Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

ISO/IEC 19757-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 34, Document Description and Processing Languages.

ISO/IEC 19757 consists of the following parts, under the general title *Document Schema Definition Languages (DSDL)*:

– *Part 1: Overview*

– *Part 2: Regular-grammar-based validation — RELAX NG*

– *Part 3: Rule-based validation — Schematron*

– *Part 4: Namespace-based validation dispatching language — NVDL*

– *Part 5: Datatypes*

– *Part 6: Path-based integrity constraints*

– *Part 7: Character repertoire description language — CRDL*

– *Part 8: Document schema renaming language — DSRL*

– *Part 9: Datatype- and namespace-aware DTDs*

– *Part 10: Validation management*

Introduction

This International Standard defines a set of Document Schema Definition Languages (DSDL) that can be used to specify one or more validation processes performed against Extensible Markup Language (XML) or Standard Generalized Markup Language (SGML) documents. (XML is an application profile of SGML — ISO 8879:1986.)

A document model is an expression of the constraints to be placed on the structure and content of documents to be validated against the model and the information set that needs to be transmitted to subsequent processes. Since the development of Document Type Definitions (DTDs) as part of ISO 8879, a number of technologies have been developed through various formal and informal consortia notably by the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). A number of validation technologies are standardized in DSDL to complement those already available as standards or from industry.

Historically, when many applications act on a single document, each application inefficiently duplicates the task of confirming that validation requirements have been met. Furthermore, such tasks and expressions have been developed and utilized in isolation, without consideration of how the features and functionality available in other technologies might enhance validation objectives.

The main objective of this International Standard is to bring together different validation-related tasks and expressions to form a single extensible framework that allows technologies to work in series or in parallel to produce a single or a set of validation results. The extensibility of DSDL accommodates validation technologies not yet designed or specified.

In the past, different design and use criteria have led users to choose different validation technologies for different portions of their information. Bringing together information within a single XML document sometimes prevents existing document models from being used to validate sections of data. By providing an integrated suite of constraint description languages that can be applied to different subsets of a single XML document, this International Standard allows different validation technologies to be integrated under a well-defined validation policy.

This multi-part International Standard integrates constraint description technologies into a suite that:

– provides user control of names, order and repeatability of information objects and their properties (elements and their attributes)

– allows users to identify restrictions on the coexistence of information objects

– allows specific information object within structured documents to be validated

– allows restrictions to be placed on the contents of specific elements and attributes, including restrictions based on the content of other elements in the same document

– allows the character set that can be used within specific elements to be managed, based on the application of the ISO/IEC 10646 Universal Multiple-Octet Coded Character Set (UCS)

– allows default values to be assigned to element contents and attribute values, and provides facilities for the incorporation of predefined fragments of structured data to be incorporated within documents

– extends SGML DTDs to include functions such as namespace-controlled validation and datatypes by adapting XML techniques for these capabilities to SGML.

Document Schema Definition Languages (DSDL) – Part 1: Overview

1 Scope

This International Standard specifies a suite of technologies that can be used to validate the structure and contents of structured documents marked up using ISO 8879 (SGML) and its derivatives (e.g. the W3C Extensible Markup Language, XML).

This International Stadnard defines a set of semantics for describing and ordering validation rules, a set of syntaxes for declaring validation rules, and a syntax for defining models for the management of validation sequences. It includes:

– Specifications of relevant validation technologies that can be used in isolation or within the DSDL framework.

– References to validation technologies defined outside of this International Standard that can be used within the DSDL framework.

– Semantics for managing the sequence in which different validation technologies are to be applied during the production of validation results.

This International Standard identifies specifications that can be used by a data validator that accepts a structured input document and produces one or more validation results. This International Standard does not standardize how these specifications shall be invoked, or the error messages they produce.

Documents that are not conformant with ISO 8879 (SGML) or one of its derivatives are not within the field of application of this International Standard. Documents prepared using SGML must be validated against an SGML DTD as the first stage in the validation process to produce a well-formed output that is conformant with the W3C XML information set.

All intermediate and final expressions of information used for DSDL processing must be expressible using the XML Information Set, except where specific extensions are defined within this standard. The information set may be generated from external sources such as the ESIS of SGML. No expression of any concept supported by DSDL shall require anything beyond which can be expressed in an XML document.

This International Standard has the following parts, whose role is explained in the following clauses of this overview:

– *Part 1: Overview*

– *Part 2: Regular-grammar-based validation — RELAX NG*

– *Part 3: Rule-based validation —* Schematron

– *Part 4: Namespace-based validation dispatching language — NVDL*

– *Part 5: Datatypes*

– *Part 6: Path-based integrity constraints*

– *Part 7: Character repertoire description language — CRDL*

– *Part 8: Document schema renaming language — DSRL*

– *Part 9: Datatype and namespace-aware DTDs*

– *Part 10: Validation management*

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 19757. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 19757 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*, Internet Standards Track Specification, August 1998, http://www.ietf.org/rfc/rfc2396.txt

SGML, *Standard Generalized Markup Language (SGML)*, ISO 8879:1986,

UCS, *Universal Multiple-Octet Coded Character Set (UCS)*, ISO/IEC 10646:2000,

W3C XML, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation, 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006

W3C XML-Infoset, *XML Information Set*, W3C Recommendation, 24 October 2001, http://www.w3.org/TR/2001/REC-xml-infoset-20011024/

W3C XML-Names, *Namespaces in XML*, W3C Recommendation, 14 January 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114/

W3C XPath, *XML Path Language (XPath) Version 1.0*, W3C Recommendation, 16 November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116

W3C XML Schema, *XML Schema*, W3C Recommendation, 24 October 2001, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

3 Terms and definitions

3.1 compact syntax

set of declarative rules expressed in a syntax other than XML against which one or more instances can be validated

3.2 DTD

Document Type Declaration: set of rules formally declared in a document type (DOCTYPE) declaration against which one or more instances can be validated

NOTE 1: In ISO 8879 the term DTD stands for Document Type Definition, of which a DOCTYPE is a partial expression.

3.3 (document) instance

a structured document that is being validated with respect to a DSDL expression of document model constraints for structure and content

3.4 schema

XML-encoded set of declarative rules against which one or more instances can be validated

4 The role of document schemas

Document schemas provide machine-readable models that can be used to validate the structure and contents of electronically marked-up documents. The Document Type Declaration (DTD) language defined within ISO 8879 provides facilities for:

– defining the names used to identify document elements

– identifying where document elements may appear in the document structure (model)

– identifying which elements were optional or repeatable (without limiting repeatability)

– identifying which markup tags are optional when they can be inferred through the model

– assigning properties (attributes) to document elements that can be used to control their processing, or can contain information that needs to be processed in conjunction with element contents

– defining default values for attributes

– defining and naming repeatable segments of text (entities)

– identifying non-standard characters using user-assigned names or character numbers

– linking together different document structures defined in parallel sets of markup.

Document structures are discussed in ISO 8879 in terms of *trees* of nested elements, though the standard also allows data sets to be defined as *graphs* of elements connected by means of unique identifiers and references to existing identifiers.

DTDs are defined in a compact syntax, and hence do not use the same descriptive components as document instances. In SGML this notation can be preceded by an SGML declaration that specifies information such as permitted character sets[1], which characters are assigned as control functions or otherwise ignored (shunned), which characters can be used within names, or to identify the boundaries of markup, which strings can be used to automatically identify markup points, and which optional functions are to be used within the DTD.

The W3C Extensible Markup Language (XML) uses an application profile of ISO 8879, known as WebSGML, together with the ISO/IEC 10646 Universal Multiple-Octet Coded Character Set, to produce a simple-to-implement, streamable application of ISO 8879 for use over the Internet and similar networks. XML document models can be defined using DTDs that conform to a well-defined set of restrictions on the options of ISO 8879 so that vaildation can be performed within streamed networks.

Various organizations have developed techniques to manage the structure of documents using XML markup. Some of these further subset the facilities provided in ISO 8879, but many also provide functions over and above those allowed within SGML and XML DTDs, including:

– control of the minimum and maximum number of times an element can occur at a particular point in the document structure (e.g. to control cardinality)

– restriction of the contents of particular elements or attributes to those that conform to particular datatypes, patterns or internally defined lists of permitted elements

– provision for distinguishing the namespaces of element and attribute names so that schema fragments can be used within other schemas without fear of name clashes

– identification of elements based on the path needed to reach them within the document structure

– validation of document structures by checking that elements conforming to particular paths exist

– provision of mechanisms for creating abstract stereotypes (similar to SGML architectural forms) that can be used to identify related classes of elements.

[1] The character definition rules predate the development of the ISO/IEC 10646 Universal Multiple-Octet Coded Character Set and are to some extent made redundant by this standard.

5 Other user requirements

Users have requested the following additional functionality:

– The ability to control the character set permitted within the contents of a particular type of element or attribute, or within specified sets of elements within the document model.

– The ability to restrict the range of entries conforming to a particular use of a datatype within a specific element or attribute.

– The ability to restrict element or attribute contents to values specified in either internally defined or externally defined lists of permitted values.

– The ability to restrict the set of permitted values in one element or attribute based on the contents of another element or attribute (e.g. not Sex=Male and Diagnosis=Pregnant).

– The ability to generate compact forms of schemas that are easily readable by humans, and to use such compact representations to generate schemas or DTDs that can be used to validate documents.

– The ability to visualize schemas using navigable diagrammatic representations.

6 Validation management

The various parts of this International Standard are designed to satisfy a declaration of validation requirements, without the need to use processes in a predefined sequence. Some parts of the standard, however, need to be applied before others. For example, validation of the contents of a specific element requires prior identification of element boundaries and nesting, while the validation of the relationship between elements may require that the document structure be validated first so that the paths specified can be checked.

Consider the example in Figure 1 where two different results can be determined from two different applications of technology to the validation process: validating after or before processing XInclude. The order of these two steps may be critical in the correct processing of the information in the instance.

In a more complex example, consider Figure 2 where two different technologies must be applied to separate portions of one document. In this case, one part of the input document must be validated by a W3C XML Schema while the other part of the input document must be validated by a RELAX-NG schema. The validation result, choreographed by DSDL, expresses the consolidated validation of all steps.

7 Path-based addressing

Non-hierarchical links between information items in a structured resource can be identified by addressing items within the document tree and then expressing relationships among them. The addressing mechanism enables hierarchy-based paths of steps along a tree's branches to an information item.

Paths-based constraints are based on:

– a method for identifying information items dependent on:
  – the ancestry of the information item
  – the use of keys (e.g. references to unique identifier values)
  – a mechanism that can be referenced through DSDL's extensiblity features.
– a method for describing relationships that are not hierarchical.

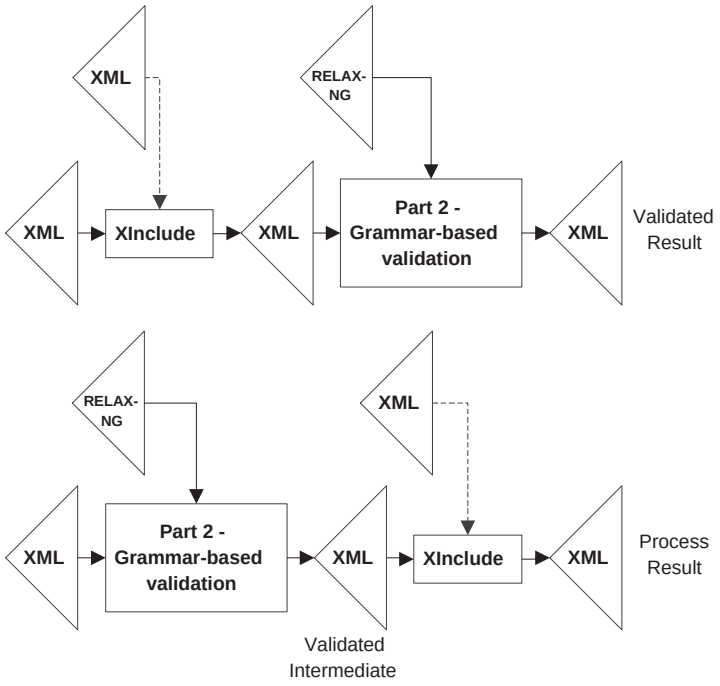

Figure 1: Two different orders of application of technology

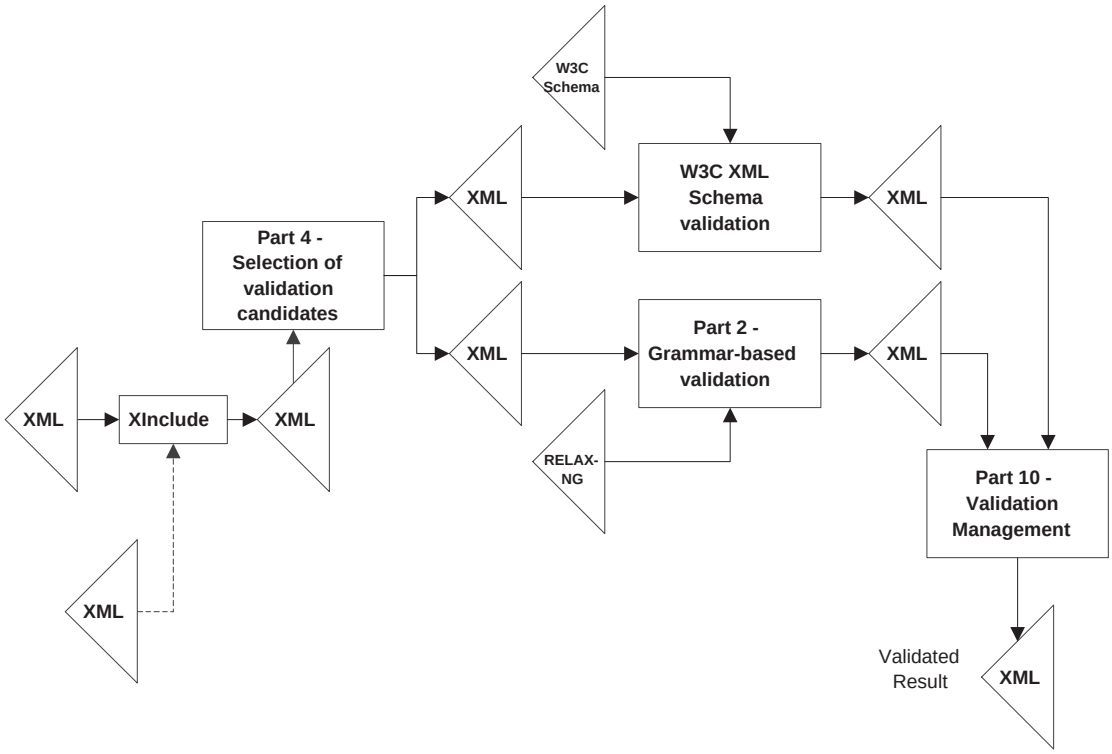

Figure 2: A multi-step validation process

A number of Parts within this International Standard utilize the concept of a path to address components of document instances.

Paths are used in Parts 3, 6 and 10.

Should path identification be described in a separate part referenced by the others? Would it belong in Part 10 with scope over all the others?

8 Overview of the parts

8.1 Part 1: Overview

This part of the standard introduces the role of each of the other parts of the standard, and identifies the user requirements that the standard addresses.

8.2 Part 2: Regular-grammar-based Validation

Regular-grammar-based schema languages can validate that the structure and content of information items in a document instance conforms to a model described by a tree grammar.

Tree grammars are characterized by the specification of node patterns. Validation is based on the matching of elements identified in the stream being analyzed with one of the pattern definitions permitted at a particular point in a data tree.

The regular-grammar-based language defined in this Part is based on the OASIS RELAX NG specification. The RELAX-NG grammar includes a syntax for specifying:

– which elements can make up a data hierarchy, and the rules for ordering these elements

– which attributes can be assigned to an element

– the identity of datatypes and their permitted ranges of values

– which datatypes should be used to validate the contents of a particular element or attribute.

RELAX-NG provides a generalized mechanism for the identification of datatypes, without defining how a particular datatype can be validated. In addition to being able to use the datatype definitions provided by Part 5, other datatype definitions, in particular W3C XML Schema Data types, can be utilized as part of the validation process.

A compact syntax has been defined for RELAX NG that allows grammars to be prepared in a highly readable format in a manner that allows existing grammars to be easily and quickly extended to meet new needs.

In the evolution of other grammar-oriented schema languages may be defined in the evolution of DSDL as separate parts of this International Standard.

8.3 Part 3: Rule-based Validation

Rule-based schema languages allow documents to be validated by confirming that they do not conflict with a set of rules describing permitted relationships between document components. Rules do not need to be based on hierarchical relationships, but can use hierarchical relationships to identify applicable parts of data streams.

Rules are required to allow the specification of constraints such as 'If the contents of the element named "Sex" is "Male" then the contents of the element "Diagnosis" may not include "Pregnant".' Rules can also ensure that sets of data are compatible, e.g. "If there are multiple items in an order for which different delivery dates have been specified, ensure that all delivery dates are between the order date and the date specificed as the maximum permitted time for completion of the order."

The rule-based grammar defined in this part is based on the widely adopted Schematron specification. It provides a syntax for specifying:

– a set of variables used when comparing or calculating tests

– assertions to be tested

– the contexts in which one or more assertions are to be tested

– abstract patterns which can be matched by different elements in different contexts

– sets of rules that are applied sequentially so that only the first matching rule in the pattern is applied

– what is to be reported when an assertion is not verified, optionally accompanied by diagnostics showing how to correct errors that are encountered

– phases of validation, in which sequences of rules may be applied depending on the phase specificed when the validation schema is invoked

– keys that allow components to be linked during subsequent processing.

In the evolution of DSDL other rule-based validation languages may be defined as separate parts of this International Standard.

8.4 Part 4: Namespace-based Validation Dispatching Language — NVDL

Part 4 provides an XML-based language for selecting elements and attributes in specific namespaces within a document instance that are to be validated by a specified schema. Such elements are known as validation candidates. This part also enables sets of validation rules to be shared between applications that share data components.

Schema languages other than DSDL (for example RDF Schema[1] and the TMCL[2]) may be used for validating selected validation candidates. For example, an XHTML document containing metadata expressed in RDF can be decomposed into an XHTML validation candidate and a metadata validation candidate, which can be validated independently.

Selection methods include:

– namespace-based element selection, where schema selection is controlled by the namespaces of elements that are to be validated against different schemas

– namespace-based attribute selection, where attribute value datatype validation is controlled by the namespace of attributes.

The namespace-based validation dispatching language defined in this Part is derived from the Namespace Rule Langauge (NRL) specification.

We currently provide no way of selecting particular elements in a specific namespace for validation of fragments.

Should we provide a mechanism for processing specific fragments, and must this restrict validation in embedded elements to elements or attributes which have other namespaces?

It is outside the scope of this part to specify which schema and schema language shall be used for validating validation candidates.

8.5 Part 5: Datatypes

This Part defines a syntax for:

– creating named sets of user-defined datatypes (e.g. `ISBN-number`)

– parsing element and attribute contents using regular expression grammars to identify component parts which need to be validated

– defining sets of permitted (enumerated) values that can be used to validate the contents of a specific element or attribute

– defining sets of repeatable, choice or optional items which can make up a datatype definition

– identifying constraints that can be used to limit a particular datatype (e.g. minimum and maximum values)

– defining conditions that must be met if a datatype is to be considered valid

– identifying relationships between datatypes (e.g. supertype/subtype relationships) and techniques for mapping from one datatype to another

– defining collation orders for datatypes, or identifying externally defined collation rules

8.6 Part 6: Path-based Integrity Constraints

Path-based integrity constraints allow path-based languages, such as the XML Path Language (XPath), to be used to identify relationships between elements that must, or may not, occur in valid documents.

This Part is based on the four-directional tree path navigation paradigm (parent, child, preceding sibling and following sibling) defined in XPath.

Path-based constraints can be used to identify a fragment of a document against which a specific schema may be applied. For example, if a document has adopted the CALS table model without assigning a specific namespace to CALS-conformant elements, this Part can be used to select a table and parse it according a schema fragment that defines the structure of CALS tables.

Path-based constraints will also permit the specification of inter-document relationships, allowing the validity of one document to depend on the presence of specific information in another document.

Should Part 6 also contain a mechanism for selecting fragments from other documents that are to be included at a particular point in a document instance (e.g. to provide boiler plate text or to provide a default value when no specific entry exists in a document instance?

What else should path-based integrity constraints provide us with? What do these integrity constraints do that cannot be done using paths defined within Part 3?

Paths are used in both Parts 3 and 6. Should paths be described in a separate part referenced by the others?

Is a normative reference to XML Paths 2.0 sufficient (or should we just specify 1.0 plus those extensions we wish to support in DSDL)? Should we put constraints on which parts of XPath 2.0 can be used within DSDL? Could other forms of path specification be required at a future date?

If we do not have a separate Part 6 how do we enable the validation of fragments of a document?

8.7 Part 7: Character Repertoire Declaration Language — CRDL

At present SGML and XML users have no control over which set of characters can appear in a particular element or attribute value. For example, an element could have an `xml:lang` attribute indicating it is in English but contain Chinese or Sanskrit characters. This Part provides a mechanism for checking that the contents of an element or attribute are taken from a formally defined subset of the ISO/IEC 10646 Universal Multiple-Octet Coded Character Set (UCS) that is the basis for XML encoded documents.

This Part provides a syntax for:

– defining named subsets of the ISO/IEC 10646 character set

– identifying which named character set shall be used to validate the content of a specific element or attribute.

CRDL utilizes the hull and kernel approach to character set definition whereby a minimal set of "compulsory characters" (the kernel) can be supplemented by characters in a wider set of "optional characters" (the hull) from which certain "eceptions" have been excluded.

8.8 Part 8: Document Schema Renaming Language — DSRL

The Document Schema Renaming Language provides mechanisms whereby locally meaningful names can be associated with element, attribute and entity names used within schemas, and for the definition of locally meaningful "templates" of reusable data.

DSRL templates can be used to assign default values to specific parts of a data stream. This includes mechanisms for defining standard sequences of data that can be incorporated into document instances by reference to an identifying name, the provision of default content for elements and attributes for which no value is provided, and the matching of local element and attribute names to those used in a specific schema.

DSRL defines a syntax for describing simple modifications to be made to the information set of a DSDL document instance, without requiring the full power of a general-purpose transformation language such as XSLT[3].

This Part provides a syntax for:

– identifying elements, attributes and entities whose names have been altered from those required by the validation schema

– assigning a default value to the contents of a specific type of element or attribute for which no value is specified in the document instance

– defining named entities which can be refernced at points at which users wish to incorporate predefined sets of data elements (a template) within a document instance

– removing elements or attributes from specific locations within the document model.

8.9 Part 9: Namespace and Datatype-aware DTDs

This Part specifies how the ISO 8879 and XML Document Type Declaration (DTD) syntaxes can be extended to validate documents that make use of XML Namespaces and Datatypes. Doing so will ensure that the investment that individuals and organizations have made in DTD development and deployment over many years can be preserved. It also simplifies conversion between DTDs and other forms of schema languages.

The specification does not require documents using the schema language to violate XML's well-formedness or validity checks. It simply identifies processing instructions whose role can be considered to be that of specifying additional validation rules to be applied to specific elements or attributes.

8.10 Part 10: Validation Management

This Part provides a language for orchestrating the validation and pre-validation transformation processes described in the other parts of this standard.

Validation management includes:

– a mechanism to invoke parsers which read non-XML sources (and XML sources that can't be identified by a single URI) to create XML infosets that can be used for subsequent processing. Examples of such sources include SGML and HTML documents, RDBMS query results, CSV documents and Web Services query results

– pre-validation transformations used to normalize and/or subset documents before validation

– multiple validations and transformations that are to be applied to the same document

– transformations that split a document into multiple resulting documents

– facilities to generate customized validation reports which can be output as XML document instances so that they can be further processed by other applications.

This part also illustrates how technologies other than those specified in the parts of this International Standard, such as the W3C XML Schema and XSLT transformation language, can be used in combination to manage XML and other forms of structured documentation.

Bibliography

[1] *RDF Vocabulary Description Language 1.0: RDF Schema*, http://www.w3.org/TR/rdf-schema

[2] *Topic Map Constraint Language*, http://www.w3.org/TR/rdf-schema

[3] *XSL Transformations (XSLT) Version 1.0*, http://www.w3.org/TR/xslt

Summary of editorial comments:

[7] Path-based addressing

Paths are used in Parts 3, 6 and 10.

Should path identification be described in a separate part referenced by the others? Would it belong in Part 10 with scope over all the others?

[8.4] Part 4: Namespace-based Validation Dispatching Language — NVDL

We currently provide no way of selecting particular elements in a specific namespace for validation of fragments.

Should we provide a mechanism for processing specific fragments, and must this restrict validation in embedded elements to elements or attributes which have other namespaces?

[8.6] Part 6: Path-based Integrity Constraints

Should Part 6 also contain a mechanism for selecting fragments from other documents that are to be included at a particular point in a document instance (e.g. to provide boiler plate text or to provide a default value when no specific entry exists in a document instance?

What else should path-based integrity constraints provide us with? What do these integrity constraints do that cannot be done using paths defined within Part 3?

Paths are used in both Parts 3 and 6. Should paths be described in a separate part referenced by the others?

Is a normative reference to XML Paths 2.0 sufficient (or should we just specify 1.0 plus those extensions we wish to support in DSDL)? Should we put constraints on which parts of XPath 2.0 can be used within DSDL? Could other forms of path specification be required at a future date?

If we do not have a separate Part 6 how do we enable the validation of fragments of a document?