INTERNATIONAL STANDARD

ISO/IEC 19757-3

First edition
2006-06-01

Information technology — Document Schema Definition Languages (DSDL) —

Part 3: Rule-based validation — Schematron

Technologies de l'information — Langages de définition de schéma de documents (DSDL) —

Partie 3: Validation de règles orientées — Schematron

Reference number
ISO/IEC 19757-3:2006(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

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 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19757-3 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 *Information technology — 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*

The following parts are under preparation:

— *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

ISO/IEC 19757 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 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 (DTD) 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 of how the features and functionality available in other technologies might enhance validation objectives.

The main objective of ISO/IEC 19757 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, ISO/IEC 19757 allows different validation technologies to be integrated under a well-defined validation policy.

This part of ISO/IEC 19757 is based on the Schematron[1] assertion language. The `let` element is based on XCSL[2]. Other features arise from the half-dozen early Open Source implementations of Schematron in diverse programming languages and from discussions in electronic forums by Schematron users and implementers.

The structure of this part of ISO/IEC 19757 is as follows. Clause 5 describes the syntax of an ISO Schematron schema. Clause 6 describes the semantics of a correct ISO Schematron schema; the semantics specify when a document is valid with respect to an ISO Schematron schema. Clause 7 describes conformance requirements for implementations of ISO Schematron validators. Annex A is a normative annex providing the ISO/IEC 19757-2 (RELAX NG) schema for ISO Schematron. Annex B is a normative annex providing the ISO Schematron schema for constraints in ISO Schematron that cannot be expressed by the schema of Annex A. Annex C is a normative annex providing the default query language binding to XSLT. Annex D is an informative annex providing a ISO/IEC 19757-2 (RELAX NG compact syntax) schema and corresponding ISO Schematron schema for a simple XML language Schematron Validation Report Language. Annex E is an informative annex providing motivating design requirements for ISO Schematron. Annex F is a normative annex allowing certain Schematron elements to be used in external vocabularies. Annex G is an informative annex with a simple example of a multi-lingual schema.

Considered as a document type, a Schematron schema contains natural-language assertions concerning a set of documents, marked up with various elements and attributes for testing these natural-language assertions, and for simplifying and grouping assertions.

Considered theoretically, a Schematron schema reduces to a non-chaining rule system whose terms are Boolean functions invoking an external query language on the instance and other visible XML documents, with syntactic features to reduce specification size and to allow efficient implementation.

Considered analytically, Schematron has two characteristic high-level abstractions: the pattern and the phase. These allow the representation of non-regular, non-sequential constraints that ISO/IEC 19757-2 cannot specify, and various dynamic or contingent constraints.

Information technology — Document Schema Definition Languages (DSDL) —

Part 3:
Rule-based validation — Schematron

1 Scope

This part of ISO/IEC 19757 specifies Schematron, a schema language for XML. This part of ISO/IEC 19757 establishes requirements for Schematron schemas and specifies when an XML document matches the patterns specified by a Schematron schema.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE Each of the following documents has a unique identifier that is used to cite the document in the text. The unique identifier consists of the part of the reference up to the first comma.

W3C XML 1.0, *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, 04 February 2004

XPath, *XML Path Language (XPath) Version 1.0*, W3C Recommendation, 16 November 1999

XSLT, *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation, 16 November 1999

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1 abstract pattern

pattern in a rule that has been parameterized to enable reuse

3.2 abstract rule

collection of assertions which can be included in other rules but which does not fire itself

3.3 active pattern

pattern belonging to the active phase

3.4 active phase

one particular phase, whose patterns are used for validation

3.5 assertion

natural-language assertion with corresponding assertion test and ancillary attributes: assertions are marked up with `assert` and `report` elements

3.6 assertion test

assertion modelled or implemented by a Boolean query; an assertion test "succeeds" or "fails"

3.7 correct schema

schema that satisfies all the requirements of this part of ISO/IEC 19757

3.8 diagnostic

named natural language statements providing information to end-users of validators concerning the expected and actual values together with repair hints

3.9 elaborated rule context expression

single rule context expression which explicitly disallows items selected by lexically previous rule contexts in the same pattern

3.10 good schema

correct schema with queries which terminate and do not add constraints to those of the natural-language assertions.

NOTE It may not be possible to compute that a schema is good.

3.11 implementation

implementation of a Schematron validator

3.12 name

token with no whitespace characters

3.13 natural-language assertion

natural-language statement expressing some part of a pattern; a natural-language assertion is "met" or "unmet"

3.14 pattern

named structure in instances specified in a schema by a lexically-ordered collection of rules

3.15 phase

named, unordered collection of patterns; patterns may belong to more than one phase; two names, `#ALL` and `#DEFAULT`, are reserved with particular meanings

3.16 progressive validation

validation of constraints in stages determined or grouped to some extent by the schema author rather than, for example, entirely determined by document order

3.17 query language binding

named set, specified in a document called a Query Language Binding, of the languages and conventions used for assertion tests, rule-context expressions and so on, by a particular Schematron implementation

NOTE 1 Schematron is defined as a framework, with a default query language binding, but other query language bindings are possible.

NOTE 2 6.4 specifies the information to be required by a query language binding and Annex C defines the default query language binding for Schematron.

3.18 rule

unordered collection of assertions with a rule-context expression and ancillary attributes

3.19 rule context

element or other information item used for assertion tests; a rule is said to fire when an information item matches the rule context

3.20 rule-context expression

a query to specify subjects; a rule-context is said to match an information item when that information item has not been matched by any lexically-previous rule context expressions in the same pattern and the information item is one of the information items that the query would specify

3.21 schema

specification of a set of XML documents

3.22 subject

particular information item which corresponds to the object of interest of the natural-language assertions and typically is matched by the context expression of a rule

3.23 valid with respect to a schema

member of the set of XML documents described by the schema: an instance document is valid if no assertion tests in fired rules of active patterns fail

3.24 variable

constant value, evaluated within the parent schema, phase, pattern or rule and scoped within the parent schema, phase, pattern or rule

4 Notation

4.1 XPath

This part of ISO/IEC 19757 uses XPath to identify information items in Schematron schemas.

4.2 Predicate Logic

This part of ISO/IEC 19757 uses predicate logic to express the semantics of Schematron schema. The following symbols are defined for use in s6.3:

— ()
Grouping delimiters

— ∀
"for all". Prefix operator.

— ¬
"not". Prefix operator.

— ∈
"is member of", in set operative sense. Prefix operator.

— ,
"and" (sequence). Infix operator.

— :
"where". Such that. Infix operator.

— ⇔
"if and only if". Infix operator.

5 Syntax

5.1 Well-formedness

A Schematron schema shall be a well-formed XML document, according to the version of XML used.

5.2 Namespace

All elements shown in the grammar for Schematron are qualified with the namespace URI:

`http://purl.oclc.org/dsdl/schematron`

In subsequent clauses, the prefix `sch` is taken as bound to the Schematron namespace URI for exposition purposes. The prefix `sch` is not reserved or required by this part of ISO/IEC 19757.

Any element can also have foreign attributes in addition to the attributes shown in the grammar. A foreign attribute is an attribute with a name whose namespace URI is neither the empty string nor the Schematron namespace URI. Any non-empty element may have foreign child elements in addition to the child elements shown in the grammar. A foreign element is an element with a name whose namespace URI is not the Schematron namespace URI. There are no constraints on the relative position of foreign child elements with respect to other child elements.

5.3 Whitespace

Any element can also have as children strings that consist entirely of whitespace characters, where a whitespace character is one of U+0020, U+0009, U+000D or U+000A. There are no constraints on the relative position of whitespace string children with respect to child elements.

NOTE Leading and trailing whitespace should be stripped from attributes defined by this part. Whitespace should be collapsed in elements defined by this part that allow text. Whitespace may be stripped from elements defined by this part that do not allow text.

5.4 Core Elements

The grammar for Schematron elements is given in Annex A.

5.4.1 `active` element

The required `pattern` attribute is a reference to a pattern that is active in the current phase.

5.4.2 `assert` element

An assertion made about the context nodes. The data content is a natural-language assertion. The required `test` attribute is an assertion test evaluated in the current context. If the test evaluates positive, the assertion succeeds. The optional `diagnostics` attribute is a reference to further diagnostic information.

The natural-language assertion shall be a positive statement of a constraint.

NOTE The natural-language assertion may contain information about actual values in addition to expected values and may contain diagnostic information. Users should note, however, that the `diagnostic` element is provided for such information to encourage clear statement of the natural-language assertion.

The `icon`, `see` and `fpi` attributes allow rich interfaces and documentation. They are defined below.

The `flag` attribute allows more detailed outcomes. It is defined below.

The `role` and `subject` attributes allow explicit identification of some part of a pattern. They are defined below.

5.4.3 `extends` element

Abstract rules are named lists of assertions without a context expression. The required `rule` attribute references an abstract rule. The current rule uses all the assertions from the abstract rule it extends.

5.4.4 `include` element

The required `href` attribute references an external well-formed XML document whose document element is a Schematron element of a type which is allowed by the grammar for Schematron at the current position in the schema. The external document is inserted in place of the `include` element.

5.4.5 `let` element

A declaration of a named variable. If the let element is the child of a `rule` element, the variable is calculated and scoped to the current rule and context. Otherwise, the variable is calculated with the context of the instance document root.

The required `name` attribute is the name of the variable. The required `value` attribute is an expression evaluated in the current context.

It is an error to reference a variable that has not been defined in the current schema, phase, pattern, or rule, if the query language binding allows this to be determined reliably. It is an error for a variable to be multiply defined in the current schema, phase, pattern and rule.

The variable is substituted into assertion tests and other expressions in the same rule before the test or expression is evaluated. The query language binding specifies which lexical conventions are used to detect references to variables.

An implementation may provide a facility to override the values of top-level variables specified by `let` elements under the `schema` element. For example, an implementation may allow top-level variables to be supplied on the command line. The values provided are strings or data objects, not expressions.

5.4.6 `name` element

Provides the names of nodes from the instance document to allow clearer assertions and diagnostics. The optional `path` attribute is an expression evaluated in the current context that returns a string that is the name of a node. In the latter case, the name of the node is used.

An implementation which does not report natural-language assertions is not required to make use of this element.

5.4.7 `ns` element

Specification of a namespace prefix and URI. The required `prefix` attribute is an XML name with no colon character. The required `uri` attribute is a namespace URI.

NOTE Because the characters allowed as names may change in versions of XML subsequent to W3C XML 1.0, the ISO/IEC 19757-2 (RELAX NG Compact Syntax) schema for Schematron does not constrain the prefix to particular characters.

In an ISO Schematron schema, namespace prefixes in context expressions, assertion tests and other query expressions should use the namespace bindings provided by this element. Namespace prefixes should not use the namespace bindings in scope for element and attribute names.

5.4.8 `param` element

A name-value pair providing parameters for an abstract pattern. The required `name` attribute is an XML name with no colon. The required `value` attribute is a fragment of a query.

5.4.9 `pattern` element

A structure, simple or complex. A set of rules giving constraints that are in some way related. The `id` attribute provides a unique name for the pattern and is required for abstract patterns.

The `title` and `p` elements allow rich documentation.

The `icon`, `see` and `fpi` attributes allow rich interfaces and documentation.

When a `pattern` element has the attribute `abstract` with a value `true`, then the pattern defines an abstract pattern. An abstract pattern shall not have a `is-a` attribute and shall have an `id` attribute.

Abstract patterns allow a common definition mechanism for structures which use different names and paths, but which are at heart the same. For example, there are different table markup languages, but they all can be in large part represented as an abstract pattern where a table contains rows and rows contain entries, as defined in the following example using the default query language binding:

```
<sch:pattern abstract="true" id="table">
    <sch:rule context="$table">
        <sch:assert test="$row">
        The element <name/> is a table. Tables contain rows.
        </sch:assert>
    </sch:rule>

    <sch:rule context="$row">
        <sch:assert test="$entry">
        The element <name/> is a table row. Rows contain entries.
        </sch:assert>
    </sch:rule>

</sch:pattern>
```


When a `pattern` element has the attribute `is-a` with a value specifying the name of an abstract pattern, then the pattern is an instance of an abstract pattern. Such a pattern shall not contain any `rule` elements, but shall have `param` elements for all parameters used in the abstract pattern.

The following example uses the abstract pattern for tables given above to create three patterns for tables with different names or structures.

```
<sch:pattern is-a="table" id="HTML_Table">
    <sch:param name="table" value="table"/>
    <sch:param name="row"   value="tr"/>
    <sch:param name="entry" value="td|th"/>
</sch:pattern>

<sch:pattern is-a="table" id="CALS_Table">
    <sch:param name="table" value="table"/>
    <sch:param name="row"   value=".//row"/>
    <sch:param name="entry" value="cell"/>
</sch:pattern>

<sch:pattern is-a="table" id="calendar">
```