

Contents

	Page
Foreword.....	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	2
3 Terms and definitions.....	2
4 The role of document schemas.....	2
5 Other user requirements.....	3
6 Validation management.....	4
7 Path-based addressing.....	4
8 Overview of the parts.....	4
8.1 Part 1: Overview.....	4
8.2 Part 2: Regular-grammar-based Validation.....	5
8.3 Part 3: Rule-based Validation.....	6
8.4 Part 4: Selection of Validation Candidates.....	7
8.5 Part 5: Datatypes.....	7
8.6 Part 6: Path-based Integrity Constraints.....	7
9 Part 7: Character Repertoire Validation.....	8
10 Part 8: Declarative Document Architectures.....	8
11 Part 9: Namespace and Datatype-aware DTDs.....	9
12 Part 10: Validation Management.....	9
Bibliography.....	10

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: Selection of validation candidates*
- *Part 5: Datatypes*
- *Part 6: Path-based integrity constraints*
- *Part 7: Character repertoire validation*
- *Part 8: Declarative document manipulation*
- *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 Stylesheet Language (XML) or Standard Generalized Markup Language (SGML) documents. XML is an application profile 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 with the model. A number of technologies have been developed through various formal and informal consortia since the development of Document Type Definitions (DTDs) as part of ISO 8879, 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.

To validate that a structured document conforms to specified constraints in structure and content relieves the potentially many applications acting on the document from having to duplicate the task of confirming that such requirements have been met. Historically, such tasks and expressions have been developed and utilized in isolation, without consideration for how the features and functionality available in other technologies might enhance validation objectives.

The main objective of this International Standard is to bring together varied validation-related tasks and expressions to form a single extensible framework that allows the 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 standard integrates constraint description technologies into a suite that:

- provides user control of names, order and repeatability of information objects (elements)
- allows users to identify restrictions on the co-concurrence of elements and element contents
- allows specific subsets of structured documents to be validated
- allows restrictions to be placed on the contents of specific elements, 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
- allows SGML to be used to declare document structure constraints that extend DTDs to include functions such as namespace-controlled validation and datatypes.

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).

The Document Schema Definition Language (DSDL) 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 that 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.

DSDL identifies specifications to 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 in 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 specification.

All intermediate and final expressions of information used for DSDL processing are restricted to the XML Information Set and to XML documents, though these can 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 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*
- *Part 3: Rule-based Validation*
- *Part 4: Selection of Validation Candidates*
- *Part 5: Datatypes*
- *Part 6: Path-based Integrity Constraints*
- *Part 7: Character Repertoire Validation*
- *Part 8: Declarative Document Architectures*
- *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

4 The role of document schemas

Document schemas provide machine-understandable models that can be used to validate the structure and contents of electronically marked-up documents. The Document Type Definition (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 defined 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 not defined using the same descriptive components as marked-up document instances, being defined using a sparser notation. In SGML this notation can be preceded by a declaration of permitted character sets, 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 DTD¹.

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 validation 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 namespace 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.

5 Other user requirements

The following additional functionality has been identified as being required by users:

- 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.

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.

- The ability to visualize schemas using navigable diagrammatic representations.

Question: What other entries need to be added to this list?

6 Validation management

The various parts of the DSDL standard are designed to be choreographed to satisfy a declaration of validation requirements, without the need to use processes in a predefined sequence. Some parts of the standard will, however, need to be applied before others. For example, validation of the contents of a specific element will require 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 accordingly.

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 the 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, orchestrated by DSDL, expresses the consolidated validation of all steps.

7 Path-based addressing

The non-hierarchical links between information items in a structured resource can be identified by addressing the items within the document tree and then expressing the relationship between them. The addressing mechanism includes hierarchy-based paths of steps along the tree's branches to the information item being addressed.

Paths can be based on:

- a method of identifying information items dependent on:
 - the ancestry of the information item
 - the use of keys (e.g. references to unique identifier values)
 - an extensible basis for supporting mechanisms not currently available
- a method of describing the role of relationships that are not hierarchical.

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

Paths are used in both parts 3 and 6.

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.

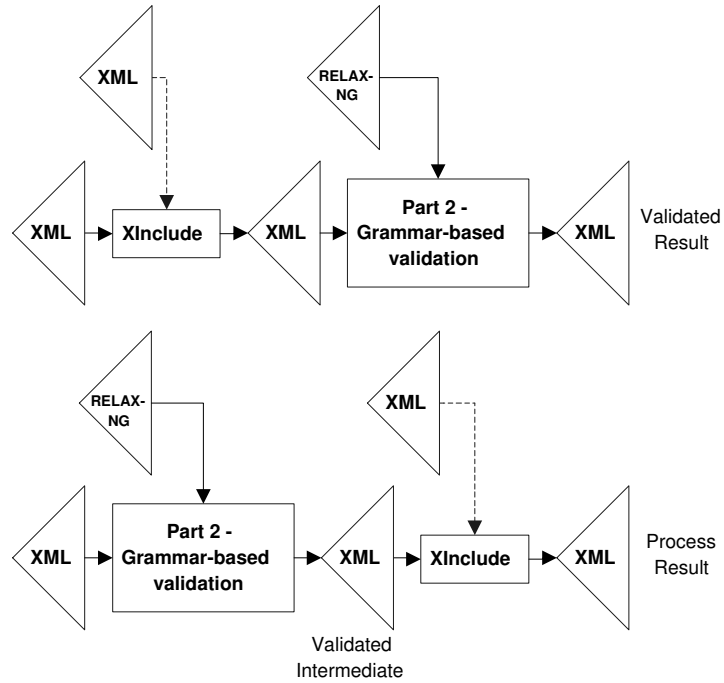


Figure 1: Two different orders of application of technology

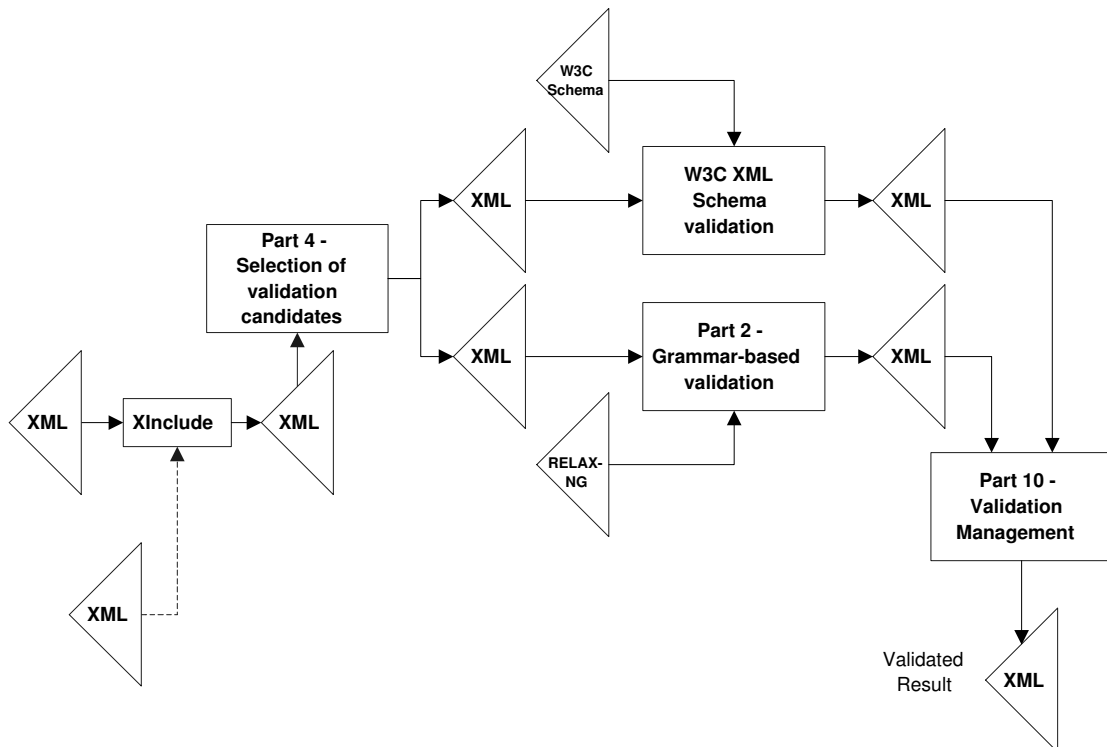


Figure 2: A multi-step validation process

8.2 Part 2: Regular-grammar-based Validation

Regular-grammar-based schema languages can be used to 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 of this standard other datatype definitions, in particular W3C XML Schema Data types, can be utilized as part of the validation process.

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 which parts of data streams they should be applied to.

Rules are required to allow indirect constraints such as 'if the contents of the element named "Sex" is "Male" then the contents of the element "Diagnosis" may not include "Pregnant"' to be specified. Rules can also be used to 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 specified as the maximum permitted time for completion of the order".

The rule-based grammar defined in this Part is an extension of the Schematron specification. It provides a syntax for specifying:

- a set of variables to be used when comparing or calculating tests
- assertions that need to be tested
- the context in which one or more assertions are required to be tested
- abstract patterns, which can be matched by different elements in different contexts
- sets of rules that are to be 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 specified when the validation schema is invoked
- keys that allow components to be linked during subsequent processing.

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

8.4 Part 4: Selection of Validation Candidates

The selection of validation candidates is required to allow different sets of validation rules to be applied to selected parts of data streams. It also enables sets of validation rules to be shared between applications that share data components.

This Part provides an XML-based language for selecting specific elements within a document instance that are to be validated by a specified schema. Such elements are called validation candidates.

This Part can be used to identify elements that are embedded within an existing XML document which may be used to create an independent XML document.

Selection methods include:

- namespace-based selection, which is controlled by conditions on namespaces of elements, defined using the rules in Parts 2 or 9
- partial tree selection, whose subtrees can be validated separately
- attribute-based selection, where validation is controlled by the values of attributes.

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.

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 minimal set of standardized named primitive datatypes (e.g. *integer*)
- a set of parameters (controlling facets) and their permitted values for each primitive datatype (e.g. minimum and maximum values)
- a set of constraints describing a possibly infinite set of strings representing values of the data type
- a set of commonly required derived datatypes that provide subsets, or combinations, of primitive datatypes
- techniques for creating application defined datatypes that are combinations of one or more existing datatypes.

This Part has been developed from the set of primitive datatypes and their facets defined in Part 2 of the W3C XML Schema specification.

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. It allows:

- the identification of paths that must exist in the document if a particular element, attribute or subtree exists within the document
- the expression of identity and integrity constraints on components of document instances identified through the use of path expressions.

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 the normative reference to XML Paths 2.0 sufficient? Could other forms of path specification be required at a future date?

9 Part 7: Character Repertoire Validation

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.

10 Part 8: Declarative Document Architectures

Declarative document architectures allow default values to be assigned 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.

This Part 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:

- using XPath to address elements and attributes to be modified
- assigning a default value to the contents of a specific type of element or attribute
- defining named entities of predefined data elements that can be used to include template data within a document instance
- renaming elements and attributes in specific locations within the document model, including the assignment of element or attribute names to different namespaces
- the definition of replacement contents for specific elements or attributes
- removing elements or attributes from specific locations within the document model.

Under consideration for this part are Architectural Form and Architecture Support Attribute approaches that will provide an XML representation of architectural forms of the type defined in the AFDR specification in Annex A.3 of ISO 10744.

11 Part 9: Namespace and Datatype-aware DTDs

This Part specifies how the ISO 8879/XML Document Type Definition (DTD) syntax can be extended to validate documents that make full use of XML Namespaces and Part 5 Datatypes. This will ensure that the investment that individuals and organizations have made in DTD development and deployment over many years can be preserved. It will also help those converting 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 attribute names whose role can be considered to be that of specifying additional validation rules to be applied to specific elements or attributes.

12 Part 10: Validation Management

This Part provides a language for orchestrating the validation and pre-validation transformation processes described in the other parts of the 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 may 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 other parts of this standards, 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:

[5] Other user requirements

Question: What other entries need to be added to this list?

[7] Path-based addressing

Paths are used in both parts 3 and 6.

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?

[8.6] Part 6: Path-based Integrity Constraints

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 the normative reference to XML Paths 2.0 sufficient? Could other forms of path specification be required at a future date?

[10] Part 8: Declarative Document Architectures

Under consideration for this part are Architectural Form and Architecture Support Attribute approaches that will provide an XML representation of architectural forms of the type defined in the AFDR specification in Annex A.3 of ISO 10744.