

Contents

Page

Foreword.....	iv
Introduction.....	v
1 Scope	1
2 General approach	1
3 Syntax	1
4 Inclusion of namespace information in DTDs	2
5 Inclusion of datatype information in DTDs	2

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 19757-9 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 34, Document Description and Processing Languages.

- *Part 1: Interoperability framework*
- *Part 2: Grammar-based validation — RELAX NG*
- *Part 3: Rule-based validation — Schematron*
- *Part 4: Namespace-based validation dispatching language*
- *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*

Introduction

Document Schema Definition Languages (DSDL) – Part 9: Namespace- and datatype-aware DTDs

1 Scope

2 General approach

The defined approach involves the use of processing instructions, all of which take the following general syntactical form:

```
<?DSDL-9 DTD-extension-declaration ?>
```

where the string 'DTD-extension-declaration' varies according to the function of the processing instruction in question.

This approach is applied specifically to extend the functionality of DTDs to support the use of namespaces and datatypes.

3 Syntax

The following is a summary of the syntax of the processing instructions defined by this part of ISO/IEC 19757.

```
DTD-extension-processing-instruction ::= pio "DSDL-9" s (target-namespace-declaration |
namespace-prefix-binding-declaration | wildcard-namespace-qualifier-declaration | default-datatype-library-declaration
| datatype-library-prefix-binding-declaration | datatype-declaration) pic
```

```
target-namespace-declaration ::= "target-namespace=" URI-literal
```

```
namespace-prefix-binding-declaration ::= "namespace-prefix-binding=" URI-prefix-pair s "applies-to=" name-locator-literal
```

```
wildcard-namespace-qualifier-declaration ::= "wildcard-namespace-qualifier=" (other-namespace | URI-literal) s
"applies-to=" name-locator-literal
```

```
default-datatype-library-declaration ::= "default-datatype-library=" URI-literal
```

```
datatype-library-prefix-binding-declaration ::= "datatype-library-prefix-binding=" URI-prefix-pair
```

```
datatype-declaration ::= "datatype=" (datatype-name | datatype-qname) s "applies-to=" name-locator-literal
```

```
URI-literal ::= ((lit URI lit) | (lita URI lita))
```

```
URI-prefix-pair ::= ((lit URI s prefix lit) | (lita URI s prefix lita))
```

```
other-namespace ::= ((lit "#OTHER" lit) | (lita "#OTHER" lita))
```

```
name-locator-literal ::= ((lit name-locator-sequence lit) | (lita name-locator lita))
```

```
name-locator-sequence ::= name-locator (s name-locator)*
```

```
name-locator ::= "element" s (name-token-group | "**") ( s "attribute" s (name-token-group | "**"))?
```

```
datatype-qname ::= ((lit prefix ":" datatype-name lit) | (lita prefix ":" datatype-name lita))
```

```
prefix ::= NCName
```

```
datatype-name ::= NCName
```

4 Inclusion of namespace information in DTDs

A target namespace may be associated with the DTD as a whole, in which case all unqualified names are in the target namespace.

A namespace URI may be associated with specified element type name(s) or attribute name(s) by including a corresponding namespace prefix binding declaration in the DTD.

A namespace URI may be associated with the content of an element type whose content model is declared to be "ANY", by including a wildcard namespace qualifier declaration in the DTD. In this case any element that is in the qualifying namespace and is also declared in the DTD is valid in that context. The content of an element type with content model ANY may alternatively be more weakly constrained to be in any namespace other than the target namespace.

If an instance is to be DTD-valid, all element types and attributes must be declared in the DTD, regardless of the namespace to which their names belong.

All names that are in namespaces other than the target namespace must be qualified names.

A DTD may only contain one default namespace declaration.

A DTD may contain more than one namespace prefix binding declaration for any given prefix, provided the contexts of their use do not overlap.

5 Inclusion of datatype information in DTDs

A datatype may be associated with a specified element type name or attribute name, by including a corresponding datatype declaration in the DTD.

Every datatype name must be associated with a declared datatype library. If a specified datatype name includes a qualifying prefix, a corresponding datatype library prefix binding declaration must be included in the DTD. A DTD may not contain two datatype library prefix binding declarations for the same prefix.

If a datatype name is specified without a qualifying prefix, the DTD must include a default datatype library declaration. A DTD may only contain one default datatype library declaration.

The URI specified in either a datatype library prefix binding declaration or a default datatype library declaration must resolve to an XML instance that conforms to Part 5 of this International Standard (DTLL).

Summary of editorial comments: