:PropertyTerm>
    <ccts:RepresentationTerm>Name</ccts:RepresentationTerm>
    <ccts:DataType>Name. Type</ccts:DataType>
    ...
```

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/` directory

- the full schema expressions complete with all documentary annotation constructs
- these are the only normative formal expressions of the UBL 2.0 specification

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xsdrt/` directory

- the compact schema expressions without any documentary XSD annotation constructs
- some schema validation tools run faster without having to process the documentary XSD annotations

Copyright © Crane Softwrights Ltd.

W3C Schema model fragments (cont.)

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/maindoc/` directory:

- one schema fragment for each document type (recall page 88)
  - corresponds to the document model for each document type
  - used as the invocation schema fragment for the validation task
- `UBL-ApplicationResponse-2.xsd`
- `UBL-AttachedDocument-2.xsd`
- `UBL-BillOfLading-2.xsd`
- `UBL-Catalogue-2.xsd`
- `UBL-CatalogueDeletion-2.xsd`
- `UBL-CatalogueItemSpecificationUpdate-2.xsd`
- `UBL-CataloguePricingUpdate-2.xsd`
- `UBL-CatalogueRequest-2.xsd`
- `UBL-CertificateOfOrigin-2.xsd`
- `UBL-CreditNote-2.xsd`
- `UBL-DebitNote-2.xsd`
- `UBL-DespatchAdvice-2.xsd`
- `UBL-ForwardingInstructions-2.xsd`
- `UBL-FreightInvoice-2.xsd`
- `UBL-Invoice-2.xsd`
- `UBL-Order-2.xsd`
- `UBL-OrderCancellation-2.xsd`
- `UBL-OrderChange-2.xsd`
- `UBL-OrderResponse-2.xsd`
- `UBL-OrderResponseSimple-2.xsd`
- `UBL-PackingList-2.xsd`
- `UBL-Quotation-2.xsd`
- `UBL-ReceiptAdvice-2.xsd`
- `UBL-Reminder-2.xsd`
- `UBL-RemittanceAdvice-2.xsd`
- `UBL-RequestForQuotation-2.xsd`
- `UBL-SelfBilledCreditNote-2.xsd`
- `UBL-SelfBilledInvoice-2.xsd`
- `UBL-Statement-2.xsd`
- `UBL-TransportationStatus-2.xsd`
- `UBL-Waybill-2.xsd`

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

W3C Schema model fragments (cont.)

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/common/` directory:

- the arrangement of schema fragments does not correspond to library model files
  - the partitioning is by role of the contained information items, definitions (types) and declarations (elements)
- CCTS definitions used by the UBL schemas:
  - `CodeList_CurrencyCode_ISO_7_04.xsd`
    - fixed set of enumerations for currencies of the world
  - `CodeList_MIMEMediaTypeCode_IANA_7_04.xsd`
    - fixed set of enumerations for MIME types of the Internet
  - `CodeList_UnitCode_UNECE_7_04.xsd`
    - fixed set of enumerations for units of measure for UN/CEFACT
  - `UnqualifiedDataTypeSchemaModule-2.xsd`
    - corresponding to the "`udt`" module in the schema dependency diagram on page 88
      - `urn:un:unece:uncefact:data:specification:UnqualifiedDataTypesSchemaModule:2`
      - imports all the CCTS fragments in use
    - definition of unqualified core component data types
      - included definitions used by UBL
        - `AmountType`, `BinaryObjectType`, `CodeType`, `DateType`, `IdentifierType`, `IndicatorType`, `MeasureType`, `NameType`, `NumericType`, `PercentType`, `QuantityType`, `RateType`, `TextType`, and `TimeType`
      - included definitions not used by UBL
        - `GraphicType`, `PictureType`, `SoundType`, `VideoType`, `DateTimeType`, and `ValueType`
    - defines the structure of core components and supplementary components
      - uses the CCTS code lists for the supplementary components
- CCTS definitions not used by the UBL schemas (but included in the package):
  - `CCTS_CCT_SchemaModule-2.xsd`
    - definition of core component types corresponding to UN/CEFACT concepts
    - provided only for documentation of UN/CEFACT constructs
  - `CodeList_LanguageCode_ISO_7_04.xsd`
    - fixed set of enumerations for languages of the world

Copyright © Crane Softwrights Ltd.

W3C Schema model fragments (cont.)

`http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/common/` directory (cont.):

- UBL definitions:
  - corresponding to the fragments depicted on page 88
    - the cited three-letter prefixes are only documentary and are not mandatory
  - `UBL-CommonAggregateComponents-2.xsd` ("cac")
    - `urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2`
    - declaration of ABIE elements and definition of associated type for each
  - `UBL-CommonBasicComponents-2.xsd` ("cbc")
    - `urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2`
    - declaration of BBIE elements and definition of associated type for each
  - `UBL-CommonExtensionComponents-2.0.xsd` ("ext")
    - `urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2`
    - definition of extension meta data types
  - `UBL-ExtensionContentDatatype-2.0.xsd` ("ext")
    - `urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2`
    - definition of extension content
    - this module is replaced for customization schemas with an equivalent module that imports the extension constructs in the customization namespace
  - `urn:oasis:names:specification:ubl:schema:xsd:QualifiedDatatypes-2`
  - `UBL-QualifiedDatatypes-2.xsd` ("qdt")
    - definition of meta data for non-ATG2-based coded-value data types
- UBL housekeeping:
  - housekeeping declaration of a namespace URI for documenting core component parameters of the information elements
  - `UBL-CoreComponentParameters-2.xsd` ("ccts")
  - `urn:un:unece:uncefact:documentation:2`
  - module contains no information item declarations or type definitions
  - module is not imported by any fragment

Copyright © Crane Softwrights Ltd.

W3C Schema model fragments (cont.)

Recall the schema structures at Formal naming and design rules (page 88)

- note the use of `<xsd:include>` rather than `<xsd:import>`
  - `xsd:include` is obliged to be used when the included fragment defines constructs in the same namespace as the including fragment
  - `xsd:import` is obliged to be used when the imported fragment defines constructs in a different namespace as the importing fragment

Document validation is done only starting with the document schema

- the document is not imported or included by any other schema fragment

Note how the extension namespace construct declarations are split between those that are common and the single construct that defines the extension point

- when defining a customization with extensions in a customization namespace, the content data type fragment is replaced with a fragment importing the customization constructs
- this fragmentation prevents the common meta data extension constructs from being potentially disturbed

Copyright © Crane Softwrights Ltd.

UBL document validation

XML document validation relieves an application only of the responsibility of checking UBL vocabulary constraints

- the documented processing model is only a candidate for consideration and is not mandatory
- an application can be implemented without structure and value validation when other processes are responsible for reporting violations of the document constraints
- a set of applications reflect the same structure and value validation when all are using the same outboard processing model

Structure constraint checking

- confirms the nesting of elements within other elements
    - cardinality of labeled constructs
    - ordering of labeled constructs
- confirms the lexical limitations on the structure of strings of XML text
    - some lexical limitations are non-existent
        - e.g. names and address
            - an arbitrary string
    - some lexical limitations are simple
        - e.g. normalized string values
            - a normalized string has a set of white-space-separated values
    - some lexical limitations are stringent
        - e.g. numeric values
            - a number can only have a sign, digits, and "." for the decimal separator
    - note there are still XML syntax rules governing the escaping of sensitive markup characters in text
        - lexical validation performed on the interpreted marked up XML text
- no constraints on the length of the text string for any value

Value constraint checking

- can only be performed when information items are where they belong
- if the document structure is incorrect then assertions about content cannot be verified
- UBL is shipped with constrained values for coded-value information items for which the technical committee has supplied values
- item length is considered a value constraint, not a lexical constraint
    - the UBL schemas do not impose any constraints on the lengths of values

Semantic (business) interpretation

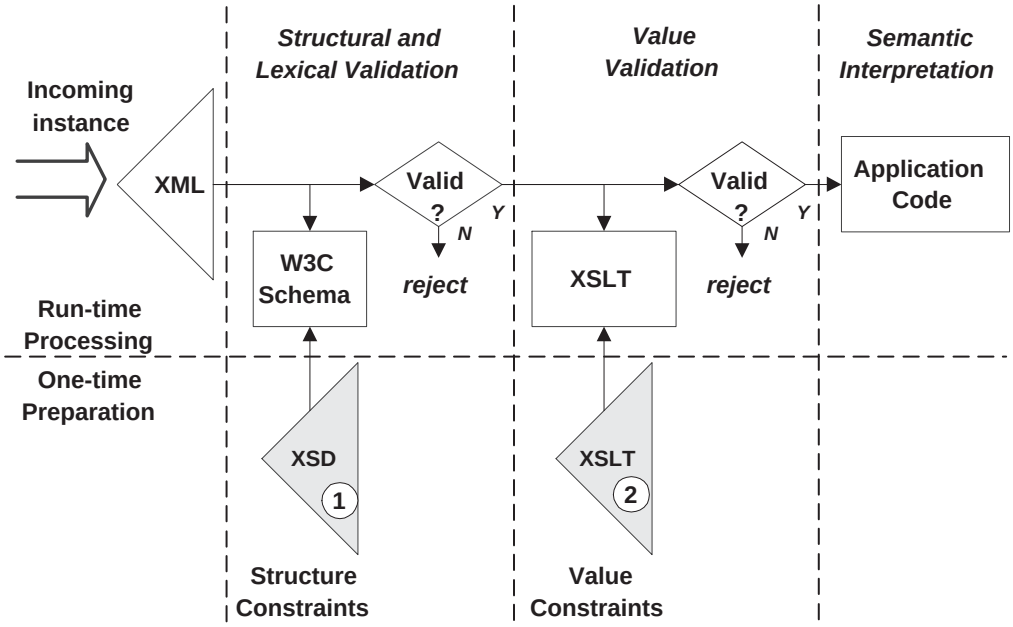
- a UBL application is still responsible for checking semantic content violations
- no amount of document structure and value checking can replace the application's need to implement the application's logic
- e.g. appropriate taxation application
- e.g. checking of inventory against ordered parts

Copyright © Crane Softwrights Ltd.

UBL document validation (cont.)

The UBL 2.0 published document validation process involves only two distinct steps:

- first-pass structure (well-formed, structural and lexical) validation using XSD (1)
  - ensures that all information items are correctly labeled and correctly positioned
  - e.g. for an invoice the document schema is:
  - `http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/maindoc/UBL-Invoice-2.0.xsd`
- second-pass coded-value validation using XSLT (2)
  - ensures the values used for coded-value information items are as expected
  - `http://docs.oasis-open.org/ubl/os-UBL-2.0/val/defaultCodeList.xsl`
  - the XSLT is an implementation of ISO/IEC 19757-3 Schematron
    - the specification of the constraints is described in Chapter 8 - Controlled vocabulary overview (page 147)
    - the specification is translated into Schematron
    - the Schematron is translated into XSLT



Demonstrated using `http://docs.oasis-open.org/ubl/os-UBL-2.0/val/validate.bat` (`.sh`)

- invoked by `test.bat` and `test.sh` in the `val/` directory

Recall UBL information item constraints (page 73)

- an initial set of coded values is defined by the TC for some elements and attributes

Copyright © Crane Softwrights Ltd.

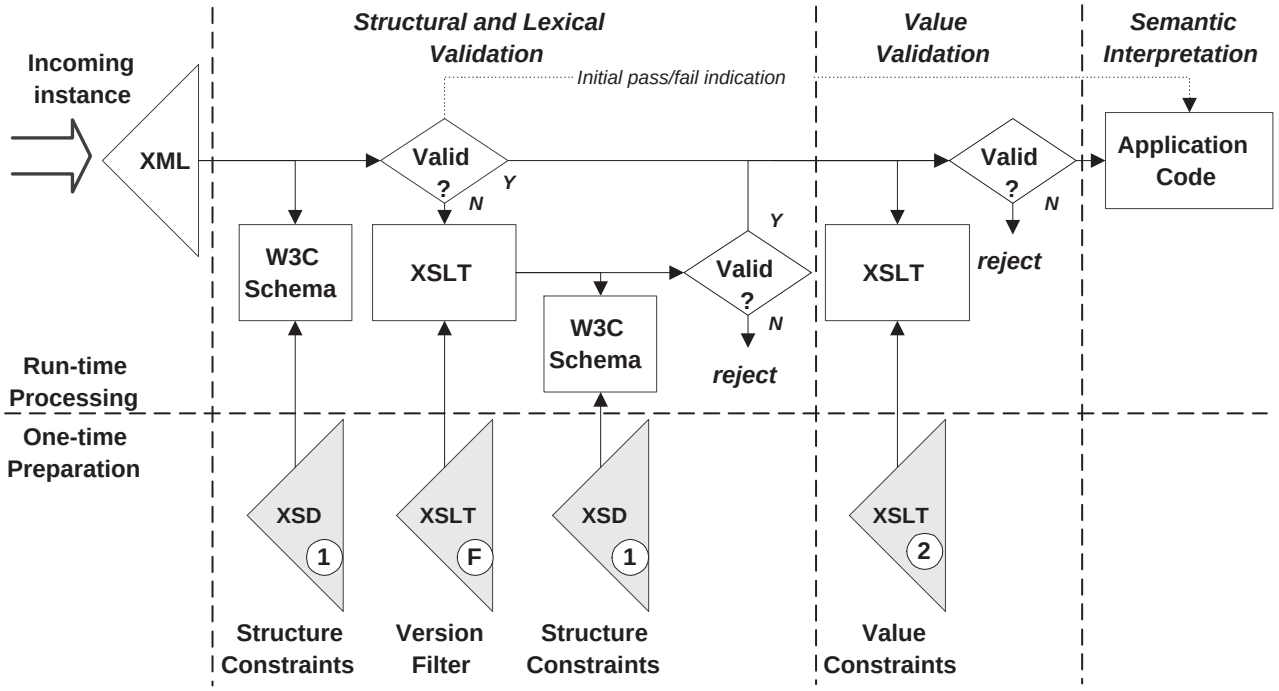
UBL document validation (cont.)

Not all UBL users may be using the same version of the schema

- different communities or different members of any one community may not be in sync
- in a heterogeneous network of users, there will be "older" systems needing to interact with "newer" systems because of differing product cycles

To support forward and subset compatibility, the processing model is modified:

- e.g. supporting UBL 2.x when receiving UBL 2.y instances where y > x
  - i.e. 2.y is published after 2.x
- e.g. supporting a subset of UBL when using complete UBL instances
- to support a given schema, filter (F) removes unexpected items from the instance
  - but only bother filtering if the instance as given has unexpected items
- the same schema is again used to check both unfiltered and filtered content
  - a failure after both means the filtered instance is also unacceptable
  - success proceeds as if the unfiltered instance was acceptable
- the application is informed of initial pass/fail indication
  - can use automated or out-of-band decision making regarding the fact that the filtered instance is being used instead of the original instance



Copyright © Crane Softwrights Ltd.

UBL validation errors

XML well-formedness error

- the document is not XML because it does not pass all the productions in the XML Recommendation
- e.g. missing a right angle bracket from a tag
- checked by an XML processor without the need for a document model

XSD structural or lexical error

- the XML document is well-formed but the use of element and attribute labels violates the constraints expressed in the XSD schemata
- e.g. cardinality
    - e.g. a mandatory element is missing
- e.g. sequence
    - e.g. elements are not in the correct order
- e.g. recognition
    - i.e. an information item being used is not defined by the schema
    - e.g. a typographical error in the spelling of an element or attribute name
- e.g. lexical values
    - i.e. an element's content is not allowed for the element's data type
    - e.g. having alphabetic letters in a numeric data type
- checked by an XSD validating processor according to the constraints of a document model

Controlled vocabulary error

- an error in the information item valued from expected sets of code lists and identifiers
- a test of a value being used violates the set of allowed values for that item
- checked by an XSLT processor using a synthesized stylesheet that checks the values expected for controlled value items

UBL semantic (business) error

- e.g. the item ordered is out of stock
- e.g. the sum of all of the line item amounts does not equal the total amount
- e.g. an incorrect tax rate is being applied based on the tax status of the customer
- checked by an application acting on the UBL instance

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 6 - Model semantics

- Introduction - Model semantics
- Section 1 - Localizations

Outcomes

- be aware of the localization projects for the translation of UBL meta data for information items

Copyright © Crane Softwrights Ltd.

Model semantics

Semantics is about the meaning of things

- by definition
- by association

The information item "UBL name", i.e. element type name, is mnemonic

- the label of the information item in the XML document structure
- the mnemonic is meant to be a useful reminder of the semantic behind the term
- the mnemonic is ***not*** meant to define the term
- NDR rule GNR1 requires information item names to be in Oxford English

The information item "Definition" and "Alternate Business Terms" are descriptive and meant to convey the semantic meaning

- inappropriate for use in other languages because multiple readers of a given language may interpret or translate the definition differently
- would need a language-specific version of these to consistently convey meaning to a group of readers of a language other than English

That the UBL schemas are normative mandates the mnemonics used for element names be unchangeable in all uses of UBL

- the descriptions associated with the mnemonics are informative
- transliteration of the element types does not maintain the "UBL-ness" of the document
- it is inappropriate, and not compatible with UBL, to use translated element and attribute names in instances

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Localizations

Chapter 6 - Model semantics
Section 1 - Localizations

A localization documents descriptive meanings of the UBL mnemonics in different languages

- thus a common understanding of the meaning of information items underpins international interoperability

A UBL localization subcommittee is formed for each locale

- responsible for determining the descriptions and alternate business terms
- responsible for local outreach and promotion
- each committee is named using the two-character locale (language or country code) followed by "LSC" for "Localization Subcommittee"
    - e.g. ESLSC is the Spanish Localization Subcommittee

Localizations available or announced or with a committee formed for development:

- Chinese (Simple)
- Chinese (Traditional)
- Danish
- German
- Italian
- Japanese
- Korean
- Spanish
- Turkish

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Localizations (cont.)

A localization provides a definition and example business terms for each UBL name

- there is no UBL oversight governance on how a localization committee determines the definition of a localization
    - a localization subcommittee may have its own internal processes, meeting schedules and methodologies for establishing the values used in a localization
- the publishing of a localization is a UBL process triggered by a localization committee fulfilling the UBL committee process requirements
    - there is no measurement of accuracy performed by the UBL committee
- the acceptance of a localization by users is the ultimate determination of accuracy and suitability

A localization is not a customization

- a localization cannot change any of the UBL-defined properties of information items
- a customization can compatibly change the use of information items in a UBL interchange
    - this is described in more detail in Chapter 9 - UBL customization (page 159)

A localization facilitates but does not specify semantic interoperability for a given culture

- a localization is not a machine-processed expression of formal semantics
    - but the format of the localization content could be one that can be machine-processed by applications if so desired
        - could be used in the adaptation of user interfaces in different languages
- a localization only provides a common definition for understanding the meaning behind a construct defined by UBL

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Localizations (cont.)

Chapter 6 - Model semantics
Section 1 - Localizations

The UBL name is the normative information item name used in the XML interchange

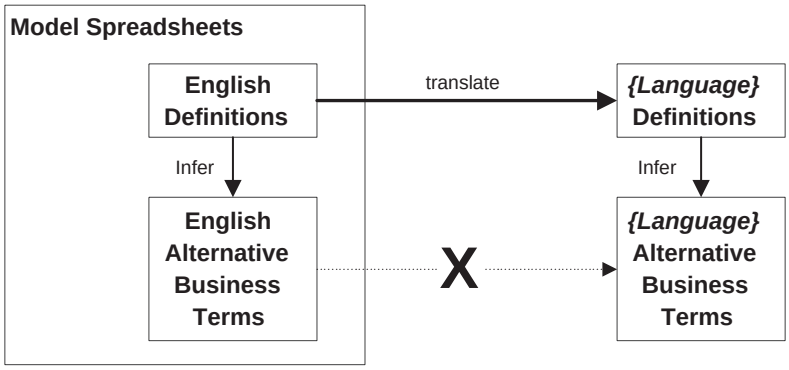
- the name of the element or the name of the attribute
- this forms the key of the information items defined localization definition

The associated UBL definition is the basis of the translation that creates the localized definition

- a localized definition is *not* determined from the UBL name or UBL business term

The localized definition is the basis of the determination of the associated localized business terms

- a localized business term is *not* determined from the UBL name or UBL business term



Copyright © Crane Softwrights Ltd.

Localization spreadsheets

Chapter 6 - Model semantics
Section 1 - Localizations

The UBL International Data Dictionary (IDD) documents the translations of the descriptions and business terms in a spreadsheet:

- `http://lists.oasis-open.org/archives/members/200811/msg00001.html`
  - public review draft for Spanish, Italian, Japanese
- copy of English definition for reference in Column D (scrolled left in screen shot)
- localization committee supplies two columns G and H for their language
- left-click cell at row 5 column D, use menu item Window/Freeze
- optionally hide width of columns A and B and height of row 1
  - left-click on row/column label, then right-click "Hide"

	C	E	F	G	H
2	**UBL Name**	**Cardi-nality**	**Compo nent Type**	***Spanish***	
3				**Business Terms**	**Definition**
4	Address		ABIE	Dirección	Información acerca de una dirección estructurada
5	ID	0..1	BBIE	ID	Identificador de una dirección especifica correspondiente a un esquema de direcciones
6	AddressTypeCode	0..1	BBIE	Tipo de Dirección	Código que especifica el tipo de dirección, como por ejemplo un domicilio social o una dirección postal personal
7	AddressFormatCode	0..1	BBIE	Código de Formato de Dirección	Código para identificar el formato específico de esta dirección
8	Postbox	0..1	BBIE	Apartado postal	Número de apartado postal
9	Floor	0..1	BBIE	Planta	Texto que identifica la planta de una de un edficio
10	Room	0..1	BBIE	Habitación	Habitación, despacho o apartamento de un edificio
11	StreetName	0..1	BBIE	Calle	Nombre de la calle para esta dirección
12	AdditionalStreetName	0..1	BBIE	Calle adicional	Nombre adicional de la calle (aporta información más detallada acerca del nombre de la calle)
13	BlockName	0..1	BBIE	Manzana	Nombre de una manzana de edificios, expresado como texto. Se refiere a una área rodeada de calles y que habitualmente contiene varios edificios identificados en su conjunto por esta dirección
14	BuildingName	0..1	BBIE	Edificio	Nombre identificativo de un edificio
15	BuildingNumber	0..1	BBIE	Número	Número de un edificio
16	InhouseMail	0..1	BBIE	Dirección interna	Ubicación especifica dentro de un edificio

Localization Template

Copyright © Crane Softwrights Ltd.

Public involvement

UBL.xml.org forum for public involvement in creating a localization

- `http://ubl.xml.org/forums/ubl-international-data-dictionary-idd-contributions`
- xml.org membership agreement covers the intellectual property of contributions to the site

The editorial board chairman sets up Google spreadsheets for candidate languages

- participants are given write permissions to add their own content to the spreadsheets
- any language can be requested
- 33 spreadsheets are created for each language

Not limited to only those languages covered by existing localization committees

- once the critical mass of definitions and terms is contributed by the community, a subcommittee could then be formed
    - all spreadsheets must be complete for a submission to go ahead
    - completion may happen in UBL.xml.org or within the subcommittee
- the new subcommittee takes ownership of the spreadsheet contents
- the new subcommittee reviews the contents for any glaring issues
- the new subcommittee packages up the spreadsheets for submission for public review
- the technical committee reviews the submission and approves the public review
- the public review process finds any further problems to be addressed

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 7 - XPath enumerations

- Introduction - Exhaustive enumeration of information items
- Section 1 - Composition of XPath files

Outcomes

- understand the makeup, structure and use of UBL XPath files

Copyright © Crane Softwrights Ltd.

Exhaustive enumeration of information items

Constraint expressions (as used in the UBL schemas) reveal only the parent/child relationship between information items

- one cannot readily tell the impact of the additional number of possible descendents being added to a given element when choosing to include a particular child for that element

The complete ancestry of possible information items is only available through an analysis of the parent/child constraints at every possible level of the document tree

- the UBL TC has performed the analysis and summarized all of the elements and attributes in all contexts of all document types

The information is made available in four forms of what the committee has termed "XPath files"

- a normative description in XML of all possible ancestral and descendent paths of an XML instance
    - this description can be processed for analytical purposes
- a simple text report of the absolute minimum set mandatory information items required to satisfy the document model
- a simple text report of all possible information items defined by the document model use of elements
- an exhaustive sample XML instance that can be processed by non-validating applications

Copyright © Crane Softwrights Ltd.

Exhaustive enumeration of information items (cont.)

Complete suite of XPath information created from the document models

- could be based on the regular nature of the W3C Schema XSD expressions
- could be based on the model information of parent/child relationships
- drawback of being very large files
    - e.g. there are over 800,000 information items (not including recursion) in the UBL Order model

Can also create XPath files from instances

- useful when working with a limited number of information items instead of an exhaustive enumeration
- drawback of being fragile
    - changing anything in the instance renumbers all reference numbers from that point to the end of the document
- Crane has a number of XPath-related resources available in the "Free resources" section of the web site linked from the right-hand marginalia of:
    - `http://www.CraneSoftwrights.com/links/trn-20090212.htm`

UBL XPath files are used in the creation of a number of related resources

- e.g. stylesheet development

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Cataloguing information items in context

W3C XML Path Language (XPath) 1.0 defines two specifications

- a data model for the information found in an XML document
- a syntax for addressing components of the data model
- `http://www.w3.org/TR/1999/REC-xpath-19991116`

XPath exposes only the information found in an XML document

- the syntax used to mark up the information in the document is not preserved nor represented in the data model

XPath defines only seven kinds of tree nodes in its data model

- root node (the parent of the document element node)
    - the document element is the top-most (when thinking of a tree hierarchy) and the outer-most (when thinking of element wrapping) element in the tree
- element, attribute and text nodes (for information content)
- namespace node (for rich labeling of element and attribute nodes)
- comment and processing-instruction nodes (for document annotations)

XPath data model is different than the Document Object Model (DOM)

- the DOM data model for an XML document includes more syntactic information regarding the non-informational components of the document
    - e.g. use of CDATA sections and entity references are preserved in the DOM data model
- the XPath data model for an XML document is agnostic to any choices of syntax made when expressing the information
    - e.g. all CDATA sections and entity references are resolved and are not preserved in the XPath data model

Copyright © Crane Softwrights Ltd.

Cataloguing information items in context (cont.)

Basic use of XPath syntax is to reference the ancestry of an information item

- absolute XPath address to an element begins at outermost element (document element)
  - `/DocumentElement/ChildElement/GrandchildElement`
- relative XPath address to an element can begin anywhere in the document tree
  - `ParentElement/ChildElement`
- XPath address to all elements that are descendants (including children) of an ancestor
  - `AncestorElement//DescendentElement`
- XPath address to an attribute uses "@" as a distinction
  - `AttachedElement/@attributeName`
- XPath address to an attribute found attached to any element
  - `@attributeName`
- XPath address to all attributes attached to an ancestor or its descendants
  - `AncestorElement//@attributeName`

Using XPath syntax to address items can incorporate many aspects of nuance

- looking in different directions of the document tree from the current node
- identifying nodes of different types along the directions
- qualifying the nodes identified by boolean predicates as filters for rejecting unwanted nodes
- `Axis::NodeTest[boolean-predicate-1][boolean-predicate-2]`

XPath typically used to find information in XML documents

- used by XSLT (the Extensible Stylesheet Language Transformations)
  - `http://www.w3.org/TR/xslt`
  - `http://www.w3.org/TR/xslt20/`
  - publishing-oriented construction of new arrangements of structured information
- used by XQuery (An XML Query Language)
  - `http://www.w3.org/TR/xquery/`
  - data-oriented construction of new arrangements of structured information
- used by XPointer (the XML Pointer Language)
  - `http://www.w3.org/TR/xptr`
  - standalone specification for addressing pieces or portions of XML documents

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Cataloguing information items in context (cont.)

UBL uses the most minimal XPath addressing syntax to document a catalogue of element and attribute information items

- used in a descriptive manner rather than its designed use in an expositive manner
- all location steps of a location path only address the nodes in the path
  - steps do not utilize functions or predicates
  - only the child axis, with `child::` omitted, is used in an element's location step
  - the "`@`" abbreviation is used for the `attribute::` axis
- both absolute and relative XPath addresses are used in various UBL support package files
  - note that the use of white-space around information item names is arbitrary
    - no white-space is used in the XPath files generated for UBL document models
    - white-space is used in this training material to make the lengthy addresses more readable within the confines of the page margins
  - XPath text reports use only absolute XPath addresses
    - `UBL-2.0-20070414-XPath.zip` - XPath files for UBL 2.0 models
    - example absolute addresses start at the root node and indicate the document element and then its descendent elements:
      - ```
        /in:Invoice/cbc:IssueDate
        /in:Invoice/cbc:IssueTime
        /in:Invoice/cbc:InvoiceTypeCode
        /in:Invoice/cac:InvoiceLine/cac:ItemInstance/
                                          cbc:ProductTraceID
        ```
  - XPath context reports use both absolute and relative XPath addresses
    - `UBL-2.0-20081231-clsupport.zip` - code list support files for UBL 2.0
    - example relative addresses indicate an element that could be anywhere in the document and then its descendent elements:
      - ```
        cac:SellersItemIdentification/cac:PhysicalAttribute/
                                            cbc:PositionCode
        cac:StandardItemIdentification/cac:PhysicalAttribute/
                                             cbc:PositionCode
        ```


Copyright © Crane Softwrights Ltd.

Namespaces in XPath

XPath 1.0 has special considerations for namespaces

- the use of prefixes in XPath is independent of the use of prefixes in an instance being addressed
- an instance may use the default namespace by specifying that no prefix for an element name is a proxy to a namespace URI string
    - attributes without prefixes in instances are always in no namespace and not in the default namespace
- an XPath location step addresses an element node or an attribute node either with or without a namespace prefix
    - all XPath element and attribute names written without a prefix are, by definition, names in no namespace
    - the XPath name for an element or attribute in a given namespace must have a namespace prefix
- XPath 2.0 adds to 1.0 the setting of the default namespace for un-prefixed element names
    - `http://www.w3.org/TR/xpath20/`

All UBL elements are in a namespace and all UBL attributes are in no namespace

- all element names must be prefixed in XPath addresses
- all attribute names must not be prefixed in XPath addresses
- XPath 2.0 default namespace is not used or assumed

Copyright © Crane Softwrights Ltd.

XPath file XML vocabulary

The normative XPath file is an XML instance enumerating elements and attributes in context

- useful for programmatic analysis of document models that follow the UBL NDR
  - not necessarily useful for models that contain choice groups or mixed content
- this artefact is normative with respect to other XPath files, not with respect to UBL

`<XPath>`

- document element
- `id=`
  - identifier used for referencing this document in other contexts

`<Element>`

- an element in context
- `name=` and `type=`
  - the element type, the type definition and the base type
- `restricts=` and `extends=`
  - data type derivations
- `uri=` and `prefix=`
  - namespace URI string and documentary prefix
- `minOccurs=` and `maxOccurs=`
  - cardinality
- `any=`
  - indication of unconstrained content
  - mutually exclusive with `text=` and `loop=`
- `text=`
  - indication of text content, not element content
  - mutually exclusive with `any=` and `loop=`
- `loop=`
  - indication, using its own name, of this element already being referenced in this element's ancestry (thus is an infinite loop)
  - mutually exclusive with `any=` and `text=`

`<Attribute>`

- an attribute in context
- `name=` and `type=`
  - the attribute name and base type
- `use=`
  - cardinality

Foreign elements and attributes

- any XPath element may contain attributes or child elements not defined by XPath, provided these constructs are not in the XPath namespace

Copyright © Crane Softwrights Ltd.

XPath text reports

Chapter 7 - XPath enumerations
Section 1 - Composition of XPath files

The XPath text report is a human-legible summary for each information item:

- ordinal information item position as a reference number
    - colloquially referred to as "the bang numbers" because of the use of exclamation marks surrounding them to distinguish them in close proximity
    - ordinal element position
    - ordinal separator "."
        - this is not to be interpreted as a decimal point
    - alphabetical ordinal attribute position within each element
        - without zero fill
    - the ordinal pair "123.10" will follow "123.9" and be distinct from "123.1"
- cardinality
- absolute XPath location path address

```
1   1..1 /in:Invoice/
2   0..1 /in:Invoice/ext:UBLExtensions/
3   1..n /in:Invoice/ext:UBLExtensions/ext:UBLExtension/
4   0..1 /in:Invoice/ext:UBLExtensions/ext:UBLExtension/cbc:ID
...
17   1..1 /in:Invoice/cbc:ID
17.1 0..1 /in:Invoice/cbc:ID/@schemeAgencyID
17.2 0..1 /in:Invoice/cbc:ID/@schemeAgencyName
17.3 0..1 /in:Invoice/cbc:ID/@schemeDataURI
17.4 0..1 /in:Invoice/cbc:ID/@schemeID
17.5 0..1 /in:Invoice/cbc:ID/@schemeName
17.6 0..1 /in:Invoice/cbc:ID/@schemeURI
17.7 0..1 /in:Invoice/cbc:ID/@schemeVersionID
18   0..1 /in:Invoice/cbc:CopyIndicator
...
20   1..1 /in:Invoice/cbc:IssueDate
21   0..1 /in:Invoice/cbc:IssueTime
...
23   0..n /in:Invoice/cbc:Note
23.1 0..1 /in:Invoice/cbc:Note/@languageID
24   0..1 /in:Invoice/cbc:TaxPointDate
25   0..1 /in:Invoice/cbc:DocumentCurrencyCode
25.1 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@languageID
25.2 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listAgencyID
25.3 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listAgencyName
25.4 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listID
25.5 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listName
25.6 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listSchemeURI
25.7 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listURI
25.8 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@listVersionID
25.9 0..1 /in:Invoice/cbc:DocumentCurrencyCode/@name
...
```


Copyright © Crane Softwrights Ltd.

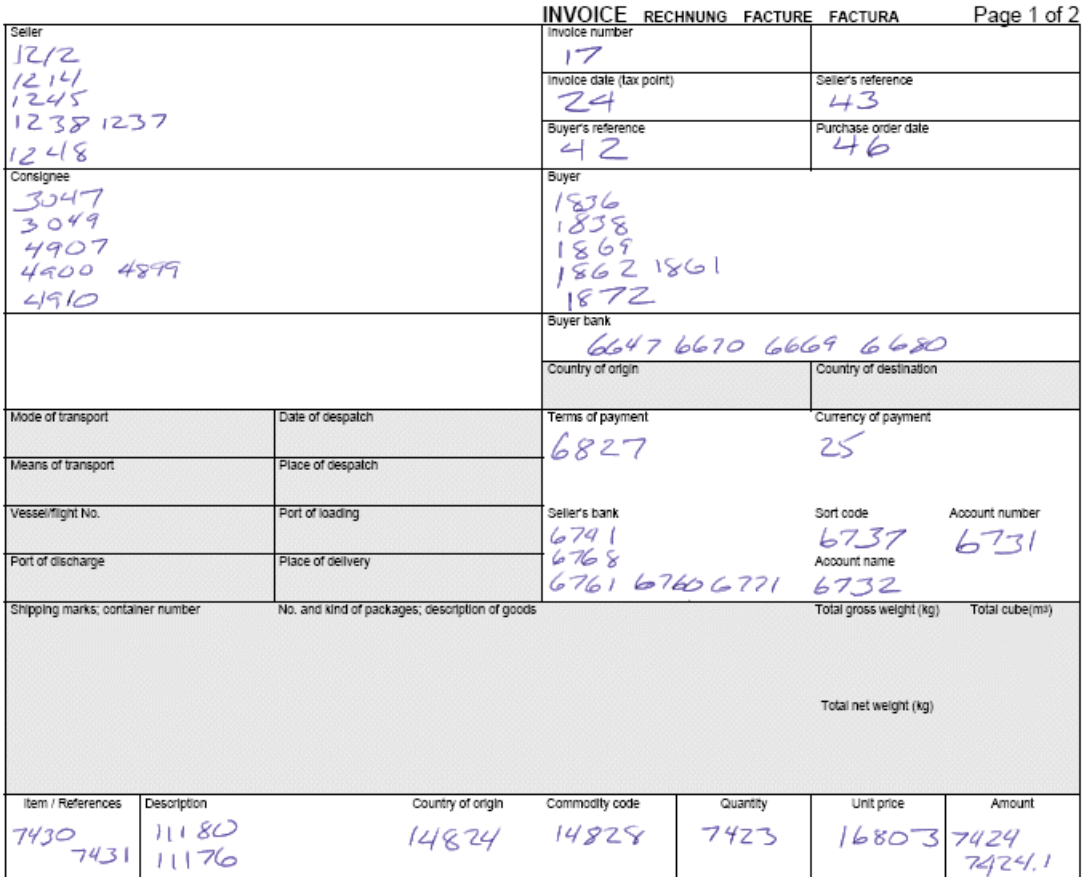
XPath reference numbers

Chapter 7 - XPath enumerations
Section 1 - Composition of XPath files

Using the reference numbers is a concise way to represent long XPath addresses

- quicker to write
- more room for more values
- unambiguously identifies a location in an XML model or document
- very fragile to model or document changes

Sample use of UBL 2.0 XPath Invoice reference numbers in a layout:



Copyright © Crane Softwrights Ltd.

XPath exhaustive instances

Chapter 7 - XPath enumerations
Section 1 - Composition of XPath files

The XPath exhaustive instance is an XML document that instantiates one of every possible information item:

- impossible to validate because reference numbers do not satisfy data types
- quite suitable for non-validating processes such as XSLT

```
<in:Invoice xmlns:in="urn:...:Invoice-2"
  xmlns:cac="urn:...:CommonAggregateComponents-2"
  xmlns:cbc="urn:...:CommonBasicComponents-2"
  xmlns:ext="urn:...:CommonExtensionComponents-2">
   <ext:UBLExtensions>
      <ext:UBLExtension>
         <cbc:ID schemeAgencyID="!4.1!" schemeAgencyName="!4.2!"
schemeDataURI="!4.3!" schemeID="!4.4!" schemeName="!4.5!"
schemeURI="!4.6!" schemeVersionID="!4.7!">!4!</cbc:ID>
          ...
   <cbc:UBLVersionID schemeAgencyID="!14.1!" schemeAgencyName="!14.2!"
schemeDataURI="!14.3!" schemeID="!14.4!" schemeName="!14.5!"
schemeURI="!14.6!" schemeVersionID="!14.7!">!14!</cbc:UBLVersionID>
   <cbc:CustomizationID schemeAgencyID="!15.1!"
schemeAgencyName="!15.2!" schemeDataURI="!15.3!" schemeID="!15.4!"
schemeName="!15.5!" schemeURI="!15.6!" schemeVersionID="!15.7!"
>!15!</cbc:CustomizationID>
   <cbc:ProfileID schemeAgencyID="!16.1!" schemeAgencyName="!16.2!"
schemeDataURI="!16.3!" schemeID="!16.4!" schemeName="!16.5!"
schemeURI="!16.6!" schemeVersionID="!16.7!">!16!</cbc:ProfileID>
   <cbc:ID schemeAgencyID="!17.1!" schemeAgencyName="!17.2!"
schemeDataURI="!17.3!" schemeID="!17.4!" schemeName="!17.5!"
schemeURI="!17.6!" schemeVersionID="!17.7!">!17!</cbc:ID>
   <cbc:CopyIndicator>!18!</cbc:CopyIndicator>
   <cbc:UUID schemeAgencyID="!19.1!" schemeAgencyName="!19.2!"
schemeDataURI="!19.3!" schemeID="!19.4!" schemeName="!19.5!"
schemeURI="!19.6!" schemeVersionID="!19.7!">!19!</cbc:UUID>
   <cbc:IssueDate>!20!</cbc:IssueDate>
   <cbc:IssueTime>!21!</cbc:IssueTime>
   <cbc:InvoiceTypeCode languageID="!22.1!" listAgencyID="!22.2!"
listAgencyName="!22.3!" listID="!22.4!" listName="!22.5!"
listSchemeURI="!22.6!" listURI="!22.7!" listVersionID="!22.8!"
name="!22.9!">!22!</cbc:InvoiceTypeCode>
   <cbc:Note languageID="!23.1!">!23!</cbc:Note>
   <cbc:TaxPointDate>!24!</cbc:TaxPointDate>
   <cbc:DocumentCurrencyCode languageID="!25.1!" listAgencyID="!25.2!"
listAgencyName="!25.3!" listID="!25.4!" listName="!25.5!"
listSchemeURI="!25.6!" listURI="!25.7!" listVersionID="!25.8!"
name="!25.9!">!25!</cbc:DocumentCurrencyCode>   ...
```

(note the indentation of attributes and compression of namespace URI strings is documentary)

Copyright © Crane Softwrights Ltd.

XPath exhaustive instances (cont.)

Using the XPath instance document for rendering will expose the bang numbers

- can use the handwritten version (page 138) for confirmation

Sample rendering of UBL 2.0 XPath instance to reveal the reference numbers:

INVOICE RECHNUNG FACTURE FACTURA Page 1 of 2

Seller !1212! !1214! !1245! !1238! !1237! !1248!		Invoice number !17!	
		Invoice date (tax point) !24!	Seller's reference !43!
		Buyer's reference !42!	Purchase order date !46!
Consignee !3047! !3049! !4907! !4900! !4899! !4910!		Buyer !1836! !1838! !1869! !1862! !1861! !1872!	
		Buyer bank !6647! !6670! !6669! !6680!	
		Country of origin	Country of destination
Mode of transport	Date of despatch	Terms of payment !6827!	Currency of payment !25!
Means of transport	Place of despatch		
Vessel/flight No.	Port of loading	Seller's bank !6741! !6768! !6761! !6760! !6771!	Sort code !6737! Account name !6732!
Port of discharge	Place of delivery		Account number !6731!
Shipping marks; container number	No. and kind of packages; description of goods		Total gross weight (kg) Total net weight (kg)
			Total cube(m³)

Item / References	Description	Country of origin	Commodity code	Quantity	Unit price	Amount
!7430! !7431!	!11180! !11176!	!14824!	!14828!	!7423!	!16803!	!7424! !7424.1!

Copyright © Crane Softwrights Ltd.

XPath file normative content

Chapter 7 - XPath enumerations
Section 1 - Composition of XPath files

The normative XPath XML file contains for each information item:

- name information
- type base
- cardinality
- only normative in regard of XPath files, not in regard of the UBL delivery

```
<XPath xmlns="urn:oasis:names:tc:ubl:schema:XPath-1.0"
       id="urn:oasis:names:tc:ubl:XPath:Invoice-2.0">
  <Namespace prefix="in" uri="urn:...:Invoice-2"/>
  <Namespace prefix="cac" uri="urn:...:CommonAggregateComponents-2"/>
  <Namespace prefix="cbc" uri="urn:...:CommonBasicComponents-2"/>
  <Namespace prefix="ext" uri="urn:...:CommonExtensionComponents-2"/>
  ...
  <Element name="Invoice" type="InvoiceType"
           prefix="in" minOccurs="1" maxOccurs="1">
    <Element name="UBLExtensions" type="UBLExtensionsType"
             prefix="ext" minOccurs="0" maxOccurs="1">
      <Element name="UBLExtension" type="UBLExtensionType"
               prefix="ext" minOccurs="1" maxOccurs="unbounded">
        <Element name="ID" type="IDType"
                 extends="udt:IdentifierType"
                 prefix="cbc" minOccurs="0" maxOccurs="1" text="">
          <Attribute name="schemeAgencyID" use="optional"
                     type="xsd:normalizedString"/>
          ...
        </Element>
        ...
      </Element>
    </Element>
    <Element name="UBLVersionID" type="UBLVersionIDType"
             extends="udt:IdentifierType"
             prefix="cbc" minOccurs="0" maxOccurs="1" text="">
      <Attribute name="schemeAgencyID" use="optional"
                 type="xsd:normalizedString"/>
      ...
    </Element>
    <Element name="CustomizationID" type="CustomizationIDType"
             extends="udt:IdentifierType"
             prefix="cbc" minOccurs="0" maxOccurs="1" text="">
      ...
```

(note the indentation of attributes and compression of namespace URI strings is documentary)

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

XPath file collections

The XPath files are available in the committee repository

- `http://docs.oasis-open.org/ubl/submissions/XPath-files/`
  - the `XPath-files-readme.html` index links to the files and the publicly-archived posts that describe the files
  - a local copy - 300Mb expanding to 8Gb:
    - `Crane/UBL/UBL-2.0-20070414-XPath.zip`

For each document type, a number of XPath files are available:

- `text/full/*-XPath.txt` - full XPath report file
  - every element and attribute information item is enumerated
- `text/minimal/*-Minimal-XPath.txt` - minimal XPath report file
  - only those mandatory element and information items that would make up a minimally-valid XPath instance are listed
  - the reference numbers shown are from the full XPath report file
- `xml/instance/*-Instance.xml` - non-validated exhaustive instance
  - a non-validating instance including one of every element and attribute information item
  - each item's content is the reference number enclosed in exclamation marks
- `xml/XPath/*-XPath.xml` - normative XPath XML expression
  - an XML instance enumerating every element and attribute in every possible context
  - suitable for processing by XML-based analysis tools

The XPath normative XML document model is expressed using ISO/IEC 19757-2 RELAX-NG

- `xpath.rnc`
- an approximation is expressed using W3C Schema
  - `xpath.xsd`
- a sample XSLT stylesheet illustrating the processing of an XPath file
  - `xpath.xsl`
  - note that there are XSLT processors that cannot handle the magnitude of some of the XPath files even with this small stylesheet
    - the limiting factor is the size of the source node tree created from the input XPath XML file

Copyright © Crane Softwrights Ltd.

XPath file collections (cont.)

XPath files are used by the UBL TC for many purposes:

- initially developed only for their first purpose but then many new uses became evident
- stylesheet implementation validation
  - the HISC subcommittee innovated the XPath file concept for two purposes:
    - business experts could specify the placement of UBL information items in a presented layout
    - stylesheet implementations could validate the placement of UBL information items in a presented layout
- code list contexts of use
  - the UBL XPath files were integral in synthesizing the context files and resulting `val/defaultCodeList.xsl` stylesheet used in the UBL 2.0 specification
  - see Chapter 8 - Controlled vocabulary overview (page 147) for overview
- schema subset validation
  - a customization methodology that utilizes XPath files is proposed for validating that the constraints of a customization do not violate the constraints of UBL 2.0
    - this is described in more detail in Chapter 9 - UBL customization (page 159)
- Small Business Subset 1.0 specification
  - version 1.0 of the UBL Small Business Subset was specified using XPath files

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Exhaustive XPath file drawbacks

File composition is a challenge for review and application processing

- line endings are linefeed characters (not carriage return characters)
    - not suitable for Notepad application in Windows environments
- some files are very large
    - e.g. over 850Mb for the `OrderResponse` full text report file
- in Windows environments the WordPad application accommodates both line end issues and file sizes
    - but does so very slowly

Tree-based XML processing paradigms may not be able to accommodate the magnitude of the normative XPath files

- the memory footprint of most tree-based XML processing approaches is a factor of the size of the XML document itself
    - the larger the input XML document, the more memory is required to store the information
- XSLT-based and DOM-based processors may have problems
    - exceeding capacity limits
        - the processor must have the resources to build an in-memory representation of the XML document
    - exceeding acceptability in performance
        - the processor must take the time to build the in-memory representation of the XML document
    - some research is going into processors that have an external non-memory-based representation of the input document that is suitable for processing
    - the memory footprint and processing speed of a SAX-based process is not a factor of the size of the XML document
        - the XML document can be any size and the memory utilization is only a factor of the application's algorithm
        - an application need only maintain whatever information it wants to maintain as the XML document is processed
        - most of the committee XPath files were generated using the freely-available Python programming language and its SAX interface to XML documents
            - `http://www.python.org`

Fragile to model changes

- XPath files are only useful as long as the document model does not change
- any change to the document model will shift some of the numbers being used
- always cite the version of the corresponding XPath file when citing reference numbers

Copyright © Crane Softwrights Ltd.

XPath reports of arbitrary XML instances

A smaller XPath report, based on an arbitrary XML instance, is also very useful

- one might create an XML instance of a UBL document and need to summarize or talk about the information items in the instance
- unlike the XPath file for a document model revealing all possible items of all possible instances, the XPath file for a given instance reveals only the information items found in the instance

Crane has released XSLT stylesheets to create XPath reports of XML instances

- found online at `http://www.CraneSoftwrights.com/links/trn-20090212.htm`
  - go to the "Free resources" link in the right-hand marginalia
  - go to the "UBL, XPath and Code List" page linked from the first section
  - go to the "XPath file resources" section
  - download the "XPath for XML instances" package
- the `readme-xml2xpath.htm` file documents the use of the stylesheets

`Crane-xml2xpath.xsl`

- produces a text XPath report of the information items in the XML instance

`Crane-xml2xpath-html.xsl`

- produces an HTML XPath report of the information items in the XML instance

`Crane-xml2xpath-instance.xsl`

- produces a text XPath instance report of the information items in the XML instance

Copyright © Crane Softwrights Ltd.

Exploiting XPath reference numbers

Model reference numbers are more stable than instance reference numbers

- XPath reference numbers are very stable when obtained from the document model
    - the document model doesn't change very often
- XPath reference numbers are very fragile when obtained from an XML instance
    - the XML instance may have to change if something was forgotten

Stylesheet writing methodology:

- works with XPath files from either the document model or from a representative instance
- steps 1 and 2 can be performed by a non-technical resource familiar with the form
- steps 3 and 4 would be performed by a stylesheet writer
- note that the nature of the UBL instance having only final values not needing any calculations supports this approach
    - the stylesheet is not responsible for performing calculations on the instance data
    - the stylesheet is only laying instance information out on the page and not manipulating values

Step 1 - prototype the layout without using stylesheets

- draw the layout by hand or use some kind of technology to represent the layout content
- when laying out boxes this involves determining box locations, dimensions and labels
- e.g. see the layout sample on page 138
    - only the layout areas need defining first, not the data content

Step 2 - annotate the layout with a set of XPath reference numbers

- get the reference numbers from the XPath file of either the document model or a representative XML instance
- e.g. see the reference numbers on page 137
    - find the path to the desired item in order to find the bang number

Step 3 - write the stylesheet using the XPath reference numbers as a guideline

- the reference number in combination with the XPath file provides an unambiguous specification of the XPath address for each information item in the result
- e.g. see the instance example on page 139
    - each element and attribute has a unique bang number

Step 4 - test the stylesheet using the corresponding XPath instance report

- the stylesheet rendering of the XPath instance report should render the same reference numbers as the annotated layouts produced in step 2
- e.g. see the layout results on page 140
    - may need to parameterize the stylesheet to avoid numeric formatting of non-numeric bang numbers

Copyright © Crane Softwrights Ltd.

Chapter 8 - Controlled vocabulary overview

- Introduction - Controlled vocabularies in business documents
- Section 1 - Controlled vocabulary overview

Outcomes

- overview the concepts of validating UBL documents against a collection of controlled vocabularies such as code lists and identifier lists

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Controlled vocabularies in business documents

Business documents have many information items valued using controlled vocabularies

- an abstract and compact value expressed to represent an agreed-upon semantic
- often mnemonic in a particular language
    - e.g. "`USD`" for the US dollar currency code
    - e.g. "`ES`" for the Spain country code
- sometimes non-mnemonic to be language independent
    - e.g. "`42`" for "Payment to bank account" payment means code

Controlled vocabularies include codes and identifiers

- codes represent abstract concepts
- identifiers distinguish concrete instantiations of concepts
    - e.g. account codes specific to a trading partner

Registration authorities are responsible for publicly-available value lists

- e.g. International Organization for Standardization (ISO)
- e.g. United Nations Economic Commission for Europe (UN/ECE)

Trading partner agreements need a rigorous expression of constraints

- there is an opportunity for misunderstanding if the parties cannot agree a priori on the coded values acceptable to their document exchanges
- value constraints are layered on top of structural and lexical constraints for a business document vocabulary
    - so as not to disturb the structural and lexical constraints for the documents

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Controlled vocabularies in business documents (cont.)

Traditional use of XSD Schema enumerations to specify value lists is too restrictive

- ties the value validation to the structural and lexical validation in a single expression of the document constraints
  - communities of users work with standardized expressions of document constraints
  - when business requirements need to be tailored, the structural expressions are tampered with
  - interoperability is promoted when the document constraint expressions are read-only and unchanged from the published standards
- globally-declared information items have document-wide value constraints
  - business rules for trading partners my require an information item to have different value constraints in different document contexts

Emerging standards for the outboard expression of controlled vocabularies

- OASIS code list representation technical committee
  - `http://www.oasis-open.org/committees/codelist`
- OASIS genericode 1.0
  - `http://docs.oasis-open.org/codelist/genericode`
  - an XML vocabulary for the expression of a list of values
- OASIS context/value association using genericode (draft)
  - `http://www.oasis-open.org/committees/document.php?document_id=29990`
  - an XML vocabulary for the expression of the association of XML document contexts with lists of values
  - useful for validation or user interface implementation or any other purpose
  - independent of the XML vocabulary of the documents being validated
    - works in step with any structural validation technology (e.g. XSD, RELAX-NG, DTD)

Crane's Schematron-based validation using CVA using genericode: CVA2sch

- one way to use CVA using genericode files for validation
  - there are many possible uses of genericode files without obligation to use this approach
- other OASIS committees and companies using XSD are considering adopting this methodology
- migrating to become part of an Apache project for Schematron

Crane's "Practical Code List Implementation" book details the methodology

- see "Book excerpts" at `http://www.CraneSoftwrights.com/links/trn-20090212.htm`
- the methodology applies to any XML vocabulary, not just UBL

Copyright © Crane Softwrights Ltd.

Controlled vocabularies in business documents (cont.)

The UBL package includes a representative default set of controlled vocabularies

- the genericode XML vocabulary is used to create an instance of an enumeration of values
  - meta data identifies the set of values
- a snapshot of controlled vocabularies is included as genericode files
  - `http://docs.oasis-open.org/ubl/os-UBL-2.0/cl/`
  - includes meta data for all UBL lists
  - the UBL 2.0 package uses genericode 0.4
  - the UBL 2.0 update package uses genericode 1.0
- trading partners can agree on their own lists of values to use
  - would include meta data to identify the custom lists

The `defaultCodeList.xsl` stylesheet is an informative (non-normative) implementation of the default set of codes

- `http://docs.oasis-open.org/ubl/os-UBL-2.0/val/`
- recall the validation scenarios page 119 and page 120
- created using an early version of `CVA2sch`

Trading partners can exchange context/value association files and genericode files

- can choose to use the default set of controlled vocabularies "out of the box"
- can choose to select a different set of vocabularies
  - represents an agreement to conform to code lists separate from the agreement to conform to UBL structures
- the files are tailored to the particular business process agreed upon between trading partners
- the files form part of the formal trading partner agreement
- each party can have an independent implementation of the validation that uses these declarative files
  - implementation choices are particular to a trading partner environment
- each party continues to use published, standardized and unmodified structural and lexical expressions
- partners can also agree on various business rules constraining the values of data
  - expressed as assertions that need to be true or false regarding content found in the UBL instances

Copyright © Crane Softwrights Ltd.

UBL use of controlled vocabularies

Chapter 8 - Controlled vocabulary overview
Section 1 - Controlled vocabulary overview

Some elements and attributes in UBL are governed by controlled vocabularies

- the committee provides sample validation for a subset of items
- communities can prescribe their own collections of lists
- trading partners can prescribe their own collections of lists

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice xmlns="urn:oasis:...:xsd:Invoice-2"
         xmlns:cac="urn:oasis:...:xsd:CommonAggregateComponents-2"
         xmlns:cbc="urn:oasis:...:xsd:CommonBasicComponents-2">
  <cbc:ID>A00095678</cbc:ID>
  <cbc:CopyIndicator>false</cbc:CopyIndicator>
  <cbc:UUID>849FBBCE-E081-40B4-906C-94C5FF9D1AC3</cbc:UUID>
  <cbc:IssueDate>2005-06-21</cbc:IssueDate>
  <cbc:InvoiceTypeCode>SalesInvoice</cbc:InvoiceTypeCode>
  <cbc:Note>sample</cbc:Note>
  <cbc:TaxPointDate>2005-06-21</cbc:TaxPointDate>
  <cbc:DocumentCurrencyCode>GBP</cbc:DocumentCurrencyCode>
  <cac:OrderReference>
    <cbc:ID>AEG012345</cbc:ID>
    <cbc:SalesOrderID>CON0095678</cbc:SalesOrderID>
    <cbc:UUID>6E09886B-DC6E-439F-82D1-7CCAC7F4E3B1</cbc:UUID>
    <cbc:IssueDate>2005-06-20</cbc:IssueDate>
  </cac:OrderReference>
...
  <cac:AllowanceCharge>
    <cbc:ChargeIndicator>false</cbc:ChargeIndicator>
    <cbc:AllowanceChargeReasonCode>17</cbc:AllowanceChargeReasonCode>
    <cbc:MultiplierFactorNumeric>0.10</cbc:MultiplierFactorNumeric>
    <cbc:Amount currencyID="GBP">10.00</cbc:Amount>
  </cac:AllowanceCharge>
...
```

Example coded items in the above sample:

- `<cbc:InvoiceTypeCode>`
  - sample values not provided by the UBL committee
- `<cbc:DocumentCurrencyCode>` and `currencyID=`
  - sample values provided by the UBL committee constrained by UN/CEFACT schema limitations

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL use of controlled vocabularies (cont.)

Specifying a coded value in a UBL document can include instance-level meta data

- distinguishes a value as being from that list with matching list-level meta data

The instance-level meta data varies slightly for UN/CEFACT-defined unqualified data types

- inherited by those UBL information items derived from CCTS unqualified types
- the core component value is the value of the element
- the supplementary component value is an attribute of the element
- the meta data values are in other attributes of the same element
- for `currencyID=` of amounts
  - `currencyCodeListVersionID=`
    - mapped in CVA2sch to genericode `<Version>`
- for `unitCode=` of the `<cbc:MeasureType>` element
  - `unitCodeListVersionID=`
    - mapped in CVA2sch to genericode `<Version>`
- for `unitCode=` of the `<cbc:QuantityType>` element
  - `unitCodeListID=`
    - mapped in CVA2sch to genericode `<Version>`
  - `unitCodeListAgencyID=`
    - mapped in CVA2sch to genericode `<Agency><Identifier>`
  - `unitCodeListAgencyName=`
    - mapped in CVA2sch to genericode `<Agency><LongName>`

Consider how currency values are entered in a UBL instance

- e.g. `<cbc:Amount currencyID="RON">10.00</cbc:Amount>`
  - specifies an amount of 10 Romanian new leu
  - no instance level meta data is included in the element specification
  - the recipient must make an assumption about which code list the value comes from
- e.g. `<cbc:Amount currencyID="RON" currencyCodeListVersionID="1951">10.00</cbc:Amount>`
  - first Romanian leu in 1867
  - Romanian new leu "RON" in 1947
  - in 1952 "RON" was replaced with "ROL" for Romanian leu
  - in 2005 "ROL" was replaced with "RON" for Romanian new leu
  - 1 RON(2005) is worth over 200,000 RON(1951)

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL use of controlled vocabularies (cont.)

UBL-defined code list element meta data attributes for elements <cbc:*????*Code>:

- listID=
  - mapped in CVA2sch to genericode <LongName @Identifier='listID'> or just the first <LongName>
- listAgencyID=
  - mapped in CVA2sch to genericode <Agency><Identifier>
- listAgencyName=
  - mapped in CVA2sch to genericode <Agency><LongName>
- listName=
  - mapped in CVA2sch to genericode first <LongName>
- listVersionID=
  - mapped in CVA2sch to genericode <Version>
- listURI=
  - mapped in CVA2sch to genericode <LocationUri>
- listSchemeURI=
  - mapped in CVA2sch to genericode <CanonicalVersionUri>

UBL-defined identifier element meta data attributes for elements <cbc:*????*ID>:

- schemeAgencyID=
  - mapped in CVA2sch to genericode <Agency><Identifier>
- schemeAgencyName=
  - mapped in CVA2sch to genericode <Agency><LongName>
- schemeName=
  - mapped in CVA2sch to genericode first <LongName>
- schemeVersionID=
  - mapped in CVA2sch to genericode <Version>
- schemeDataURI=
  - mapped in CVA2sch to genericode <LocationUri>
- schemeURI=
  - mapped in CVA2sch to genericode <CanonicalVersionUri>

Each code list has its own definition of list-level meta data values

- different versioning schemes
- different wording of list titles and names
- CVA2sch assumes the list-level meta data structures are genericode

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

UBL validation of controlled vocabularies

The supplied `defaultCodeList.xsl` provides code list conformance validation supporting the following:

- with the corresponding genericode file name in the UBL 2.0 delivery
- UN/ECE Recommendation 19 Transport Mode Code
    - `cl/gc/default/TransportModeCode-2.0.gc`
    - incorrectly labeled inside as "Recommendation 16"
- UN/ECE Recommendation 20 Unit of Measure Codes
    - `cl/gc/cefact/UnitOfMeasureCode-2.0.gc`
- UN/ECE Recommendation 21 Packaging Type Code
    - `cl/gc/default/PackagingTypeCode-2.0.gc`
- UN/ECE Recommendation 24 Transportation Status Codes
    - `cl/gc/default/TransportationStatusCode-2.0.gc`
- UN/ECE 3155 Communication Address Code Qualifier
    - `cl/gc/default/ChannelCode-2.0.gc`
- UN/ECE 4461 Payment Means
    - `cl/gc/default/PaymentMeansCode-2.0.gc`
- UN/ECE 4465 Adjustment Reason Description
    - `cl/gc/default/AllowanceChargeReasonCode-2.0.gc`
- UN/ECE 8053 Equipment Type Code Qualifier
    - `cl/gc/default/TransportEquipmentTypeCode-2.0.gc`
- IANA_7_04 Binary Object MIME Code
    - `cl/gc/cefact/BinaryObjectMimeCode-2.0.gc`
- ISO 3166-1 Country Codes
    - `cl/gc/default/CountryIdentificationCode-2.0.gc`
- ISO 4217 Alpha Currency Codes
    - `cl/gc/cefact/CurrencyCode-2.0.gc`
- UBL chip codes
    - `cl/gc/default/ChipCode-2.0.gc`
- UBL document status codes
    - `cl/gc/default/DocumentStatusCode-2.0.gc`
- UBL latitude direction codes
    - `cl/gc/default/LatitudeDirectionCode-2.0.gc`
- UBL line status codes
    - `cl/gc/default/LineStatusCode-2.0.gc`
- UBL longitude direction codes
    - `cl/gc/default/LongitudeDirectionCode-2.0.gc`
- UBL operator codes
    - `cl/gc/default/OperatorCode-2.0.gc`
- UBL substitution codes
    - `cl/gc/default/SubstitutionStatusCode-2.0.gc`

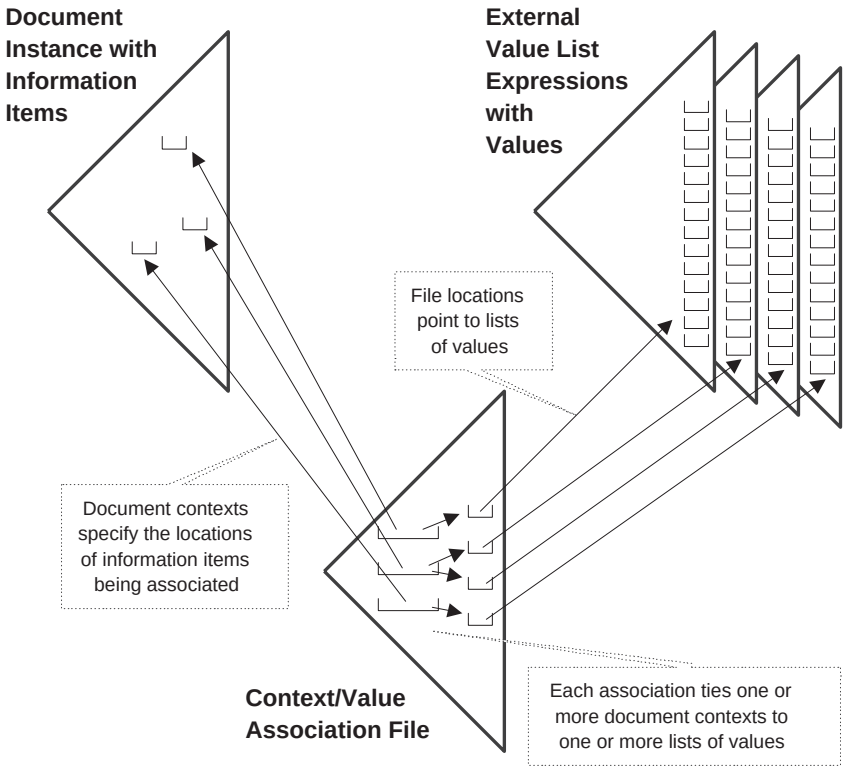
Copyright © Crane Softwrights Ltd.

Specifying code list conformance

Chapter 8 - Controlled vocabulary overview
Section 1 - Controlled vocabulary overview

Specifying, extending and restricting controlled vocabularies in XML documents

- context/value association declaratively ties a document context to sets of values
  - i.e. "those enumerated values over there are to be used to validate the specified values for this particular document context"
- an XML vocabulary is used to create an instance of a context/value association file
  - W3C XPath is used to specify document context
  - a URI is used to point to an external expression of enumerated values
  - an association ties a document context to the sets of values for that context

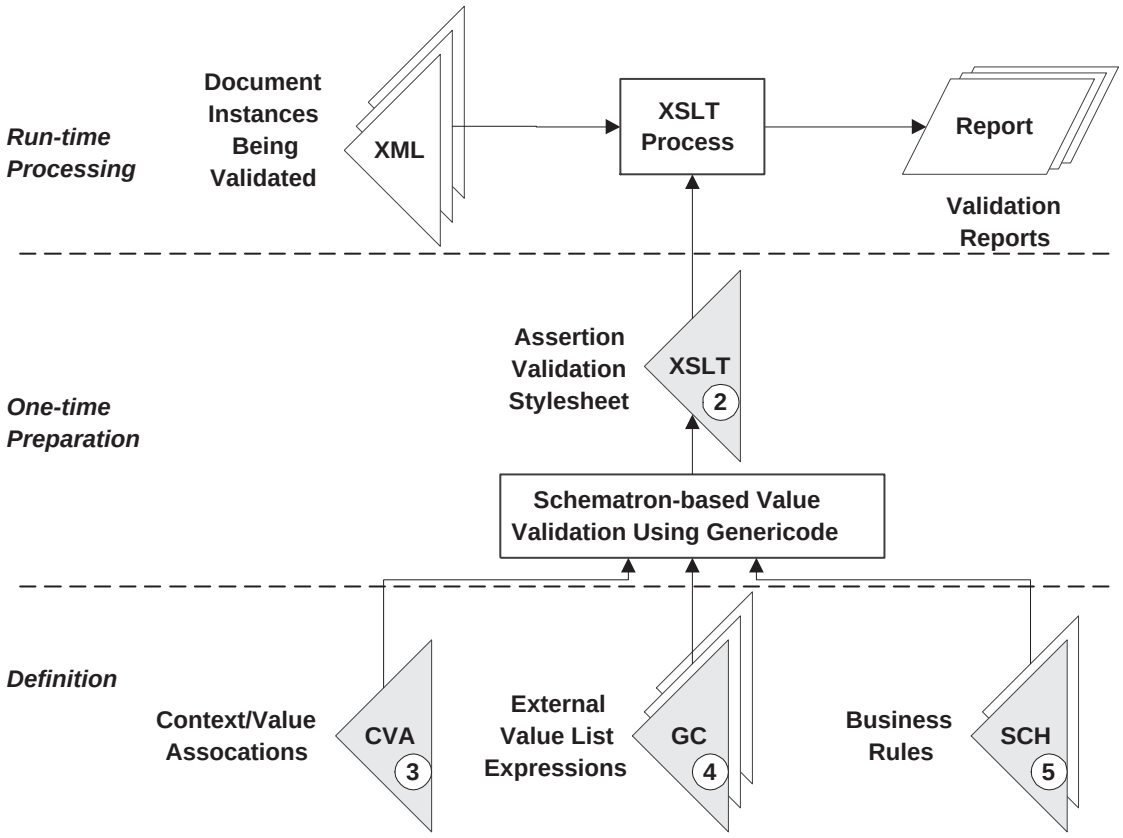


Copyright © Crane Softwrights Ltd.

Specifying code list conformance (cont.)

Validating the use of a controlled vocabulary

- recall the validation scenarios page 119 and page 120
  - first-pass performs structural and lexical validation on the input instance
  - second-pass value validation implementation:
  - only when the first pass is successful does it make sense to do a second pass to perform value validation on the input instance
    - structural validation ensures the information items are correctly found
    - lexical validation ensures the information items are correctly formed
- the UBL methodology prepares the second-pass validation artefact based on ISO/IEC 19757-3 Schematron
  - this diagram shows the use of XSLT for the implementation of Schematron
  - other implementations of Schematron are available (e.g. Python)

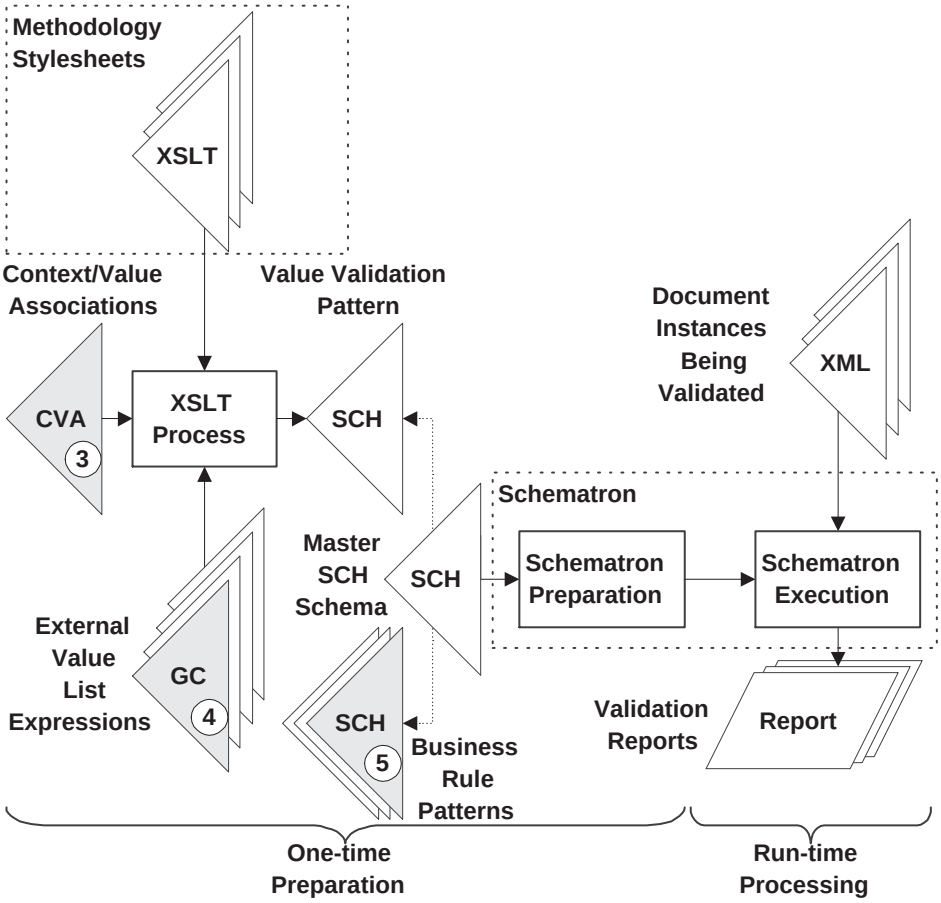


Copyright © Crane Softwrights Ltd.

Context/value validation implementation

Two separate implementations of running code are used to effect the result:

- the "Methodology Stylesheets" box represents the CVA XSLT stylesheets supplied with this methodology to transform context/value association files into a Schematron pattern
- the "Schematron" box represents the particular implementation of Schematron being deployed by a user of this methodology
  - the methodology package supplies an XSLT implementation of Schematron that exits with a non-zero return code when it reports violations to the standard error port in a simple text format
  - alternative implementations of Schematron are publicly-available from `http://www.Schematron.com` including versions that report violations in a rich XML vocabulary for subsequent downstream processing



Recall Specifying code list conformance (page 156)

Recall UBL document validation (page 119)

Copyright © Crane Softwrights Ltd.

Context/value validation implementation (cont.)

Context/value associations and external code list expressions are converted to Schematron

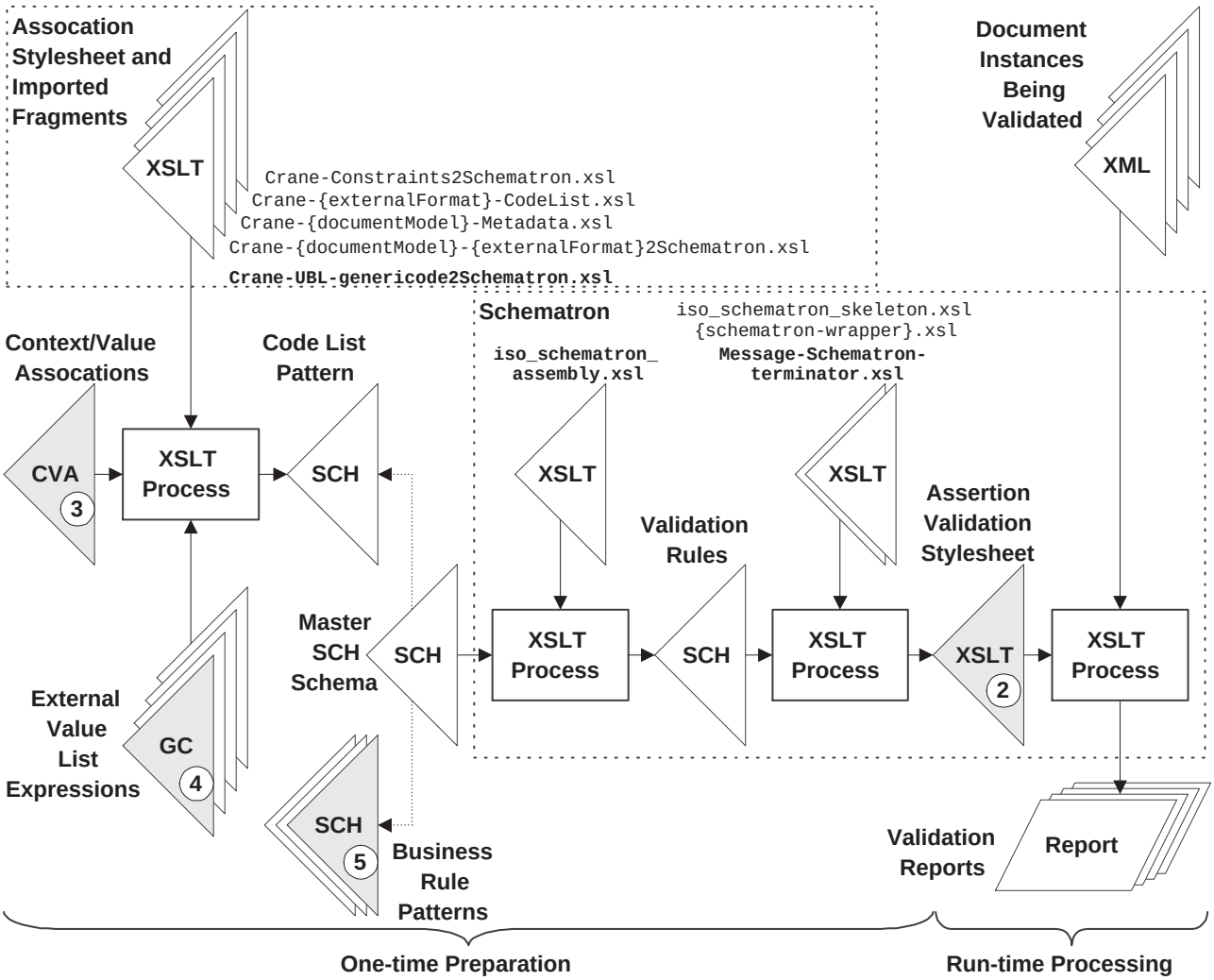
- `Crane-UBL-genericode2Schematron.xsl` (UBL elements and meta data)
- `Crane-NM-genericode2Schematron.xsl` (no meta data)

Schematron patterns assembled into a complete Schematron schema

- `schematron-ISO-assembly.xsl`

Schematron schema translated into an XSLT stylesheet

- `Message-Schematron-terminator.xsl`
  - a wrapper of the `iso_schematron_skeleton.xsl` implementation



Recall Specifying code list conformance (page 156)

Recall UBL document validation (page 119)

Copyright © Crane Softwrights Ltd.

Chapter 9 - UBL customization

- Introduction - When to customize UBL
- Section 1 - Customization considerations

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

When to customize UBL

Many criteria are considered when determining when and how to customize UBL

- the use of UBL
  - stakeholders
  - profiles of deployment
  - business vs. technical
- the scope of UBL
  - more documents in UBL than needed
  - fewer documents in UBL than needed
- the size of UBL
  - more information items in UBL than needed
  - fewer information items in UBL than needed
- the perspective of UBL
  - closed environment or open environment?
    - compatible or conformant?
  - taking advantage of existing UBL-based resources

The UBL committee is not a certification authority

- it is not in the OASIS UBL Technical Committee charter to review, comment, measure or certify a UBL customization
  - responsible only for "Standard UBL" and associated publications
  - responsible to define conformance and compatibility but not enforce it or measure it
  - committees can claim conformance or compatibility of their customization, but it is up to users to measure or confirm such is true
- however a community defines a customization is up to the community
  - following TC guidelines may help in identifying and meeting objectives
  - one is not obliged to follow TC guidelines

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

When to customize UBL (cont.)

There are common basic steps in the process of creating a UBL customization

- define the profile
    - what business function is being satisfied by the customization?
- define the processes
    - which transactions are needed to enact the business function?
- determine the documents
    - which documents are used in the transactions?
- determine the document contents
    - what information items are in the documents?
- specify the restrictions and extensions
    - which existing constructs are not needed and therefore are pruned
    - which non-existent constructs must be added to the existing constructs
- emit the other artefacts
    - what support will the user community need?

Copyright © Crane Softwrights Ltd.

Out-of-the-box UBL

Standardized UBL is defined to accommodate general accounting principles for business documents

- committee of experts included constructs necessary to communicate the essence of the business documents
- the documentary scenarios frame the choices made in the semantic concepts and expression of granularity and labels for markup

If all e-commerce systems implemented full UBL and UBL met everyone's requirements then there would be no need for customizations

- but UBL only implements 80%-90%of a general-purpose scenario
- the general-purpose solution may be so large as to be too costly to implement all of UBL
- special requirements and legacy needs may dictate going beyond UBL

UBL 2.0 defines 31 different document types

- no formal profiles in UBL 2.0
- the purpose and use of documents is illustrated using only representative workflows

UBL 2.0 defines 1972 different elements in parent/child contexts

- children are themselves parents of other elements in the nested nature of XML
- tens to hundreds of thousands of elements in full path document contexts

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Customization stakeholders

A UBL customization is an implementation of UBL tailored for a user community

- UBL defines a baseline implementation without extensions or business rules
    - needs to be tailored in a particular context to meet the objectives defined by the context
- build on UBL
    - build on top of UBL to create a customized environment for a closed set of users
- scale down UBL
    - "which of the existing UBL constructs makes sense for our user community?"
- augment UBL
    - the user community defines an extension of custom constructs on top of the UBL subset

A business has both an internal user community and an external user community

- how will a customization meet internal business requirements?
- how will a customization meet partner business requirements?

Copyright © Crane Softwrights Ltd.

Customization stakeholders (cont.)

A community could be an entire industry

- a specialized industry (e.g. aerospace) may need semantic augmentations to accommodate unique meta data
    - e.g. augmented line items in invoices

A community could be only customers or suppliers of a company

- streamlining internal processes by mandating external interfaces
- a large company may be homogenizing its interfaces
- a small company may be stepping out into standardization for internal benefits

A community could be defined by geo-political properties

- e.g. Denmark legislation for OIOXML (UBL 0.7) and OIOUBL (UBL 2.0)
- e.g. PEPPOL deployment of pan-European BII requirements (UBL 2.1)

A community could be participants in common process

- aggregating common information from multiple sources by using a single standardized format
- e.g. US Department of Transport Electronic Freight Management
    - parties from all over the world participating in the shipment of goods

A community could be customers of a particular software vendor

- offering an electronic version of a business document produced by the vendor's software

Copyright © Crane Softwrights Ltd.

Customization stakeholders (cont.)

How are members of the community of users identified?

- for legal purposes
  - e.g. having the responsibility of payment
- for communication purposes
  - e.g. successfully getting information from point A to point B

Denmark's OIOSI project for OIOUBL purchased a million EAN numbers to use as endpoints

- used as an electronic delivery address only, not a legal entity identification
- available and searchable in a public UDDI register
  - open-source software made available implementers to promote deployment
- published guidelines
  - `http://www.oioubl.info/guidelines/en/OIOUBL_GUIDE_ENDPOINT.pdf`

Existing business practices within an organization already have processes to identify and authorize legal transactions

- one of UBL's primary goals is the elimination of rekeying of information, not the retooling of business practices
- e.g. shipments and payments would be authorized in a UBL-based environment no differently than in an existing environment
  - UBL is only playing a role in getting the information between parties successfully
  - existing environments probably already have concepts of customer numbers and supplier numbers

Should the endpoint identifier be mandatory?

- using a single identifier to an internal definition of party address and contact information
  - updated using an independent change-of-information business process
- if so, are other party identification methods used for verification?
  - allow users to enter their contact information used as a backup or for verification if the endpoint identifier is entered incorrectly or other problems are experienced using it
  - should probably not be the method by which a change-of-information request is made

Copyright © Crane Softwrights Ltd.

Refining the business process models

Profiles of diverse business processes

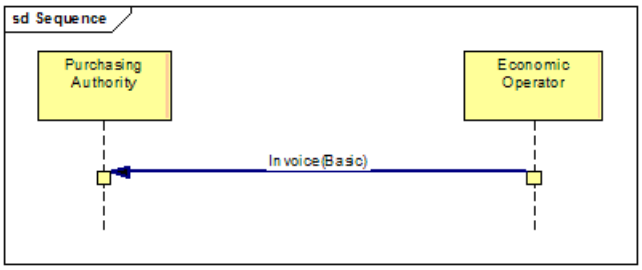
- different profiles engage the use of different sets of documents
    - e.g. the same document type can be used in different profiles of message choreography
- may require multiple customizations of one document model
- e.g. an invoice used in "simple procurement" may require far less detail than an invoice used in "advanced procurement"
    - when an invoice is raised without a formal purchase order, the simple invoice document model has no order reference information
    - when an invoice is raised with a formal purchase order, the complex invoice document model mandates order reference information

Copyright © Crane Softwrights Ltd.

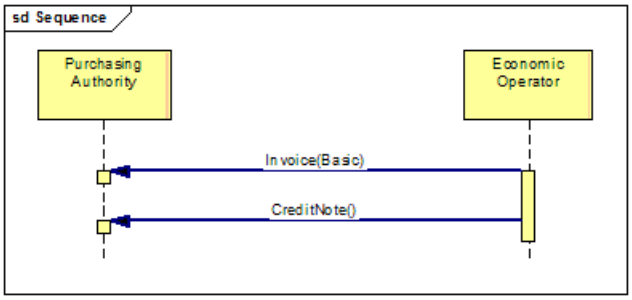
Refining the business process models (cont.)

Three of the many draft BII profiles utilizing the Invoice document are as follows:

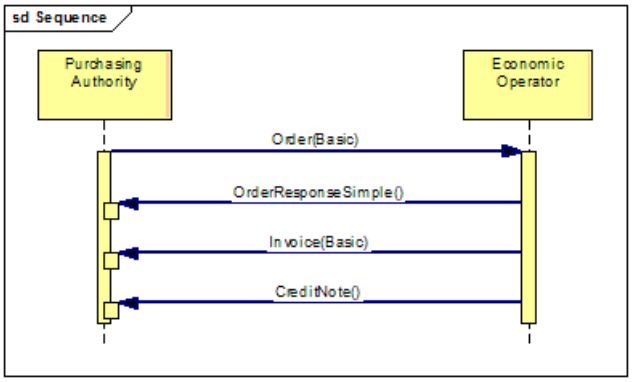
- BII04 - Basic Invoice Only



- BII05 - Basic Billing



- BII06 - Basic Procurement



The above figures are excerpted from the CEN WS-BII-WG1 phase 2 report - Report 1, Version 2.0 (undated).

Copyright © Crane Softwrights Ltd.

Technical considerations

Starting with UBL means not having to conceive of everything

- business experts participated in the UBL process
- avoiding a colloquial vocabulary avoids the problems of gaining acceptance
- can take advantage of existing processes or tools tailored to UBL
    - won't have to implement parallel processes that are already implemented for UBL
- opportunity for interoperability with other UBL deployments
    - not guaranteed to be fully interoperable because of differing customizations
    - high chance for sufficient interoperability to not inhibit accomplishing business

Business and technical reasons to not support everything

- not all constructs available are relevant to the nature of the business
    - irrelevant items can be distracting from those that are relevant
    - users may feel it necessary to populate items that end up not being used in business processes
- supporting everything in a single implementation would be a challenge
    - application development to accommodate every construct
    - testing of the support of every construct
    - rendering of every construct

Choosing to create a conformant customization can leverage standard UBL artefacts

- compatible schemas need to be created from scratch
    - tools exist for creating schema expressions based on abstract definitions
- conformant schema subsets can be mechanically derived from standard schemas
    - see Annex A - OpenOffice 3 UBL customization environment (page 229)

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Expanding the scope of coverage

To support an existing business process, a new non-UBL document might be needed

- a customization is welcome to use its own document types not already included in the UBL document suite
  - with the thought that such a document is probably a good candidate to add to a future revision of UBL
- create a new document schema in a non-UBL namespace
  - as with other UBL documents, only the document element is in the document namespace
  - all other business objects in the document schema are from a library module
- share the UBL common library
  - wherever possible use the existing business objects to promote interoperability
  - these business objects may be customized or not
- create an add-on library of business objects not already in UBL
  - use another non-UBL namespace so that the add-on library can be shared with multiple non-UBL document schemas

Requirements to the UBL committee for future consideration

- dozens of documents are already in consideration for new minor versions of UBL 2
  - identify the need for a new document type
  - contribute the add-on business objects for inclusion in the common library
- participate in the work to reduce the migration effort from the non-UBL addition to the future UBL-included specification

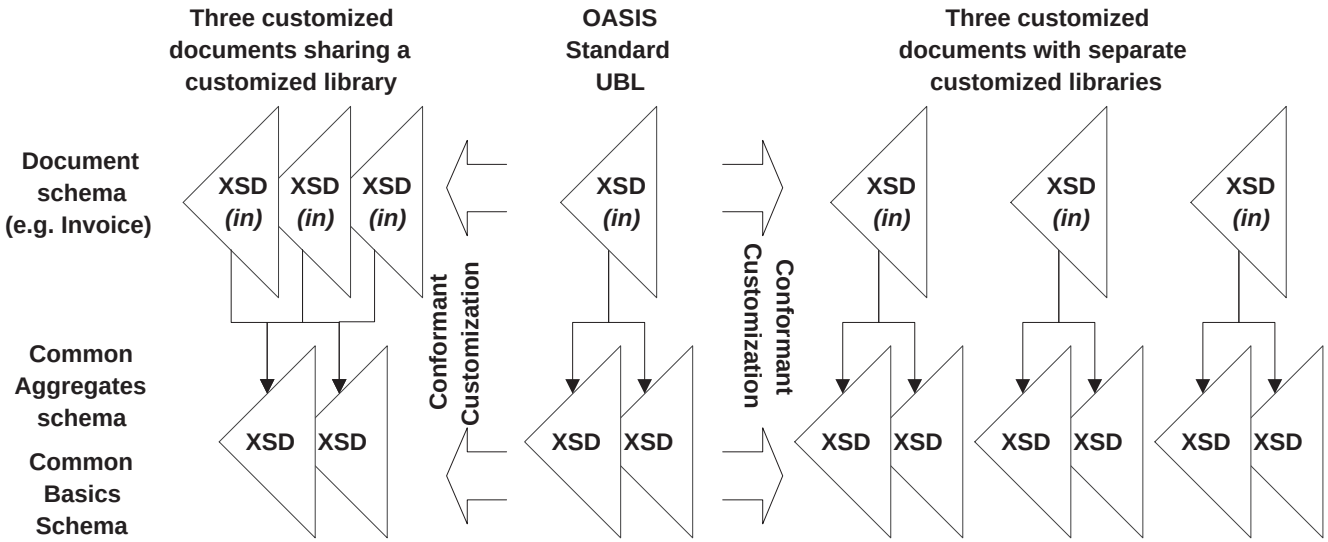
Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Profiles may affect both the common library and the document models

- may share a modified common library across different profile's customizations
- may require each profile's customization to have a different common library

Different versions of a given schema could share or distinguish common fragments

- on the left three profiles of the invoice document schema are sharing a common set of customizations of the common library
  - not expected to be very common situation
  - only "top level" constructs are defined in the document schema, thus not enough opportunity for customization
- on the right three profiles of the invoice document schema are using different customizations of the common library
  - expected to be very common situation



Recall Customizations and profiles of UBL (page 58)

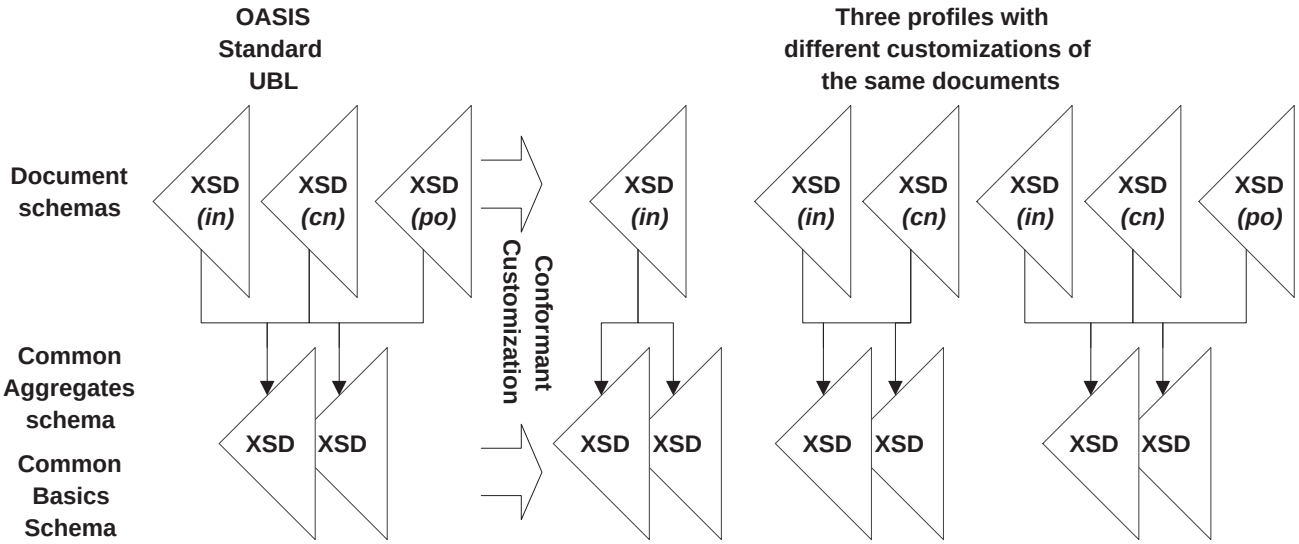
- one customized common library is used for all document customizations
- the responsibility for process-specific variations is borne by supplemental validation using Schematron

Copyright © Crane Softwrights Ltd.

Refining the business document models (cont.)

Different document schemas in a given profile would share the same customization of the common library

- the customizations of the common library are different for each profile
- all documents within any given profile are using the same common library
- the business objects are shared within a profile and different between profiles



Copyright © Crane Softwrights Ltd.

Steps refining the document model

The information model is first refined based on user requirements

- the information model is abstract and based on the business requirements
- the document model is a concrete syntactic expression of the information model
- irrespective of the document model, new information items may have an association with existing items
- e.g. an extended line item description augments the existing line item

Remove document model components not used in the information model

- only optional document model constructs can be removed

Revise document model component cardinality based on the information model

- minimums can be increased
    - e.g. an optional construct can be made mandatory in a customization
- maximums can be decreased
    - e.g. a repeated construct can be constrained to be a singleton

Copyright © Crane Softwrights Ltd.

Steps refining the document model (cont.)

Realize the information model augmentations in the document model

- compatible additions can be added anywhere in the document model
  - there is no requirement for validating the instance with the standard UBL document models
- conformant additions must utilize the extension point
  - prevents rejection of the document instance triggered by constraint violations of the standard UBL document models
  - put only the augmentations under the extension point
  - replicate the standardized information under the extension point with augmentations in context
  - utilize the extension point as an inboard repository
    - e.g. scanned images, legacy format file content, etc.
    - note: should utilize `AttachedDocument` when using outboard association of such entities with a document

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Using UBL instances as a model for customization

Rather than working from the model to the instance, work from the instance to determine the model

- create a UBL instance with the required information
    - work iteratively until all the information needed is expressed
- validate the instance with the full UBL document model
    - this confirms the content is acceptable
- use the instance to walk through the customization process
    - assume nothing but mandatory constructs are in the customization
    - walk through the model adding those elements found in the instance
        - which will likely reveal other important optional elements
    - probably (but not required to) preserve the modeled cardinality for each added construct

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Code list conformance

UBL document conformance does not include coded values

- except for a limited number of UN/CEFACT data types where coded values are embedded in the schemas
    - such exceptions are being removed from UBL 2.1
- recall UBL documents (page 101)
    - no reference in normative validation definition to code values

The community must decide to which code lists must the document values conform

- which information items in the document are constrained by controlled vocabularies?
- which controlled vocabularies are going to be used for those items?
- recall the list created and supplied by the TC for UBL (page 154)

Restricting and extending code lists must be considered

- masquerading a subset of a list as if it were the entire list
    - promotes interoperability of the coded values
    - uses best practices regarding list-level meta data

Trading partners can further restrict and extend lists for private needs

- trading relationships can change over time
- the UBL committee and the customization community cannot anticipate private trading relationship requirements

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Code list conformance (cont.)

Can choose to constrain code lists to a subset of lists provided by UBL TC

- if it is important for interoperability at the value level with a UBL 2.0 implementation using `defaultCodeList.xsl` then all values in the constrained list must also be values in the UBL published list

Can choose one's own selection of code lists for value conformance

- UBL conformance is measured at the schema level only
- `defaultCodeList.xsl` is part of an informative annex only
    - provided only as a convenience to users

In UBL 2.0 the UN/CEFACT schemas prevent extension of values for supplemental components

- `currencyID=` attribute for amount currency
- `mimeCode=` attribute for MIME type
- `unitCode=` attribute for unit of measure
- subsets of these values do not cause problems
    - no need to change the schemas
    - second-pass value validation can confirm the values used are within the subset of chosen values
- any extensions to these values will prevent an instance using such values to validate against the UBL 2.0 schemas
- in UBL 2.1 this constraint is being removed

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Identifying the customization

`cbc:CustomizationID` value

- identifies the customization as a whole
- any string can be used but a URI is recommended

`cbc:ProfileID` value

- identifies the profile within the customization
- any string can be used but a URI is recommended
- not necessary to include reference to the customization
    - the string shouldn't be used in isolation without consideration of the customization identifier

URI string can be discoverable or not

- a discoverable URI is a URL that can be accessed with HTTP without further dereferencing
    - e.g. `http://www.CraneSoftwrights.com/ns/CraneUBL`
        - points to a directory in which an `index.html` file is returned
        - the file is in XHTML with RDDL attributes
    - e.g. `urn:x-Crane:CraneUBL:InvoiceOnly`
        - this URI cannot be resolved to a URL without some kind of redirection lookup
        - the lookup could very well return a URL as in the earlier example

Use of Resource Directory Description Language (RDDL)

- `http://www.rddl.org`
- a micro-vocabulary of attributes to add to the XHTML anchor element `<a>`
- formally identifies resources by their nature, purpose and format
- allows a program mining the document to find other related resources and documents

Copyright © Crane Softwrights Ltd.

Announcing the customization

How will you get word out regarding the customization you've created?

- announce on `http://ubl.xml.org`
- announce and discuss on UBL-Dev
    - `http://lists.oasis-open.org/archives/ubl-dev/`
    - `http://www.oasis-open.org/mlmanage/`

Copyright © Crane Softwrights Ltd.

Chapter 10 - Customization specification

- Introduction - Customization specification
- Section 1 - Defining a customization of UBL

Outcomes

- review an approach towards defining a customization of UBL

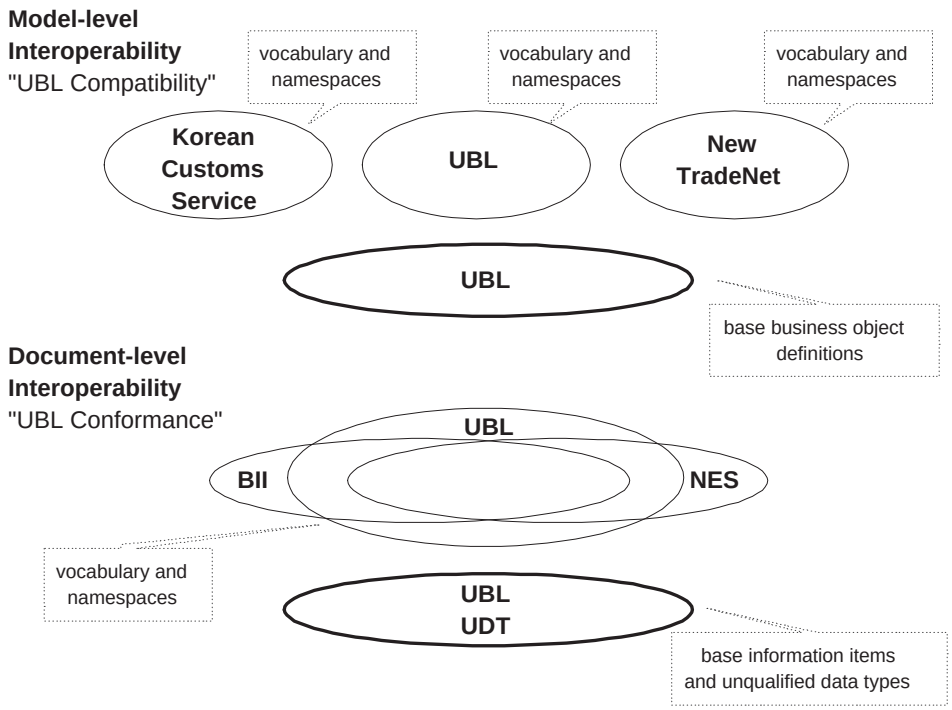
Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

At least two ways to view interoperability when customizing UBL for one's own purposes

- model-level interoperability (UBL compatibility)
    - objective to be create new information items and messages based on UBL library of constructs
- document-level interoperability (UBL conformance)
    - objective to be able to interchange UBL messages

In these two diagrams, the thin-edged ovals represent the set of vocabulary information items (names and namespaces) for an implementation of UBL, while the thick-edged ovals represent the basis upon which the vocabularies are derived

- the top diagram illustrates how there is no overlap of vocabularies between UBL, the Korean Customs Service implementation of UBL and the New TradeNet implementation of UBL
    - the business objects in all vocabularies are derived from UBL business objects
- the bottom diagram illustrates how there is a big overlap of vocabulary between UBL, the North European Subset (NES), and the BII vocabulary
    - both the NES and BII vocabularies use a subset of the UBL vocabulary
        - opportunity to add extensions, though in practice this hasn't yet happened
    - all three vocabularies are based on the information items and unqualified data types of UBL, thus the XSD module is utilized in the declaration of the constructs



Copyright © Crane Softwrights Ltd.

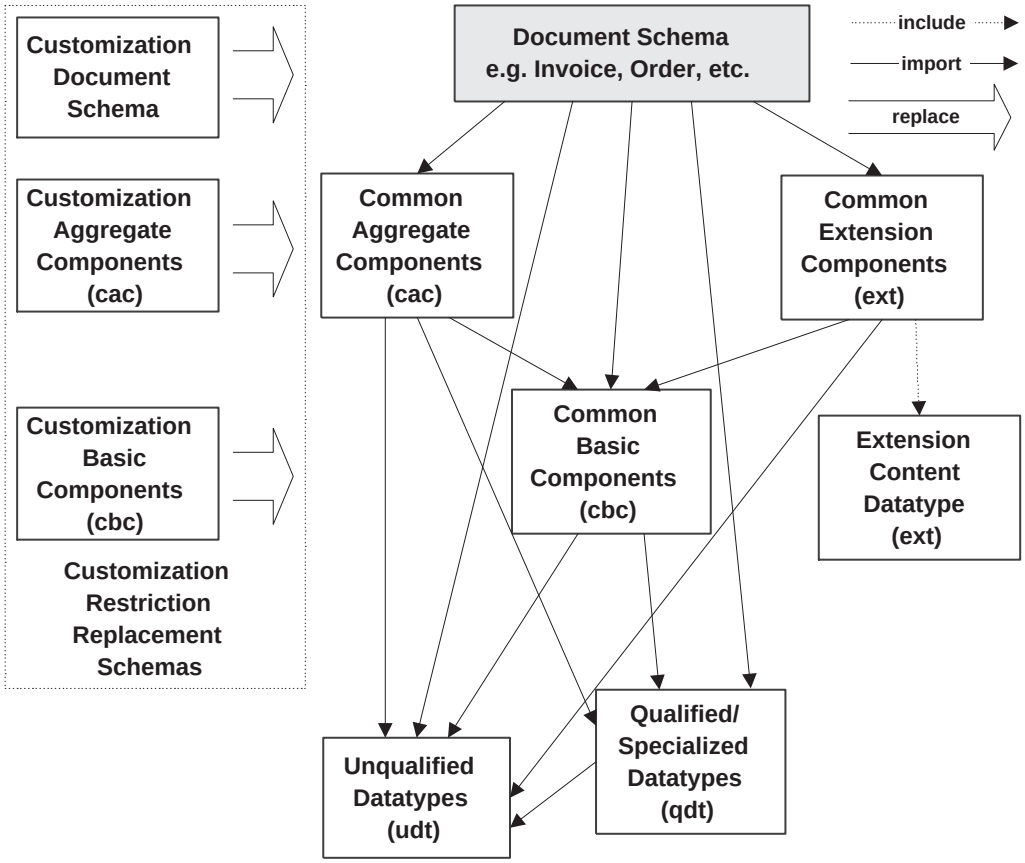
Expressing a conformant UBL subset

The W3C schema modules for a conformant UBL subset express the customization constraints

- customization fragments partially overwrite a copy of the UBL `xsd/` and `xsdrt/` directories

The following subset schema fragments replace the UBL schema fragments:

- the document schema (e.g. Statement, Waybill, etc.)
- the definition of ABIE (and, therefore, ASBIE) constructs
- the definition of BBIE constructs
- all unused constructs are removed from the fragments



Copyright © Crane Softwrights Ltd.

Could consider working with the UML models (see page 81)

- the UML models were created with the Sparx Enterprise Architect 6.5 tool and are available here:
- `http://www.oasis-open.org/committees/document.php?document_id=21918`

Instance level conformance is of secondary (or no) consideration

- for interoperability the business objects of the compatible schema should be derived from the business objects of UBL

Copyright © Crane Softwrights Ltd.

Defining a conformant UBL subset

A conformant subset of UBL is one whose constraints are constrained by UBL

- every valid instance of the conformant subset document model is a valid instance of UBL

Allowed customization changes in sequence and cardinality

- cannot rearrange elements
- limited changes to cardinality
    - minimums can be increased
    - maximums can be decreased

Applying the constraints to the normative document schemas

- any process creating a suite of schema conforming simultaneously to the UBL NDR and the constraints of the UBL model is acceptable
    - e.g. create W3C Schema XSD files from scratch
- Crane's approach in `Crane-UBLProfile` is to transform the UBL W3C Schema XSD files
    - original declarations of elided elements are commented out
    - original cardinalities of changed elements are added as comments
    - the entire original OASIS W3C Schema expression can be seen in the actual declarations or commented declarations in each file
        - thus all changes can be visually confirmed if one is uncertain
- those constructs not used anywhere in the hierarchy of the document schemas are removed from the fragments

Copyright © Crane Softwrights Ltd.

Strict and permitted subsets

At least two ways to approach specifying a subset of a UBL information model

- these were proposed in early research work done by and for the Government of Denmark
- a strict subset
    - nothing is wanted except those items with any specified cardinalities
- a permitted subset
    - everything is wanted except those items with a zero cardinality

Mandatory items are obliged to be in all subsets

- otherwise an instance of the customization without the mandatory item would not be a valid instance of UBL

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Limitations and cautions to pruning

Recall the UBL document constraint regarding empty elements (page 101)

- a UBL document is not allowed to have empty elements
- many of the constructs are optional
- very easy when defining a strict subset to end up with no child elements specified for a parent

No ABIE can have all of its items pruned if an ASBIE might use it

- it is irrelevant if an ABIE has no children if that ABIE is never used for a given document model
- one need only concentrate on those ABIE constructs that have been referenced by ASBIE constructs, recursively, from the document ABIE

`Crane-UBLProfile` reports ABIE constructs where all of the children have been pruned

- reported in the summary report and HTML reports

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Acknowledging OASIS copyright

Guidelines from UBL TC regarding UBL customizations

- `http://lists.oasis-open.org/archives/ubl/200806/msg00002.html`

Requirements for inclusion in derivative works:

- OASIS policies support implementations, subsetting and extensions of OASIS works, so long as they respect conformance, in the sense of not claiming compliance with an OASIS work, or identity with an OASIS work, incorrectly.

  Specifications and models published for use by others that incorporate OASIS work should include the following in an appropriate place, usually near the author's own copyright notice:

  Portions copyright © OASIS Open 200[8]. All Rights Reserved.
- `http://lists.oasis-open.org/archives/ubl/200802/msg00038.html`
  - guidelines received from OASIS executives

Copyright © Crane Softwrights Ltd.

Chapter 11 - Conformant customization implementation

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Conformant customization implementation

An implementation of a customization need only support the customization definition

- the community has agreed on what parts of UBL will be used
- the application can limit itself to only the parts expected

Customization validation can only check the customization definition

- requires unexpected constructs to be pruned from instances before processing

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Conformant customization implementation (cont.)

Two different implementations of conformant customizations may or may not be able to exchange UBL documents

- the use of XML and the common vocabulary allows much of the information to be interchange
- two communities may make different choices of which optional constructs to support

Only mandatory elements are guaranteed to be interchangeable

- e.g. addresses may not be interoperable
  - customization A restricts addresses to structured components only
    - an address is comprised of separately-labeled pieces
    - ```
      <cac:Address>
        <cbc:StreetName>City Road</cbc:StreetName>
        <cbc:BuildingName>Banking House</cbc:BuildingName>
        <cbc:BuildingNumber>12</cbc:BuildingNumber>
        <cbc:CityName>London</cbc:CityName>
        <cbc:PostalZone>AQ1 6TH</cbc:PostalZone>
        <cbc:CountrySubentity>London</cbc:CountrySubentity>
        <cac:AddressLine>
          <cbc:Line>5th Floor</cbc:Line>
        </cac:AddressLine>
        <cac:Country>
          <cbc:IdentificationCode>GB</cbc:IdentificationCode>
        </cac:Country>
      </cac:Address>
      ```
  - customization B restricts addresses to unstructured components only
    - an address is simply a set of address lines
    - ```
      <cac:Address>
        <cac:AddressLine>
          <cbc:Line>5th Floor</cbc:Line>
          <cbc:Line>Banking House</cbc:Line>
          <cbc:Line>12 City Road</cbc:Line>
          <cbc:Line>London, England AQ1 CTH</cbc:Line>
        </cac:AddressLine>
      </cac:Address>
      ```

Out-of-band processes and business practices can address mismatched expectations of UBL

- a rejection in processing a UBL document is not an automatic rejection of business
- it is a business decision regarding how to do business when information is complete

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Refining the document processing model

There is no set processing model for handling customized UBL instances

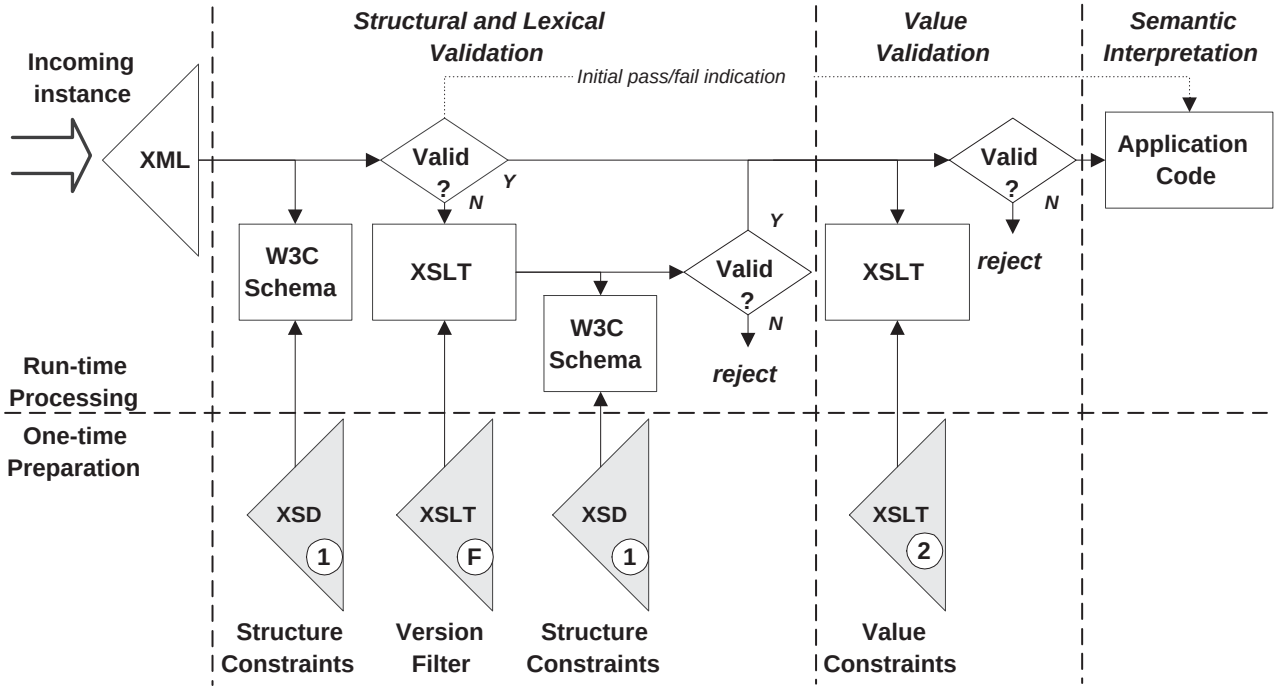
- just as there is no set processing model for handling standard UBL instances
- the information in this module is but an example for consideration

Recall the processing model published in the UBL 2.0 specification

- see UBL document validation (page 119)
  - first step confirms the structural and lexical constraints on the instance
- see Specifying code list conformance (page 156)
  - second step confirms the standardized and trading partner values used in the instance

Customization suggests the pre-validation filter removal of unexpected constructs

- the remainder of the process remains the same
- the files in use for the remainder of the process are defined for the customization
- also implements "forward compatibility" of UBL minor versions



The version filter (F) represents a customization-specific filter

- preserves only the constructs recognized by a customization definition

Copyright © Crane Softwrights Ltd.

Passing and failing validation

Passing validation is required for some applications to act on XML instances

- recall the steps in the customization process model on page 190
- an application not relying on validation but only on well-formed XML can process any instance whether it conforms to the customization or not
    - the information set is the text information found in an XML document without considering markup or syntax
- some programming languages have interfaces to XML documents that act not on the marked-up information set but on the post schema validation information set (PSVI) defined by W3C Schema
    - the PSVI interface delivers data-type-rich information to the application from the XML instance
    - if validation fails, the application may be prevented from even inspecting the content of the document

Passing initial validation indicates all expected information is present

- doesn't necessarily mean the instance was designed for use in the profile or customization
- an application can inspect the instance's declaration of conformance and use
    - such content is advised to be present but not required to be present
    - a community can make such content mandatory if it wishes, risking not being able to receive business content that is otherwise suitable
- `cbc:VersionID` - version of the UBL document model
    - e.g. "2.0", "2.1", etc.
- `cbc:CustomizationID` - domain of definition and use
    - e.g. community of users
- `cbc:ProfileID` - scenario of definition and use
    - e.g. business process

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Passing and failing validation (cont.)

Failing initial validation

- recall the steps in the customization process model on page 190
- the application knows that some of the sender's information has been pruned from the instance
- e.g. a member of the community included unexpected information
- e.g. the instance conforms to another community definition

Failing secondary validation

- the instance is not suitable for application processing because it is missing expected mandatory information
- after having filtered for only expected constructs means there is nothing unexpected in the instance

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Out of band decision making

UBL validation is not meant to interrupt business, only alleviate effort from applications

- when an instance passes structural and value validation, the user knows the application is going to find suitable data for making business decisions
- a rejection of a UBL document during validation is only a signal of bad data, not bad business
- a primary object of UBL is the elimination of the rekeying of information
    - not a magic bullet that somehow "enables" electronic commerce

Businesses probably have existing workflows accommodating incorrect information

- unidentified customers
- invalid values on paper documents

Business practices should be reviewed for all aspects of UBL integration

- how an application makes a business decision to spend money or ship product when the data is entirely correctly formed
- how the business reviews a UBL document that is rejected during validation
- how the business reviews a transaction that is rejected by the application

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 12 - Introduction to document engineering

- Introduction - Introduction to document engineering
- Section 1 - Functional dependency
- Section 2 - Creating new elements

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Introduction to document engineering

Document Engineering (the book)

- ISBN 0-262-07261-0 - Robert J. Glushko, Tim McGrath
- formal and rigorous document modeling techniques
- models, patterns and re-use
- interoperability or lack thereof

The role of documents in business transactions

- distinguishing models and documents
- designing business patterns
- implementing models and documents in applications

Techniques for deriving the model and structure of documents

- analyzing documents
- assembling document models

Copyright © Crane Softwrights Ltd.

Functional dependency

Chapter 12 - Introduction to document engineering
Section 1 - Functional dependency

Functional dependency is an important principle

- set of data analysis techniques called "normalization"
- used by database designers to create relational models
- adapted by McGrath and Glushko to create document models

Normalization captures data semantics

- from an analysis of information requirements
- minimizes redundancy and duplication
- re-uses identified common patterns of information

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Essentiality

Essentiality reduces waste and distraction

- determining about the essential nature of components
- avoids modeling derivative components
    - identifies where the semantics are similar enough to adapt already-modeled constructs rather than creating a similar-but-not-quite-the-same construct

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Structural patterns

Promoting interoperability

- rearranging the order of information items so that two aggregates can be implemented with the same constructs
- possibility to distill those same constructs into a common child aggregate

Identifying patterns by removing qualifications

- are some qualifications absolutely necessary for interchange?

Encouraging standardization

- can promote faster and more robust application development by sharing code that works on many business objects

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Checks and balances

Ensuring no similarly qualified components within the same structure

- reduces duplication

Are items sharing conceptual context, thus inferring more structural context

- e.g. country code and country name in address
    - suggests a country context with code and name members

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Creating new elements

Considering re-use vs. renaming vs. specialization

- rather than creating new elements with parallel definitions, basing a new element on an existing element will promote better code re-use
- promotes normalization of information

For example, consider the base definition of a party:

- information regarding an organization or individual
- `Party. Details` defined as an ABIE

When referring to the "owner party", there is no additional information

- `Transport Means. Owner_ Party. Party` defined as an ASBIE
- simply points to `Party. Details`
- qualified by its context of use and the party the structure represents

When referring to the "customer party", there is additional information

- `Customer Party. Details` needs to be its own ABIE
- within the ABIE is an ASBIE directly to Party
- other BBIE and ASBIE information
    - e.g. delivery contact, accounting contact, buyer contact, identifiers, etc.

When referring to the "buyer customer party", there is no additional information

- `Despatch Advice. Buyer_ Customer Party. Customer Party` defined as an ASBIE
- qualified by its context of use and the customer party the structure represents

Copyright © Crane Softwrights Ltd.

Creating new elements (cont.)

The ASBIE is named the ABIE if the ABIE is conceptually and unambiguously referring to the object class

- e.g. `Statement Line. Details` has the ASBIE `Statement Line. Exchange Rate` pointing to `Exchange Rate. Details`
  - there is only one exchange rate to refer to (statement line currency and related document currency)

The ASBIE is qualified when there are variations of use of the same ABIE

- e.g. invoice has many exchange rates "`Invoice. Pricing_ Exchange Rate. Exchange Rate`" and "`Invoice. Payment_ Exchange Rate. ExchangeRate`" and "`Invoice. Tax_ Exchange Rate. Exchange Rate`"

The ASBIE is qualified when the relationship is not obvious

- e.g. `Transport Means. Owner_ Party. Party` qualifies party with `Owner_`
  - because many parties could be related to the transport means

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Dictionary entry naming

Recall the structure of the property term in a dictionary entry name

- outlined on page 70
- `{qualifier_} {possessive-noun} primary-noun`

When to use qualifiers or possessive nouns in property terms

- identifiers of different kinds of things have different possessive nouns
- different identifiers of the same kind of thing have different qualifiers

e.g. in "`Address. Details`"

- "`Address. City Name. Name`" and "`Address. Street Name. Name`" are names of different kinds of things
  - both are names but the names belong to (are possessed by) different concepts of cities and streets
- "`Address. Street Name. Name`" and "`Address. Additional_ Street Name. Name`" are names of the same kinds of things
  - both are street names but the qualification of which street name is different

Copyright © Crane Softwrights Ltd.

Dictionary entry naming (cont.)

e.g. `Forwarding Instructions. Details`

- `Forwarding Instructions. UBL Version Identifier. Identifier`
  - `<UBLVersionID>` is not a forwarding instructions identifier
- `Forwarding Instructions. Customization Identifier. Identifier`
  - `<CustomizationID>` is not a forwarding instructions identifier
- `Forwarding Instructions. Profile Identifier. Identifier`
  - `<ProfileID>` is not a forwarding instructions identifier
- `Forwarding Instructions. Identifier`
  - `<ID>` is a forwarding instructions identifier
- `Forwarding Instructions. Carrier Assigned_ Identifier. Identifier`
  - `<CarrierAssignedID>` is also a forwarding instructions identifier

Note how an unqualified ASBIE has same property term as representation term, so representation term is suppressed

- refer to section 4 of the UBL Naming and Design Rules

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 13 - Customization extension

Copyright © Crane Softwrights Ltd.

Customization extension

Many uses for adding extension information to the document model

- customization augmentation
  - new information items are associated with existing constructs
- supplemental information
  - e.g. bit image scan
    - including a picture to back up the choices made in the data
  - e.g. legacy format
    - useful for round-tripping information to different formats

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Conformance vs. compatibility

At an abstract level, a customized model is somehow different than the standard model

- certain standardized optional content is prohibited in the customization
- the customization may have additional non-standardized content

Compatibility

- model-level interoperability
- business objects in compatible models are based on UBL business objects
- inevitable document constraint violations against the standard UBL schemas
    - non-UBL business objects must have non-UBL names

Conformance

- instance-level interoperability
- business objects in conformant models are bona fide UBL business objects
- no document constraint violations against the standard UBL schemas
- simplest conformant schema is a pure subset without extensions
    - also a good starting point to determine if extensions are even needed

Implicitly a conformant customization is a compatible customization

- but a compatible customization is not a conformant customization
    - due to the presence of non-UBL constructs

Copyright © Crane Softwrights Ltd.

Conformance vs. compatibility (cont.)

Extensions to the document model in a compatible schema are anywhere in the document

- because they are not constrained by the standard structures
- typically the information

Extensions to the document model in a conformant schema belong under the document extension point

- extensions should be developed using compatible schema creation techniques
- good practice to learn compatible derivation methodologies
- choices are available for how to capture the compatible information
    - extension content is necessarily separate from the standardized content
    - standardized content may be duplicated with the extended content

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

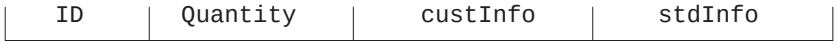
Illustration of extension approaches

Consider the need to extend standardized invoice information with custom information

- new custom BIE entities are important to a particular community
- e.g. custom line item information

The extended information record for the custom line item model:

Extended information record



- `ID`
  - standardized line item identifier
- `Quantity`
  - standardized quantity information
- `custInfo`
  - customized information not found in the standard model
- `stdInfo`
  - other standardized information already found in the standard model

To simplify the diagrams, other invoice information is not included

- bill to, ship to, etc.
- any of the aggregates may be customized as the line item aggregate is in this example

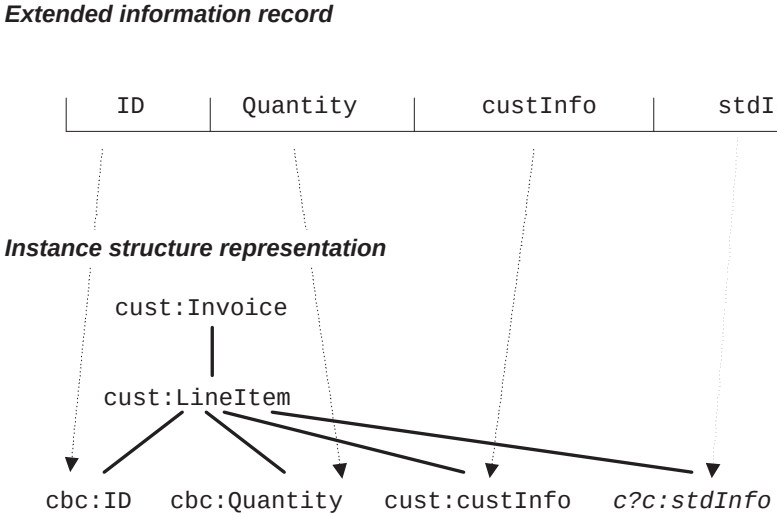
Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Illustration of extension approaches (cont.)

A compatible instance has arbitrary content based on the model

- good derivation practices would re-use UBL constructs where the semantics are identical
- no real constraint at the document level to use the UBL namespaces, but may help the reader or implementer of the customization

The instance structure representation for the compatible document model:



- `cust:Invoice`
  - the document element is necessarily in a non-UBL namespace because the line item isn't a UBL line item
- `cust:LineItem`
  - the line item is necessarily in a non-UBL namespace because there are non-UBL children
- `cbc:ID`
  - standardized line item identifier
- `cbc:Quantity`
  - standardized quantity information
- `cust:custInfo`
  - customized information not found in the standard document
- `c?c:stdInfo`
  - other standardized information already found in the standard document

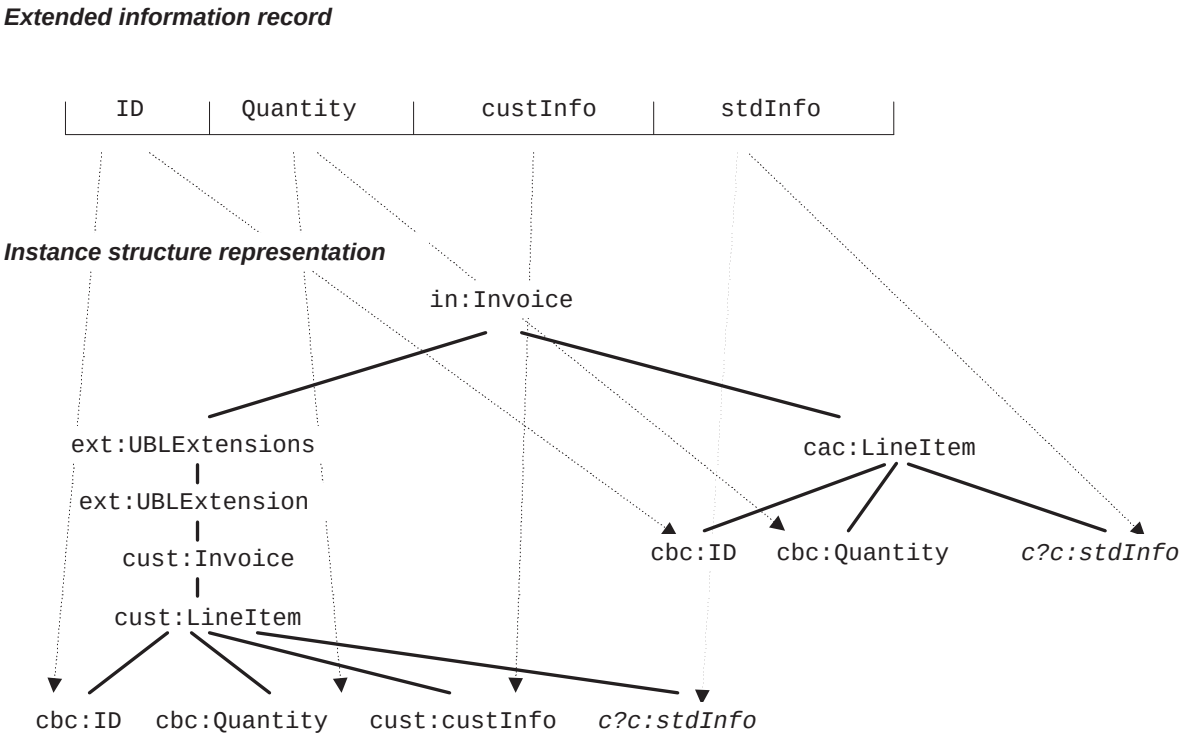
Copyright © Crane Softwrights Ltd.

Illustration of extension approaches (cont.)

A conformant instance can copy a complete compatible instance under the extension point

- pro: all information about the extended line item is found in one place in the document
- con: standardized line item information is duplicated in two places in the document
  - question of the integrity of the information (what if corresponding items are unequal)

An instance structure representation for a conformant document model:



- `in:Invoice`
  - the UBL standard document element
- `cac:LineItem`
  - the UBL standard line item and its children
- `cust:Invoice`
  - the apex extension element is the complete compatible instance

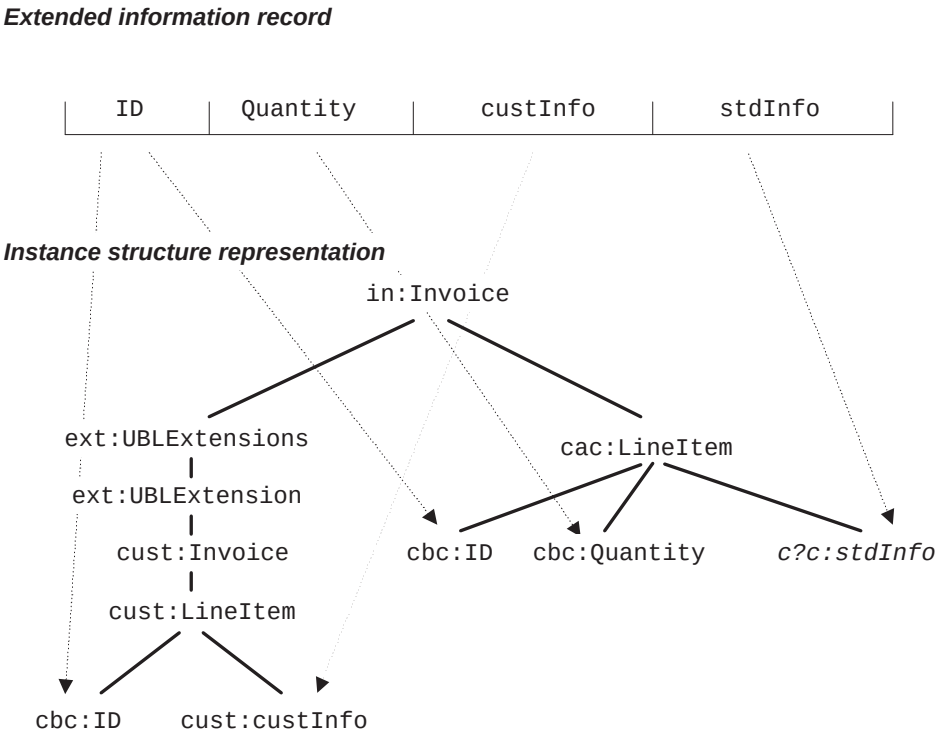
Copyright © Crane Softwrights Ltd.

Illustration of extension approaches (cont.)

A conformant instance can keep only a fragment of the compatible instance under the extension point

- pro: no duplicated information in the instance
- con: the application needs to compose the extended information record from fragments
  - question of the linking/association/reference between fragments

An instance structure representation for a conformant document model:



- `cust:Invoice`
  - the apex extension element is necessarily in a non-UBL namespace because the line item isn't a UBL line item
- `cust:LineItem`
  - the line item is necessarily in a non-UBL namespace because there are non-UBL children
- `cbc:ID`
  - standardized line item identifier
- `cbc:Quantity`
  - standardized quantity information
- `cust:custInfo`
  - customized information not found in the standard document
- `c?c:stdInfo`
  - other standardized information already found in the standard document

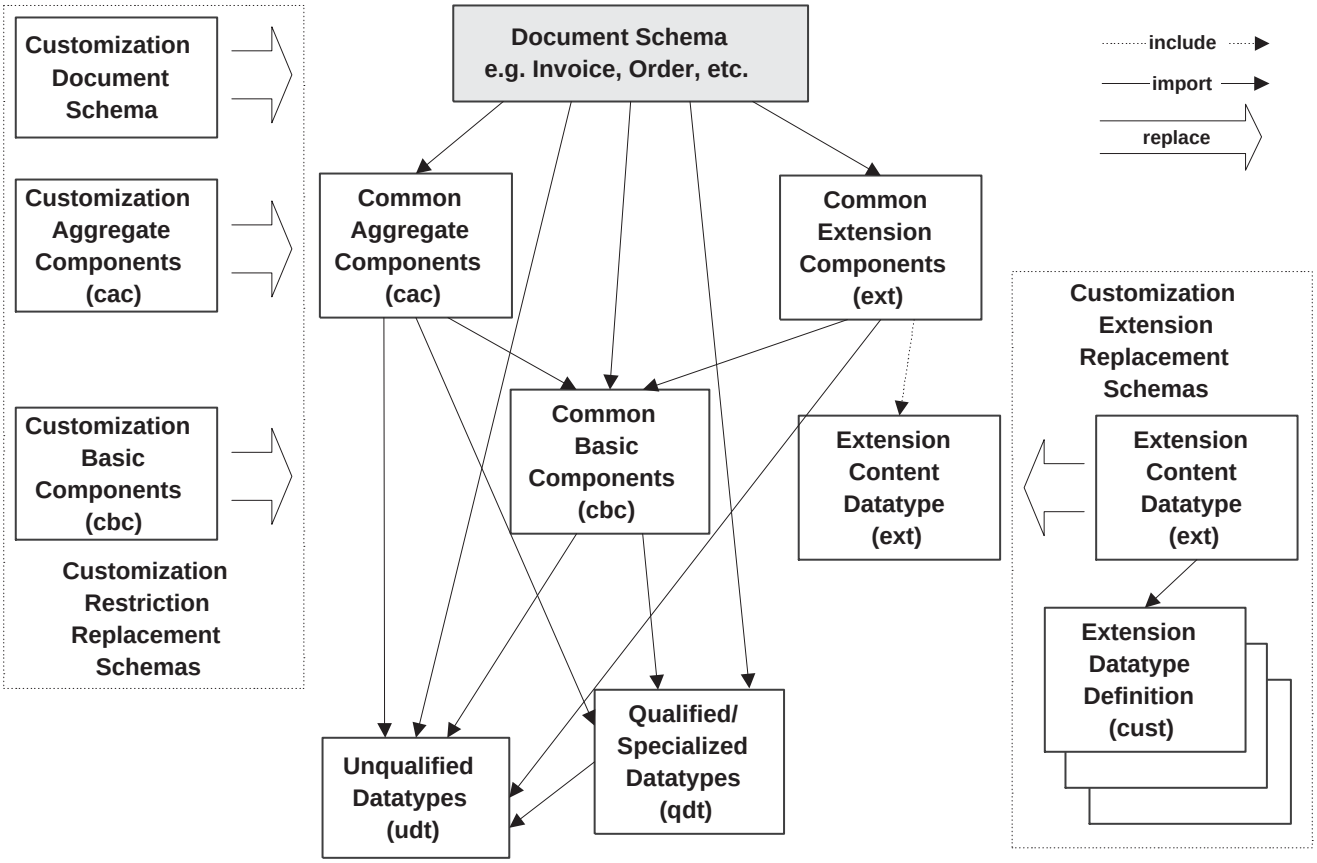
Copyright © Crane Softwrights Ltd.

Deploying an extension

Chapter 13 - Customization extension
Section 2 - Deploying an extension

Conformant customization schema structure

- start with copies of all of the UBL schemas related to a given document type
  - preserving the subdirectory structure of the fragments
- overwrite ABIE and BBIE definitions with the customized versions:
  - restricted versions of the document, aggregate and basic schemas replace corresponding fragments
- overwrite unconstrained extension specification with customized constraints:
  - extended version of the content data type schema replaces the corresponding fragment
- extension data type definition schemas are added to the directories



Compatible customization schema

- any structure is allowed because there are no constraints on namespaces
- a good practice to import the UBL aggregate and basic components for re-use

Copyright © Crane Softwrights Ltd.

Extension meta data

Extension meta data describes the extension without having to look in the extension

- all extension meta data is optional
- declared in `common/UBL-CommonExtensionComponents-2.0.xsd`

This example contains one of each meta data item defined for extensions:

```
<Invoice xmlns="urn:oasis:...:xsd:Invoice-2"
         xmlns:cbc="urn:oasis:...:xsd:CommonBasicComponents-2"
         xmlns:cac="urn:oasis:...:xsd:CommonAggregateComponents-2"
         xmlns:ext="urn:oasis:...:xsd:CommonExtensionComponents-2"
         xmlns:demo="urn:x-Demo:Demo">
  <ext:UBLExtensions>
    <ext:UBLExtension>
      <cbc:ID>Demo1</cbc:ID>
      <cbc:Name>Demonstration</cbc:Name>
      <ext:ExtensionAgencyID>CSL</ext:ExtensionAgencyID>
      <ext:ExtensionAgencyName>Crane Softwrights Ltd.
</ext:ExtensionAgencyName>
      <ext:ExtensionVersionID>0.1</ext:ExtensionVersionID>
      <ext:ExtensionAgencyURI>http://www.CraneSoftwrights.com/
links/res-dev.htm</ext:ExtensionAgencyURI>
      <ext:ExtensionURI>urn:x-Demo:Demo:0.1</ext:ExtensionURI>
      <ext:ExtensionReasonCode listURI="urn:x-Demo:Demo:ReasonCodes">1
</ext:ExtensionReasonCode>
      <ext:ExtensionReason>Illustration</ext:ExtensionReason>
      <ext:ExtensionContent>
        <demo:Demo>
          <demo:Thing>This is a test</demo:Thing>
          <cbc:ID>DemoTest</cbc:ID>
          <demo:Total currencyID="GBP">100.00</demo:Total>
        </demo:Demo>
      </ext:ExtensionContent>
    </ext:UBLExtension>
  </ext:UBLExtensions>

  <cbc:ID>A00095678</cbc:ID>
  <cbc:IssueDate>2005-06-21</cbc:IssueDate>
  <cbc:Note>sample</cbc:Note>
  <cac:AccountingSupplierParty>
    <cac:Party>
      <cac:PartyName>
...
```

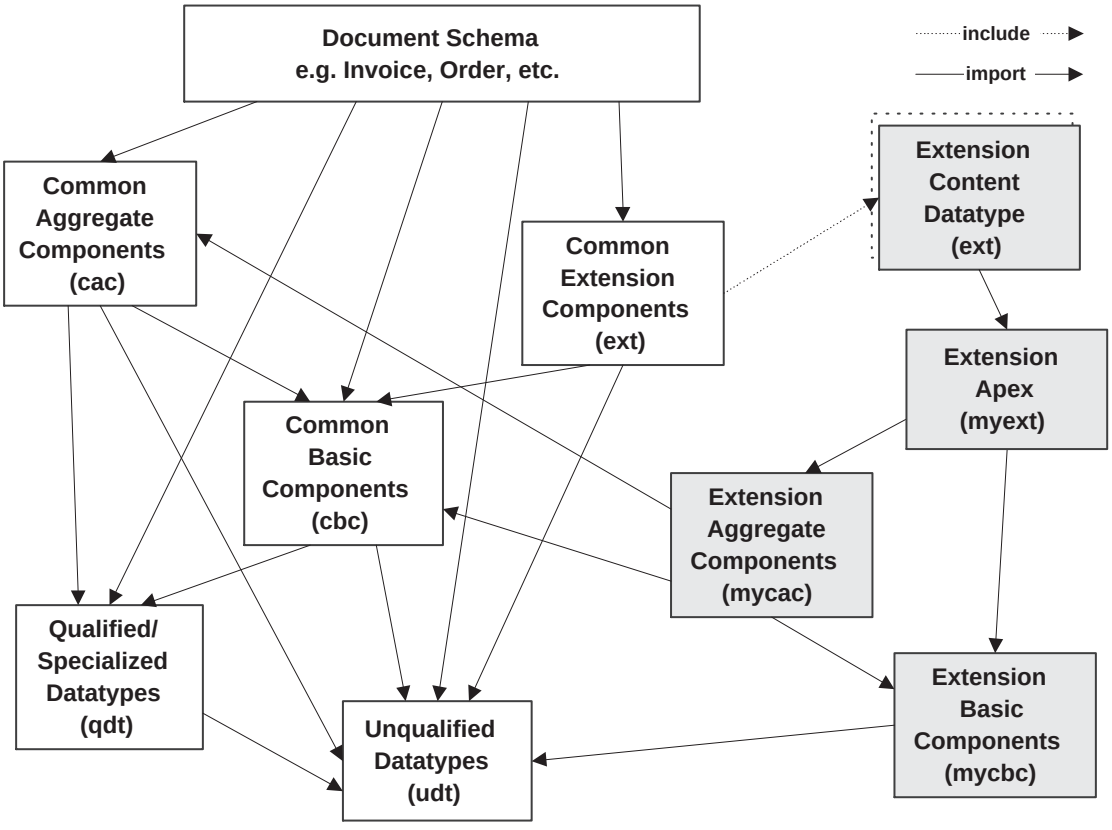

Copyright © Crane Softwrights Ltd.

Extending a UBL document

Chapter 13 - Customization extension
Section 2 - Deploying an extension

Extending an existing UBL document involves only defining new constructs under the extension point

- replace the UBL standard `UBL-ExtensionContentDatatype-2.0.xsd` fragment with a fragment that allows one of only two elements
  - the one apex element of the extension, in the extension namespace
  - any one element in any namespace other than the extension namespace
- add a definition of the apex element used to wrap all of the extensions
  - no corollary in the UBL library for this element, so this example uses a separate namespace
- add a definition of the extension ABIE and ASBIE constructs
  - use an extension namespace reserved for aggregate constructs
  - candidate additions to a future revision of UBL
- add a definition of the extension BBIE constructs
  - use an extension namespace reserved for basic constructs
  - candidate additions to a future revision of UBL

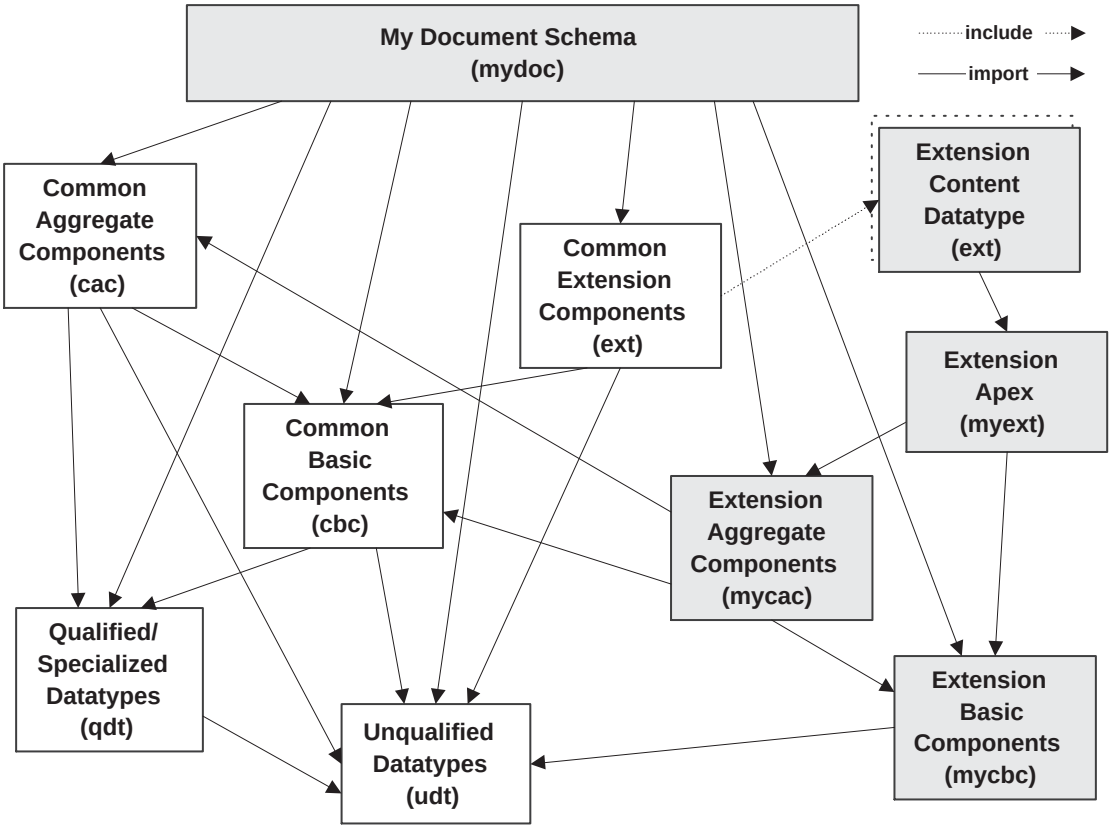


Copyright © Crane Softwrights Ltd.

Adding a new non-UBL document

Adding a new non-UBL document involves defining a new document element and new constructs under the extension point

- the new document schema adds to the set of existing UBL document schemas
  - use an extension namespace reserved for the document element only
  - this schema imports the extension aggregate and basic schemas in order to specify the use of top-level (children of the document element) ASBIE and BBIE extension constructs
- all the other modifications for an extended schema also apply as in Extending a UBL document (page 214)
  - a separate extension namespace for aggregate and basic extension constructs
    - and both different than the extension document element namespace
  - could choose to re-use the extension document element namespace for the extension apex element namespace if one wishes to reduce the number of new namespaces



Copyright © Crane Softwrights Ltd.

Sample extended UBL invoice

Following the deployment strategy for extending an existing document (page 214)

- `publd-zip/samp/custext/xsd` has a suite of extension schema fragments
- `publd-zip/samp/custext/xml` has an extended XML instance
  - two extensions, one unexpected "`junk`" and one expected "`my`"

```
<Invoice xmlns:cbc="urn:oasis:...:CommonBasicComponents-2"
         xmlns:cac="urn:oasis:...:CommonAggregateComponents-2"
         xmlns=    "urn:oasis:...:Invoice-2"
         xmlns:ext="urn:oasis:...:CommonExtensionComponents-2"
         xmlns:my=   "urn:x-company:myExtensions"
         xmlns:mycac="urn:x-company:myExtensionsAggregateComponents"
         xmlns:mycbc="urn:x-company:myExtensionsBasicComponents">
 <ext:UBLExtensions>
   <ext:UBLExtension>
     <ext:ExtensionContent>
       <junk:junk xmlns:junk="abc"/>
       ...
   <ext:UBLExtension>
     <ext:ExtensionContent>
       <my:start>
         <mycac:InvoiceLine>
           <cbc:ID>A</cbc:ID>
           <mycbc:NewFacetAmount
                currencyID="GBP">123.45</mycbc:NewFacetAmount>
           ...
 </ext:UBLExtensions>
<cbc:UBLVersionID>2.0</cbc:UBLVersionID>
<cbc:CustomizationID>urn:oasis:...</cbc:CustomizationID>
...
<cac:InvoiceLine>
 <cbc:ID>A</cbc:ID>
 <cbc:InvoicedQuantity unitCode="KG">100</cbc:InvoicedQuantity>
 ...
```

Uses the fragmented approach (page 211)

- the UBL standard invoice line is extended with the new `<NewFacetAmount>` element
- the correlation between the extension portion and the standard portion of the invoice line is made through the matching `<ID>` element value

Copyright © Crane Softwrights Ltd.

Replacement extension content data type

Following the deployment strategy for extending an existing document (page 214)

- `publd-zip/samp/custext/xsd/common/UBL-ExtensionContentDatatype-2.0.xsd` is a replacement of the UBL standard `UBL-ExtensionContentDatatype-2.0.xsd`

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:oasis:...:CommonExtensionComponents-2"
            targetNamespace="urn:oasis:...:CommonExtensionComponents-2"

            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            xmlns:my="urn:x-company:myExtensions">
  <xsd:import namespace="urn:x-company:myExtensions"
              schemaLocation="MyExtensions.xsd"/>
  <!-- ===== Type Declaration ===== -->
  <xsd:complexType name="ExtensionContentType">
   <!--only one element is allowed as the child of the extension point-->
    <xsd:choice minOccurs="0" maxOccurs="1">
      <!--allow any customization other than own-->
      <xsd:group ref="my:other"/>
      <!--allow own customization-->
      <xsd:element ref="my:start"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:schema>
```


Copyright © Crane Softwrights Ltd.

Extension apex definition

Chapter 13 - Customization extension
Section 3 - Sample extended UBL invoice

Following the deployment strategy for extending an existing document (page 214)

- `publd-zip/samp/custext/xsd/common/MyExtensions.xsd` is a specification of the apex of all extensions
- no real equivalent in the UBL structures, but probably closest to a document element
- both standard and extension definitions of ABIE and BBIE constructs are allowed

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:x-company:myExtensions"
            targetNamespace="urn:x-company:myExtensions"
            xmlns:mycac="urn:x-company:myExtensionsAggregateComponents"
            xmlns:mycbc="urn:x-company:myExtensionsBasicComponents"
            xmlns:cac="urn:oasis:...:CommonAggregateComponents-2"
            xmlns:cbc="urn:oasis:...:CommonBasicComponents-2"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
 <xsd:import namespace="urn:x-company:myExtensionsAggregateComponents"
             schemaLocation="MyAggregates.xsd"/>
 <xsd:import namespace="urn:x-company:myExtensionsBasicComponents"
             schemaLocation="MyBasics.xsd"/>
 <!--define an element in any namespace other than own namespace-->
 <xsd:group name="other">
   <xsd:sequence>
     <xsd:any namespace="##other" processContents="skip"/>
   </xsd:sequence>
 </xsd:group>
 <!--define the apex of own extension-->
 <xsd:element name="start">
   <xsd:complexType>
     <xsd:sequence>
       <!--any standard or extended construct is allowed here-->
       <!--at this time only invoice lines are extended-->
       <xsd:element ref="mycac:InvoiceLine" maxOccurs="unbounded"/>
     </xsd:sequence>
   </xsd:complexType>
 </xsd:element>
</xsd:schema>
```


Copyright © Crane Softwrights Ltd.

Extended aggregate components definition

Chapter 13 - Customization extension
Section 3 - Sample extended UBL invoice

Following the deployment strategy for extending an existing document (page 214)

- `publd-zip/samp/custext/xsd/common/MyAggregates.xsd` defines extension ABIE and ASBIE constructs
- may include UBL standard ASBIE and BBIE constructs

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:x-company:myExtensionsAggregateComponents"

targetNamespace="urn:x-company:myExtensionsAggregateComponents"
            xmlns:mycbc="urn:x-company:myExtensionsBasicComponents"
            xmlns:cac="urn:oasis:...:CommonAggregateComponents-2"
            xmlns:cbc="urn:oasis:...:CommonBasicComponents-2"
            xmlns:ccts="urn:un:unece:uncefact:documentation:2"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
  <xsd:import namespace="urn:x-company:myExtensionsBasicComponents"
              schemaLocation="MyBasics.xsd"/>
  <xsd:import namespace="urn:oasis:...:CommonBasicComponents-2"
              schemaLocation="UBL-CommonBasicComponents-2.0.xsd"/>
  <xsd:element name="InvoiceLine" type="InvoiceLineExtensionType"/>
  <xsd:complexType name="InvoiceLineExtensionType">
    ...
    <xsd:sequence>
      <xsd:element ref="cbc:ID" minOccurs="1" maxOccurs="1">
        ...
      </xsd:element>
     <xsd:element ref="mycbc:NewFacetAmount" minOccurs="1" maxOccurs="1">
        ...
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```


Copyright © Crane Softwrights Ltd.

Extended basic components definition

Following the deployment strategy for extending an existing document (page 214)

- `publd-zip/samp/custext/xsd/common/MyBasics.xsd` defines extension BBIE constructs
- every construct should be based on the UN/CEFACT unqualified data types

```
01 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
02              xmlns="urn:x-company:myExtensionsBasicComponents"
03              targetNamespace="urn:x-company:myExtensionsBasicComponents"
04              xmlns:udt="urn:un:...:UnqualifiedDataTypesSchemaModule:2"
05              elementFormDefault="qualified"
06              attributeFormDefault="unqualified">
07   <xsd:import namespace="urn:un:...:UnqualifiedDataTypesSchemaModule:2"
08               schemaLocation="UnqualifiedDataTypeSchemaModule-2.0.xsd"/>
09   <!--define the basic constructs of my extension-->
10   <xsd:element name="NewFacetAmount" type="udt:AmountType"/>
11 </xsd:schema>
```

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Chapter 14 - Customization deployment

- Introduction - Customization deployment
- Section 1 - Deployment interfaces
- Section 2 - Creating UBL instances
- Section 3 - Other deployment considerations

Outcomes

- understand the options when creating documents

Copyright © Crane Softwrights Ltd.

Customization deployment

Deployment is more than just document formats

- supporting the user community will help the deployment be successful
- different artefacts will support developers and users
- e.g. the government of Denmark offers an online validation service users can use to check the invoice instance they plan to submit for payment

What interchange protocols need to be supported?

- perhaps a standardized message handling specification?
    - use of ebXML (page 20)?
    - use of WS-*?
        - `http://www.w3.org/TR/ws-arch/`
- perhaps a custom Service Oriented Architecture (SOA)?
    - use of XML Remote Procedure Call (XML-RPC)?
        - `http://www.xmlrpc.com/spec`
    - use of Simple Object Access Protocol (SOAP)?
        - `http://www.w3.org/TR/soap/`
    - use of Representational State Transfer (REST)?
        - `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`

What application interfaces will you need for XML?

- use of Java Architecture for XML Binding (JAXB)?
    - `http://java.sun.com/developer/technicalArticles/WebServices/jaxb/`
- use of CodeSynthesis XSD/e for C++?
    - `http://www.codesynthesis.com/products/xsde/`
    - an open source development project, not an open standard
- use of the Post Schema Validation Infoset (PSVI)?
    - `http://www.w3.org/TR/xmlschema-1/#d0e504`
- use of Document Object Model (DOM)?
    - `http://www.w3.org/DOM/`
- use of Simple API for XML (SAX)?
    - `http://www.saxproject.org/`

Copyright © Crane Softwrights Ltd.

Customization deployment (cont.)

Identification of parties

- what to use for an end point identifier?
    - tax registration number?
        - could be used by a government as already managed by the government
    - GS1 Global Locator Number (GLN/EAN)?
        - a number used to identify an organization as a legal entity
    - other international register or infrastructure number?

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Interfacing with the outside world

Message handling

- responding to a synchronous request
    - e.g. an email message and its reply
- a "pull model" where the recipient goes looking for messages that may have been sent

Service Oriented Architectures

- responding to an asynchronous request
    - e.g. listening to a TCP/IP port and responding to an arbitrary access
- a "push model" where the recipient waits for a message to arrive

Use of encryption

- useful for authentication
    - confirming the sender is who they claim to be
- useful for integrity
    - confirming the content hasn't been tampered with
- useful for non-repudiation
    - proving the sender was the actual sending party
- will XML Advanced Electronic Signature XAdES be used?
    - `http://www.w3.org/TR/XAdES/`
    - the UBL committee is working on publishing guidelines for XAdES use with UBL documents

Copyright © Crane Softwrights Ltd.

Unmarshalling XML

- the verb is cited from the JAXB specification
- moving the information from the XML syntax into the application data structures
- is validation engaged?
    - the post-schema validation infoset (PSVI) interprets the text fields of XML as data-typed values
    - e.g. consider `<Quantity>123</Quantity>`
        - the XML syntax represents the text string "123"
        - if the associated schema for Quantity indicates the string value represents a number, then the PSVI delivers the number "123." to the application, not the text string "123"

Marshalling XML

- moving the information from the application data structures to XML syntax
- when using the Simple API for XML (SAX) the "`XMLGenerator`" companion library is often available
- nuanced rules regarding serialization need to be followed
    - use of some kind of library is recommended as it follows (should follow) the rules
- is validation available to check the results?
- is second-pass value validation available to check the results?

Copyright © Crane Softwrights Ltd.

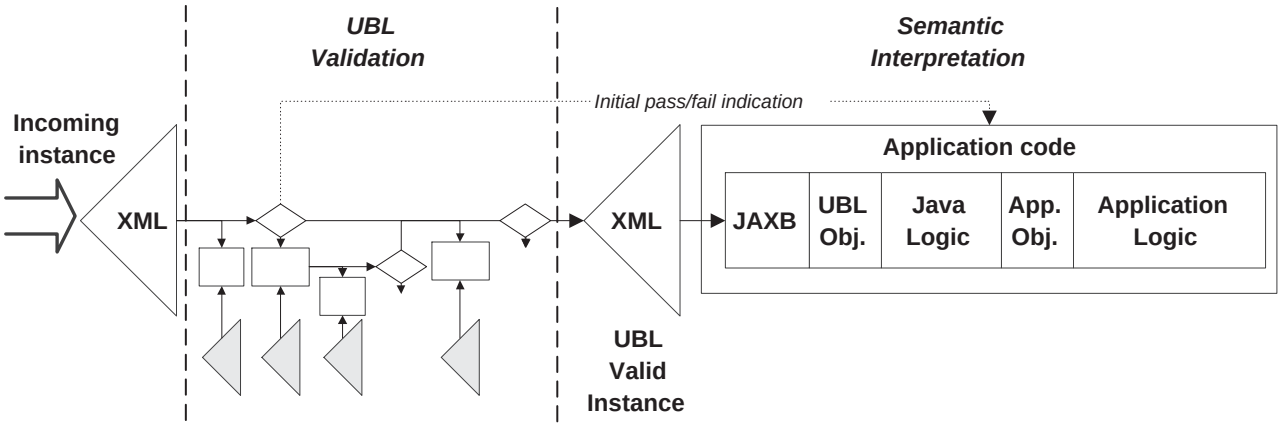
Interfacing to the application (cont.)

An example deployment incorporating XSLT and JAXB

- the application data structures should be oriented to the application and not to UBL
- opportunity to "filter out" anticipated UBL constructs defined by the customization that are of no interest to the application

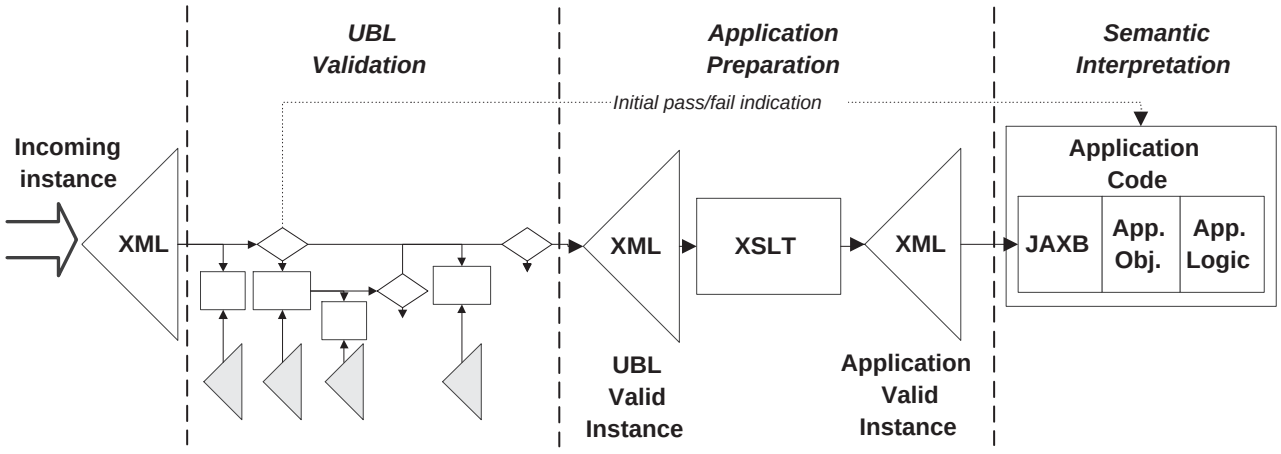
Supporting the application logic with application object data structures

- Java code is used to move information from UBL object data structures to application object data structures



Using XSLT in the application

- some developers find it awkward to maintain the Java code moving information from UBL object data structures to application object data structures
- maintaining the transformation in XSLT makes sense as XSLT is designed for XML



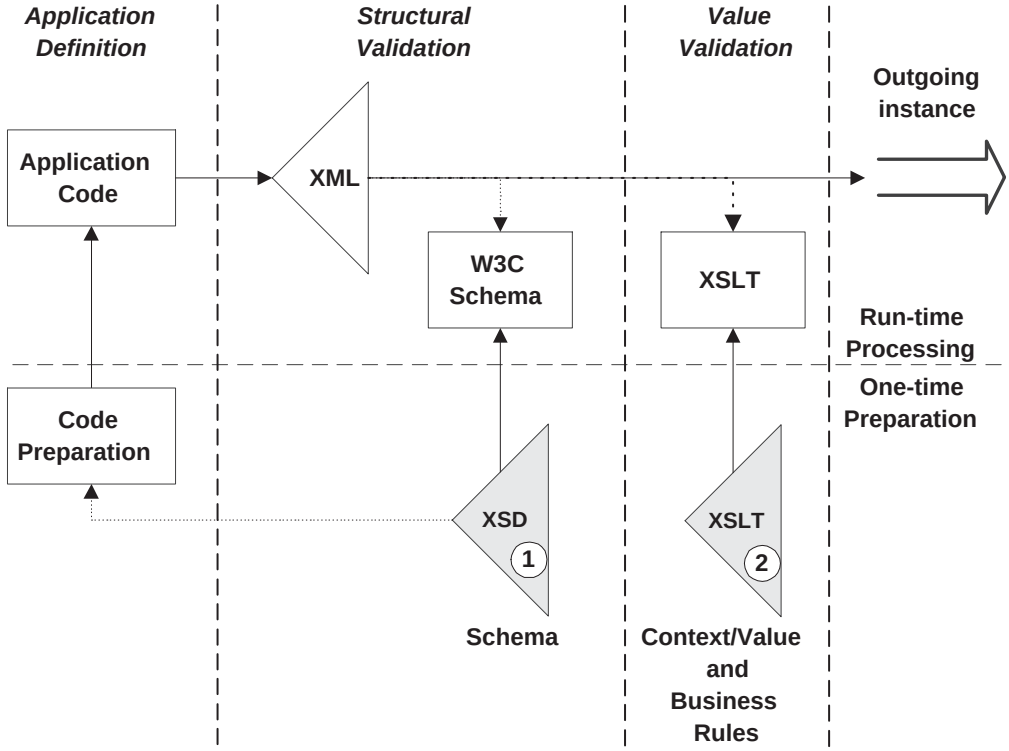
Important note regarding XML structures and XML syntax

- interfaces between programs in the data flow allow XML information to flow between programs as SAX events or DOM trees or other abstractions
- no need for instantiating XML syntax and parsing out the content again at each step

Copyright © Crane Softwrights Ltd.

Code preparation informed by customization schema

- some application development environments provide interfaces to XML documents
  - typically an XSD expression creates software components used to access the XML syntax



The XML instance must be well-formed and follow UBL document constraints

- recall UBL documents (page 101)
  - satisfy the constraints of a UBL W3C schema
  - use UTF-8 character encoding
  - no empty elements (except for `UBLExtension`)
  - no `xsi:schemaLocation`

Option to confirm the result of instance creation does not violate any document constraints

- structural validation using W3C Schema
- value validation using XSLT
- mimicking what the recipient is expected to do will ensure the recipient will not balk at the instance

Copyright © Crane Softwrights Ltd.

Other customization artefacts

Those defining a customization can make a number of resources available to support the user community

- documentation of the customization properties
    - e.g. spreadsheets and schemas
- variant business scenarios
    - e.g. profiles and variant document models
- code lists and business rules
    - e.g. genericode files, context/value association files and Schematron scripts
- rules for generating content
    - e.g. calculation models
- documentation for guidance
    - e.g. implementation guides
- samples for testing
    - e.g. sample XML instances
- tools with which to create instances
    - e.g. XForms input
- tools with which to validate that instances conform to the customization
    - e.g. filters, scripts and programs
    - e.g. online validation service
- tools with which to present instances of a customization
    - e.g. stylesheets
- communication tools for message handling
    - e.g. open source software for service oriented architectures (SOA)

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Annex A - OpenOffice 3 UBL customization environment

- Section 1 - OpenOffice 3 UBL customization environment

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

OpenOffice 3 UBL customization environment

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Crane Softwrights Ltd.'s `profile2ods` package is a pair of OpenOffice 3 XML filters

- enables OpenOffice 3 to specify and save a subset of the published UBL 2.0 document models
- supports the export of developer resources supporting a subset specification

Developer resources:

- human-readable HTML reports for consistency review
    - see Crane's UBL information model reports (page 84) for an example
    - all of Crane's model reports are created with this tool using 32 profiles
        - one profile includes all document types
        - one profile for each document type
- pruned W3C Schema XSD files
    - see Expressing a conformant UBL subset (page 181) for the role these play
- synthesized XSLT and Python instance filters
    - see Refining the document processing model (page 190) for the role these play
- XML and text document context XPath reports
    - see Chapter 7 - XPath enumerations (page 129) for the roles these play

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

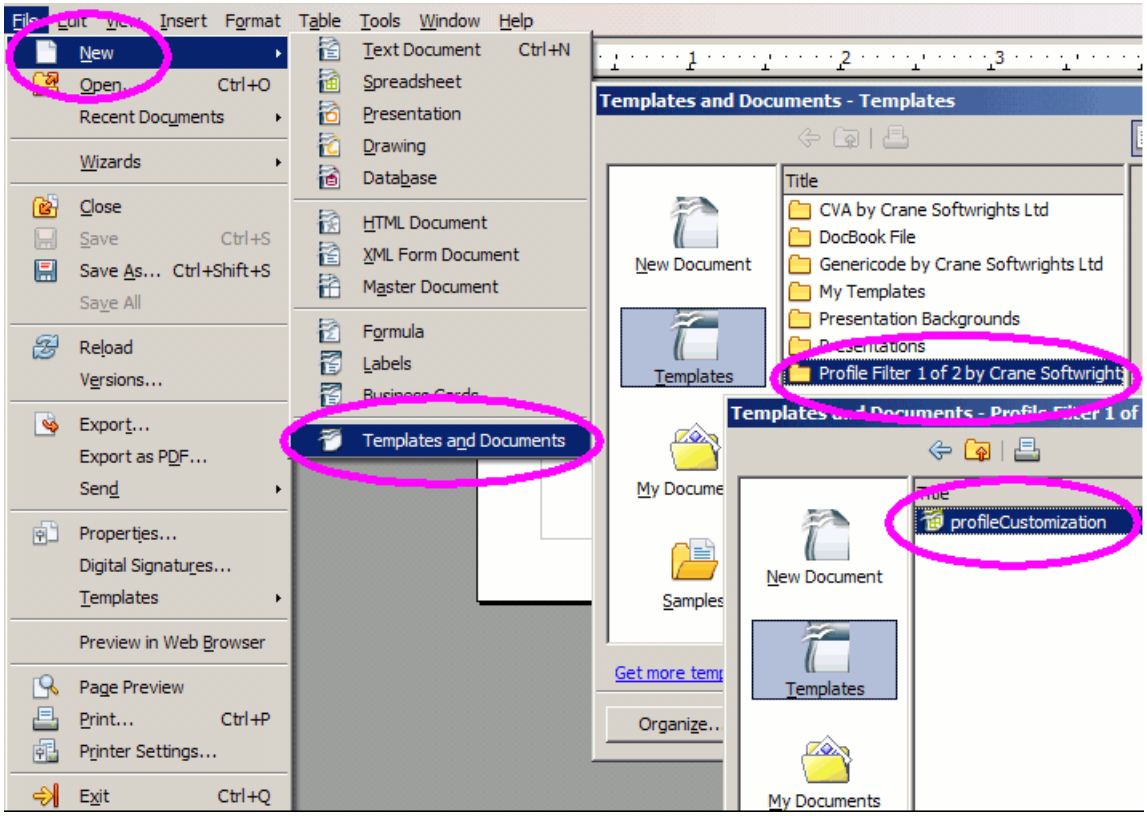
- `http://www.CraneSoftwrights.com/sales/publd/`
- your book purchase password is needed to get access to the package

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

Copyright © Crane Softwrights Ltd.

OpenOffice 3 UBL customization environment (cont.)

A new profile document is created from scratch using the `profileCustomization` template in "filter 1":



Copyright © Crane Softwrights Ltd.

OpenOffice 3 UBL customization environment (cont.)

There are thirty-seven sheets when editing the profiles:

- Configuration
  - managing the exportation and user interface properties
  - specifying the artefacts to be created during the exportation process
- Profiles
  - managing the meta data for each profile
- Summary document types
  - managing which document types are in which profiles
- Individual document types (32)
  - managing which components are in each document type for each profile
- Help
  - context-sensitive help information
- Support
  - there is no user serviceable data in this sheet

The customization is saved as an OpenOffice ODS file for maintenance

- faster operation than using the XML format
- preserves all information, including user-defined meta data

Creating the artefacts is triggered by exporting the document using the "Profile Export Filter"

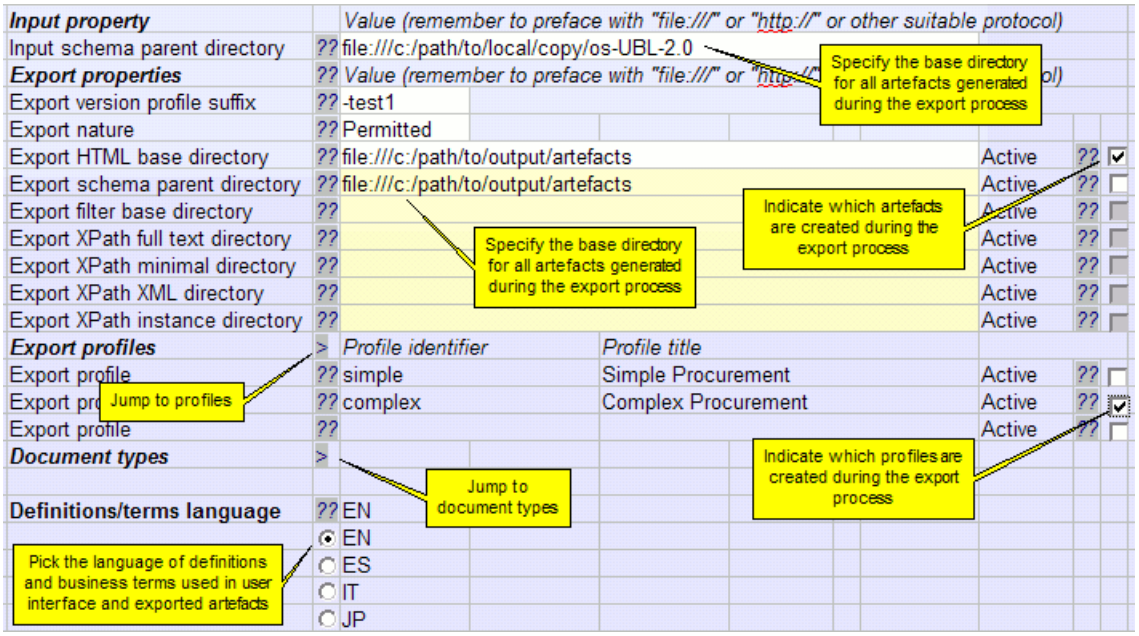
- artefact creation governed by settings on the configuration page

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Configuration sheet

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Used to configure the various export activities



- location of input schema files to be pruned
- nature of outputs
  - "permitted" subsets include all items with explicitly specified cardinality
  - "strict" subsets include only items with explicitly specified cardinality
- location of output artefact directories
- indications of export activity for each output artefact directory
- specification of language for definitions and business terms

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Configuration sheet (cont.)

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Use `File / Export...` to emit the selected artefacts

- the primary output file is a text file reporting the results of exporting multiple UBL artefacts

Example export report from the hands-on exercise:

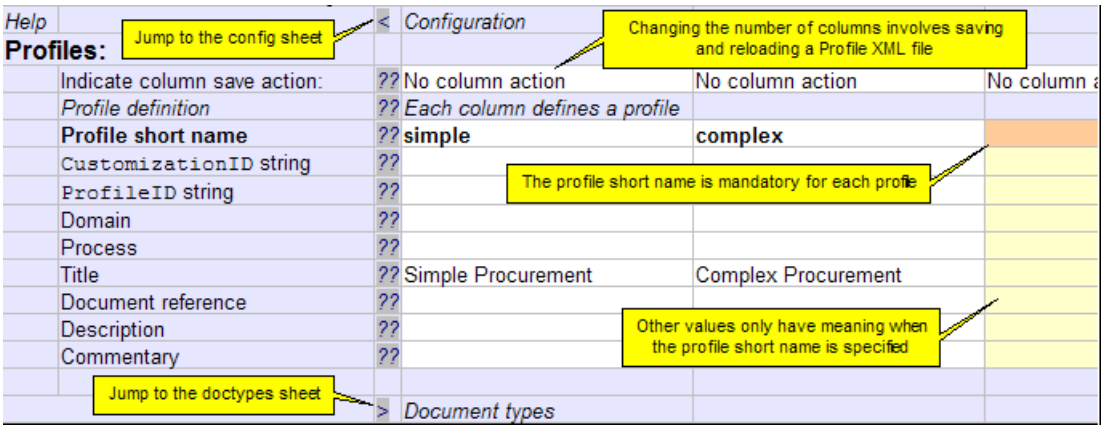
```
Crane's OASIS UBL 2.0 profile editor report output
--------------------------------------------------

Run time: 2009-02-11 20:50z
Determining requirements...
Analyzing project information...
Export version suffix:
Export nature: Strict
Producing results for profiles: Exercise

Profile short name: Exercise
Profile title: Exercise - Customization exercise
Models in profile: CommonLibrary

HTML production:
 Output HTML report file: file:///c:/publd/exer/Exercise/Exercise.html
Schema production: not active
Filter production: not active
XPath full text production: not active
XPath minimal text production: not active
XPath XML production: not active
XPath Instance production: not active

End of report.
```


Copyright © Crane Softwrights Ltd.

Profiles sheet

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Used to define properties of profiles



- column action is only engaged when files are saved and opened as XML
  - very long tasks to save and open the files as XML
  - user meta data is not preserved when using the XML format for saving the information
  - much more efficient to save and open the files in ODS when not needing to change the number of profiles
- the short name property is an identifier used for uniqueness across all profiles
  - participates in file naming conventions during export
- all other properties are documentary and do not impact on the specification
  - defining the title is useful for distinguishing the titles of reports from various configurations

Example values drawn from BII documents for illustration

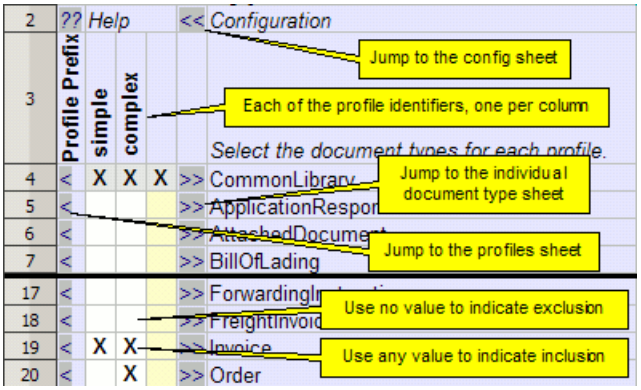
- Domain:
  - Post award procurement
- Process:
  - Ordering - Fulfillment - Billing - Payment
- Title:
  - Advanced Procurement with Dispatch
- Document reference:
  - CEN/ISSS WS/Profile BII13
- Description:
  - Profile description for Advanced Procurement with Dispatch
- Commentary:
  - e.g. status information, participants, discussion, etc.

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Document types sheet

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Used to specify the inclusion of document types in each profile



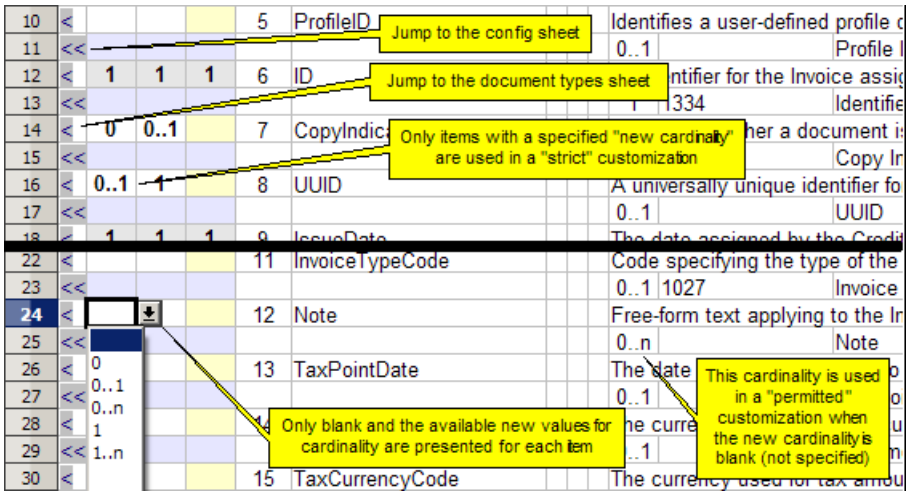
- any non-blank value can be used (the programmatic default is "X")
- data entry in white cells is used during the export process
- data entry in yellow cells is ignored during the export process and the XML save format
- defining a profile's short name enables white cells on this sheet
- specifying a document type on this document types sheet enables the corresponding white cells on the corresponding document type sheet

Copyright © Crane Softwrights Ltd.

Document type sheet

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Used to specify cardinalities of items and item-level user-defined meta data in each profile



- each new cardinality cell is constrained based on the model cardinality
- items with unspecified new cardinalities are included in permitted models but not in strict models
- data entry in white cells is used during the export process
- data entry in yellow cells is ignored during the export process and the XML save format

User-defined meta data can be defined on each sheet

- instructions for modification of the cell widths and headings are in the `readme.html` file

Copyright © Crane Softwrights Ltd.

Profile tool methodology

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Objective is to configure and support a customization of UBL

- what UBL standard constructs are needed?
- what UBL standard optional constructs can be discarded?
- what extensions need to be added?

The OpenOffice environment is for collaboration and specification

- export human-readable HTML report for review
    - are all of the constructs pruned as required?
    - are any constructs over-pruned to the point of being impossible to be UBL valid?
- when ready for testing or deployment, export other artefacts
    - schemas for validation of the instance against the subset specification
    - filters for pruning incoming UBL instances to test conformance against the subset
    - text and markup reports as developer tools for stylesheets and other applications acting on subset instances

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Profile tool methodology (cont.)

Annex A - OpenOffice 3 UBL customization environment
Section 1 - OpenOffice 3 UBL customization environment

Extension components and new document types are crafted by other tools

- the profile tool's sole purpose us to prune the published UBL 2.0 document models
- using other means create the replacement "Extension Content Datatype" module and new document schemas

Customization validation environment created from old and new components

- copy all schema fragments from UBL 2.0 update package
- overlay fragments replacing complete schemas with the pruned schemas
- overlay fragment replacing extension data type module with customization extension
- add customization extension support fragments
- see page 212 for the schema replacements

Customization validation processing model

- use a filter configured to only pass those constructs allowed for the customization
- see page 120 for the processing model
    - the version filter (F) is replaced with the customization filter

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Where to go from here?

Conclusion - Practical Universal Business Language Deployment

The work on UBL continues:

- OASIS UBL 2.0 Standard - December 12, 2006 with update May 26, 2008
    - `http://docs.oasis-open.org/ubl/os-UBL-2.0/`
    - `http://docs.oasis-open.org/ubl/os-UBL-2.0-update/`
- focus now shifts to support, deployment, awareness and evangelism
- join the UBL Technical Committee to help
    - `http://www.oasis-open.org/join/`
- committee mail list - UBL TC
    - `http://lists.oasis-open.org/archives/ubl/`
- community mail list - UBL-Dev
    - `http://lists.oasis-open.org/archives/ubl-dev/`
    - `http://www.oasis-open.org/mlmanage/`
- community resource center - UBL focus area
    - `http://ubl.xml.org`
- community contribution to UBL International Data Dictionary
    - `http://ubl.xml.org/forums/ubl-international-data-dictionary-idd-contributions`

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Colophon

Conclusion - Practical Universal Business Language Deployment

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

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Obtaining a copy of this material

Conclusion - Practical Universal Business Language Deployment

This comprehensive tutorial on UBL is available for subscription purchase and free preview download:

- "Practical Universal Business Language Deployment" Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
    - 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-20090212.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/publd/` 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`

Copyright © Crane Softwrights Ltd.

Practical Universal Business Language Deployment

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

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Table of contents

Indexed by slide number

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8

Copyright © Crane Softwrights Ltd.

Copyright © Crane Softwrights Ltd.

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Index

Third Edition - 2009-02-12 - ISBN 978-1-894049-23-8
Copyright © Crane Softwrights Ltd.

Copyright © Crane Softwrights Ltd.

Copyright © Crane Softwrights Ltd.

Y

Z

Copyright © Crane Softwrights Ltd.