Modularization of XHTML™ in XML Schema

W3C Working Draft - 22 March 2001

This version:
http://www.w3.org/TR/2001/WD-xhtml-m12n-schema-20010322
(Single HTML file [p.1] , PostScript version, PDF version, ZIP archive, or Gzip'd TAR archive)
Latest version:
http://www.w3.org/TR/xhtml-m12n-schema
Editors:
Daniel Austin, Mozquito Technologies AG
Shane McCarron, Applied Testing and Technology

Copyright ©2001 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This document describes a methodology for the modularization of XHTML using XML Schema. Modularization of XHTML allows document authors to modify and extend XHTML in a conformant way.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This is the first public "Working Draft" of "Modularization of XHTML in XML Schema" for review by members of the W3C and other interested parties in the general public. It is a stand-alone document to ease its review. Once the methodology described in this document become mature, it will be integrated into a future document forthcoming from the HTML Working Group.

This document is still in its early stage, and may be updated, replaced, or obsoleted by other documents at any time. Publication of this Working Draft does not imply endorsement by the W3C, and it is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Please send review comments to www-html-editor@w3.org (archive). Public discussion on HTML features takes place on the mailing list www-html@w3.org (archive).

This document has been produced by the W3C HTML Working Group (*members only*) as part of the HTML Activity. The goals of the HTML Working Group are discussed in the HTML Working Group charter. The W3C staff contact for work on HTML is Masayasu Ishikawa.

Quick Table of Contents

1. Introduction 5
2. Schema Modularization Framework 7
3. XHTML Schema Modules 21
A. References 39
B. Changes 43
C. Acknowledgements 45
D. XHTML Schema Module Implementations 47

Full Table of Contents

1. Introduction 5
1.1. Purpose of this document 5
1.2. Why Modularize? 5
1.3. Design Goals 6
1.4. Requirements 6
2. Schema Modularization Framework 7
2.1. How Schema Modularization Works 7
2.1.1. DTDs and XML Schema 7
2.1.2. Document Data Structures 7
2.1.3. Understanding XHTML Modularization 8
2.1.4. Mapping DTDs onto Schema 8
2.2. Framework Conventions 13
2.2.1. Modularized Schema Files 14
2.2.2. Module Naming 14
2.2.3. Module Hierarchy Structure 15
2.2.4. Names for Data Structures 16
2.2.5. Module Structure 17
2.2.6. Namespace Conventions 19
2.2.7. Documentation Conventions 19
3. XHTML Schema Modules 21
3.1. XHTML Abstract Modules 21
3.2. XHTML Schema Modules 21
3.2.1. Required Modules 21
3.2.2. Optional Modules 28

3.2.3. XHTML Hub Document (Non-normative) 36
3.3. Validity and Conformance 37
3.3.1. XHTML Conformance 37
3.3.2. Schema Modularization Conformance 38
3.3.3. The XHTML Family of Documents 38
3.3.4. Versioning 38
A. References 39
A.1. Normative References 39
B. Changes 43
B.1. Changes to Abstract Modules 43
B.2. Changes from DTD Module Implementations 43
C. Acknowledgements 45
D. XHTML Schema Module Implementations 47
D.1. XHTML Abstract Modules and XML Schema 47
D.2. XHTML Schema Modules 47
D.2.1. XHTML Hub Document 47
D.3. XHTML SCHEMA Modular Framework 50
D.3.1. XHTML Notations 52
D.3.2. XHTML Datatypes 54
D.3.3. XHTML Common Attribute Definitions 55
D.3.4. XHTML Character Entities 57
D.4. XHTML Module Implementations 58
D.4.1. XHTML Core Modules 58
D.4.2. Text Modules 64
D.4.3. Forms 67
D.4.4. Tables 74
D.4.5. Image 81
D.4.6. Client-side Image Map 82
D.4.7. Server-side Image Map 85
D.4.8. Applet 85
D.4.9. Object 87
D.4.10. Frames 88
D.4.11. Target 90
D.4.12. Iframe 92
D.4.13. Intrinsic Events 93
D.4.14. Metainformation 96
D.4.15. Scripting 97
D.4.16. Stylesheet 98
D.4.17. Style Attribute 99
D.4.18. Link 99
D.4.19. Base 101
D.4.20. Name Identification 102
D.4.21. Legacy 103

D.4.22. Ruby 104
D.5. XHTML Schema Support Modules 107
D.5.1. Block Phrasal 107
D.5.2. Block Presentational 108
D.5.3. Block Structural 109
D.5.4. Inline Phrasal 110
D.5.5. Inline Presentational 112
D.5.6. Inline Structural 113
D.5.7. Param 114
D.5.8. Miscellaneous Legacy 115
D.5.9. Legacy Frames 120
D.5.10. Optional Module Hub 121
D.5.11. Core Hub Module 127
D.5.12. XHTML 1.1 Content Model 129

1. Introduction

This section is *informative*.

1.1. Purpose of this document

The purpose of this document is to describe a modularization framework for languages within the XHTML Namespace using XML Schema [XMLSCHEMA] [p.41] . There are currently several public language variants in the XHTML namespace, including XHTML 1.0 [XHTML10] [p.40] (which includes variants corresponding to the definitions of "strict", "transitional", and "frameset") and XHTML Basic [XHTMLBASIC] [p.40] . The development of DTD-based modularization for XHTML made it possible to refashion XHTML 1.0 in a modularized way [XHTMLMOD] [p.41] , resulting in XHTML 1.1 [XHTML11] [p.40] . (Here and throughout this document, the term "XHTML-MOD" is used to refer to [XHTMLMOD] [p.41] .)

This document provides a complete set of XML Schema modules for XHTML. In addition to the schema modules themselves, the framework presented here describes a means of further extending and modifying XHTML.

To the largest extent possible, the modularization framework presented here attempts to duplicate the modularization concepts used in XHTML-MOD. Data structures in the modularized DTDs are in many cases mapped directly onto data structures in XML Schema. This method does not yet however, make extensive use of XML Schema-specific features.

This document is based on an approach to modular schemas originally suggested by Rick Jelliffe and members of the XML Schema Working Group at W3C. [APPROACH] [p.39]

1.2. Why Modularize?

In the development of any type of complex application, it is important to follow a clear conceptual standard for organizing the development. The modular approach to design reduces the application's functionality into some number of "building blocks" or "modules". These modules are then combined according to specific rules to form the entire application. This approach offers numerous advantages:

Conceptual clarity allows developers to share ideas and code

Reduces complexity by decomposition of the application's functionality

Supports object-oriented design principles by encouraging encapsulation and information hiding

Encourages reuse by creating well-defined modules that perform a particular task

Decreases debugging time by localizing errors due to design changes

Increases flexibility and maintainability because single modules can be upgraded or replaced independently of others

Eases development, testing, and maintenance by providing a logical, easy to understand, and consistent organization

Allows the **creation of generic rules, methods, and procedures** to aid in consistent development practices

Creates configurable objects that the end user can tailor for different purposes

Supports a variety of end user interface and deployment environments by allowing standardized subsets and supersets.

1.3. Design Goals

These are the design goals for this modularization framework for XHTML:

- To create coherent sets of semantically related modules within the XHTML namespace using XML Schema
- To support the creation of subsets and supersets of XHTML for specific purposes such as handheld devices and special-purpose appliances
- To facilitate future development by allowing modules to be upgraded or replaced independently of other modules
- To encourage and facilitate the reuse of common modules by developers.

1.4. Requirements

This document describes a modularization framework that attempts to reuse the conceptual ideas in XHTML-MOD, but does not attempt to literally duplicate them in all aspects.

The DTD modularization framework described in XHTML-MOD is subject to a detailed and explicit list of requirements [XHTMLMOD] [p.41] . The scope of the schema-based framework described here is also constrained by this set of requirements, and is believed to have fulfilled them in their entirety.

2. Schema Modularization Framework

This section is *informative*.

2.1. How Schema Modularization Works

2.1.1. DTDs and XML Schema

Both DTDs and XML Schema are designed to accomplish the same fundamental task: to define the structure of XML document types. In this sense both are simply different text representations for the same underlying data structures. However, Schema and DTDs differ significantly in several ways, both in structure and capabilities.

Some differences worth noting are:

Common XML features - XML Schema are XML documents themselves and therefore share many aspects of the languages they define.

Data typing - schemas are designed with a much larger set of built-in data types than DTDs, and provide methods for creating user-defined types.

Namespaces - DTDs only partially support XML Namespaces, which are inherently a part of XML Schema.

Extension - XML Schema have a rich set of extension mechanisms including inheritance, redefinition, and substitution.

Entities- there is no mechanism in XML Schema corresponding to the use of entities for data abstraction in DTDs. In many cases the functionality of entities can be replaced through other XML-based mechanisms. However, there is currently no support for named character entity references as used in XHTML within XML Schema.

DTDs and Document Order Dependence - a more subtle feature of modularized DTDs is their dependence on the document order; the order in which elements and entities are defined within DTD files has a large impact on language development. XML Schema are far less dependent on document order.

2.1.2. Document Data Structures

XML language definitions, regardless of their text representation, contain at least three types of data structures. When combined into a coherent and consistent whole, they form a complete language definition. These three components are:

- Elements
- Attributes
- Content models

Additional abstract data structures may be defined for use in the language definition, such as common content models or attribute groups, whose use is shared by other data structures within the language definition. The definition of these structures is the primary task of language development, and the core of the modularization framework.

2.1.3. Understanding XHTML Modularization

This schema modularization framework consists of two parts:

1) A set of schema modules that conform to the abstract modules in XHTML

2) A set of modularization conventions that describe how the individual modules work together, and how they can be modified or extended.

In XHTML-MOD, every object in the DTDs is represented by an XML entity. These entities are then composed into larger sets of entities and so on, resulting in a set of data abstractions that can be generalized and used modularly. These multiple levels of abstraction are tied together by the use of a specific naming convention and a set of abstract modules.

Generic classes of entities (composed of sub- and sub-sub-entities) are used to create definitions of the three components listed above. Content models, attribute lists and elements are defined separately, sometimes in separate modules, and the ordering of the modules in the DTD structure is strictly defined (due to document order dependence). They are then combined to form the resulting document type. Extensibility is accomplished through the extensive use of INCLUDE/IGNORE sections in the DTD modules. How each of these structures relates to its Schema-based counterpart is summarized in Table 1 below.

2.1.4. Mapping DTDs onto Schema

Both the DTD and schema-based modularization frameworks implement a set of formalized data structures, often in a conceptually similar way. The modularization framework described here is designed around the use of similar data structures, which can be represented (more or less) equally well in either representation. This is accomplished through the use of a straightforward mapping of data structures defined in the DTD modules onto equivalent data structures in the XML Schema language.

2.1.4.1. Content Models

In XHTML-MOD, content models for elements are defined using three classes of entities, identified through the naming conventions by the suffixes ".content", ".class", and ".mix". Each of these classes of entities is mapped onto a corresponding Schema counterpart in the following way:

".content" models - these models are used to define the contents of individual elements. For each element there is a corresponding ".content" object. IN XML Schema, ".content" entities are mapped directly onto groups:

Example 1 - Content Group

DTD	Schema

`<ENTITY % html.content "(head+,body+)">`	`<group name="html.content">`
	`  <sequence>`
	`  <element ref="head" minOccurs="1">`
	`  <element ref="body" minOccurs="1">`
	`  </sequence>`
	`</group>`

The contents of ".content" groups are often classes or mixes.

".class" models - these models are used to define abstract classes of content models made up of either ".content" entities or other ".class" entities (or elements). In XML Schema they correspond to groups that may also contain substitution groups:

Example 2 - ".class" Group

DTD	Schema

`<!ENTITY % Misc.class "%Edit.class;`
`  %Script.class;`
`  %Misc.extra;">`
	`<group name="Misc.class">`
	`  <choice minOccurs="0"`
	`  maxOccurs="unbounded">`
	`  <element ref="Edit.class"`
	`  abstract="true"/>`
	`  <element ref="Script.class"`
	`  abstract="true"/>`
	`  <element ref="Misc.extra"`
	`  abstract="true"/>`
	`  </choice>`
	`</group>`

".mix" models - these models correspond to content models that are mixed groupings of ".class", ".content", and ".mix" entities and serve as abstract content models often used in common by many elements in the DTD. They correspond to groups in XML Schema:

Example 3 - ".mix" Group

DTD	Schema

`<!ENTITY % Block.mix "%Heading.class;`
`  | %List.class;`
`  | %Block.class;`
`  %Misc.class;">`
	`<group name="Block.mix">`
	`  <choice minOccurs="0"`
	`  maxOccurs="unbounded">`
	`  <group ref="List.class"/>`
	`  <group ref="Block.class"/>`
	`  <group ref="Misc.class"/>`
	`  </choice>`
	`</group>`

In addition to these three content model groupings, XHTML-MOD includes an additional grouping ".extra". These are currently omitted from the schema modules. (If needed, a developer could add them to the schema modules in a conformant way.)

2.1.4.2. Attributes and Attribute Groups

Attributes and Attribute lists in DTDs correspond directly to attribute and attributeGroup elements in XML Schema. The translation from one to the other is relatively simple and straightforward. Here is an example:

Example 4 - Attribute Group

DTD	Schema
`<!ENTITY % title.attrib "title %Text.datatype; #IMPLIED">`	`<attributeGroup name="title"> <attribute name="title" type="string"/> </attributeGroup>`

Complex attribute groups that are used by many different elements are grouped in the DTDs using entities suffixed with ".attrib". These attribute entities map directly onto attributeGroup elements in XML Schema as shown above.

2.1.4.3. Complex Types and Element Definitions

The XML Schema specification allows elements as well as attribute values to be strongly typed. In defining elements in the modularized schema, an element type is created for each element that is a complex type composed of the content model (element.content) and the attribute list (element.attlist) as shown below:

Example 5 - Element Types In Schema

```
<complexType name="form.type">
    <group ref="form.content"/>
    <attributeGroup ref="form.attlist"/>
</complexType>
```

Elements are then declared to be of the type element.type:

Example 6 - Element Definition

```
<element name="form" type="form.type"/>
```

This allows the author the greatest degree of flexibility while retaining strict type checking via XML Schema. It also allows for extension of the element via type substitution.

Note that in the case of an element with a mixed content model, a complexType is necessary.

In summary, each element is composed of a content model and an attribute list, which are composed into a type for that element.

2.1.4.4. Attribute and Element Redefinitions

XML Schema allows inheritance and redefinition of elements, groups, attributes and attributeGroups. In several cases modules require modification of previously declared attribute lists. This is done by using the <xsd:redefine> element to redefine the attributeGroup that needs to be modified

Example 7 - attributeGroup Redefinition Example

```
<!- - new attribute to be added - - >
<attributeGroup name="align.legacy.attlist">
   <attribute name="align">
      <simpleType>
         <restriction base="NMTOKEN">
            <enumeration value="left"/>
            <enumeration value="center"/>
            <enumeration value="right"/>
            <enumeration value="justify"/>
         </restriction>
      </simpleType>
   </attribute>
</attributeGroup>

<!- - add it to the caption element's attribute group - - >
<redefine schemaLocation="xhtml-table-01.xsd">
   <attributeGroup name="caption.attlist">
      <extension base="align.attlist"/>
      <attributeGroup ref="align.legacy.attlist"/>
      </extension>
   </attributeGroup>
</redefine>
```

In this example, we redefine the attribute list for the caption element in the tables module to add the align attribute defined in align.legacy.attlist.

2.1.4.5. Support Structures

The modularized DTDs contain support mechanisms for XHTML. Some of these are DTD-specific and are not fully supported in XML Schema.

This modularization framework attempts to recreate these support structures to the greatest extent possible.

2.1.4.5.1. Notations

Notations are an SGML feature that allows non-SGML data within documents to be interpreted locally [CATALOG] [p.39] . Notations for XHTML are preserved in the Schema modules using the notation element in a straightforward way.

Example 8 - Notations

DTD	Schema
`<!NOTATION character PUBLIC "-//W3C//NOTATION XHTML Datatype: Character//EN">`	`<notation name="charset" public="-//W3C//NOTATION XHTML Datatype: Charset//EN"/>`

2.1.4.5.2. Data Types

The strong typing mechanism in XML Schema, along with the large set of intrinsic types and the ability to create user-defined types, provides for a high level of type safety in instance documents. This feature can be used to express more strict data type constraints, such as those of attribute values, when using XML Schema for validation.

Example 9 - Simple Data Types

DTD	Schema
`<!ENTITY % Length.datatype "CDATA" >`	`<simpleType name="Length"> <restriction base="string"/> </simpleType>`

2.1.4.5.3. Named Character Entities

XML Schema provides no means of duplicating XHTML's named character entity mechanism. In most cases data abstraction through entities can be dispensed with in schemas. However, in the case of named character references, no replacement method is available.

Character entities are used to represent characters that occur in document data that may not be processed natively on the user's machine, for instance the copyright symbol. XHTML makes use of 3 sets of named character entities: the ISO Latin 1, Symbols, and Special.

A general solution for the resolution of language-specific named character entities is outside the scope of this document.

Entities are currently referenced in this framework as notations; this is strictly incorrect. However it is considered desirable to include references to the XHTML-specific entity sets.

Here is an example of the use of the notation declaration syntax for named character references:

Example 10 - Entity References

```
<notation name="XHTML-Latin-1"
   id="xhtml-lat1"
   public="-//W3C//ENTITIES Latin 1 for XHTML//EN"
   system=" http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent"/>
```

2.1.4.6. Mapping Summary

The following table summarizes the mapping of DTD data structures onto XML Schema structures.

Table 1 - Mapping of DTD and Schema Data Structures

DTD Entity	Use	Schema Element
.content	Element content model	group
.class	Abstract content model	group
.mix	Abstract content model	group
.attlist	Attribute lists	attributeGroup
.attrib	attributes	Attribute
.extra	Abstract attribute group	attributeGroup
elements	Element definitions	Elements+complexType
attribute redefinition	Attribute list redefinition	AttributeGroup w/redefine
notation	SGML specific	notation
datatypes	attribute datatypes	simpleType
entities	Character replacement	notation reference
DTD "driver"	Framework document	"Hub" Schema document

One further issue of note in the conversion of DTDs to XML Schema is that it is absolutely necessary to define all elements globally. Otherwise they are not considered to be in the XHTML namespace but only "associated" with it. This document does not make use of this association feature in XML Schema.

2.2. Framework Conventions

This section is *normative*.

This modularization framework consists of a complete set of XHTML schema modules and a set of framework conventions that describe how to use them. The use of the framework conventions is required for conformance.

2.2.1. Modularized Schema Files

The modularized XHTML schema uses three types of files, which when combined comprise the entire XHTML grammar definition.

2.2.1.1. Hub document

The Schema hub document is the base document for the schema. It contains only annotations and modules, which in turn contain <xsd:include> statements referencing other module files. The hub document corresponds to the DTD "driver" file in XHTML-MOD, but is much simpler. The hub document allows the author to modify the schema's contents by the simple expedient of commenting out modules that are not used. Note that some modules are always required in order to ensure conformance.

The (non-normative) hub document described here contains <include> elements for two modules, named "required" and "optional". Each of these included modules is itself a module container file.

2.2.1.2. Module container files

Module container files, reasonably enough, include other modules. Modules and their containers are organized according to function. Including the hub document, which is a special case of a module container, there are ten included module container files.

2.2.1.3. Element modules

In addition to the module container files listed above, there are around forty schema modules which contain only element definitions and their associated attribute and content model definitions. By convention, Schema modularization files may contain either <include> statements or element definitions but not both.

2.2.2. Module Naming

In order to easily identify the contents of any particular schema module file, it is useful to provide here a module naming convention syntax. This syntax also provides a simple means of distinguishing module files based on their language version, which may improve maintainability of the modules themselves.

The module naming convention adopted here is the same in almost all respects as that used in XHTML-MOD.

Schema modules for XHTML should have names that are:

- Supported on all common platforms
- Identify the contents of the file
- Identify the language version of the file

Modules used in this modularization framework must have names that conform to the following syntax:

Example 11 - Schema Module Naming Convention

Pattern	`languagename-filecontentsdescription-versionnumber.xsd`
Example	`xhtml-table-01.xsd`

Exceptions to this rule are made for the Schema hub modules whose names are the same as above but may omit the file contents description syllable for brevity.

Version numbers of hub modules may omit the leading zero in the version number, but should include the minor version number.

Example: xhtml-1.1.xsd

In the case where a hub module contains elements or attributes from external namespaces, the name(s) of the external module(s) should be appended to the base language name using the "+" character.

Example: xhtml+fml-1.0.xsd

This module naming convention is intended also to comply with the required use of the media type in [XHTMLMIME] [p.40] .

2.2.3. Module Hierarchy Structure

In order to establish a physical structure for the composition of the Schema modules that corresponds to the abstract modules in XHTML, a module hierarchy structure has been used to organize the physical modules. The hierarchy structure looks like this:

Table 2 - Schema Module Hierarchy Structure

```
xhtml/
xhtml/req/
xhtml/req/framework/
xhtml/req/core/
xhtml/req/core/text/
xhtml/opt/
xhtml/opt/pres/
xhtml/opt/legacy/
xhtml/opt/legacy/misc/
xhtml/opt/legacy/frames/
```

These correspond to the divisions of XHTML into abstract modules described in detail in Section 3.2. The hierarchy structure is intended to match the abstract module structure as closely as possible. This feature is not present in DTD modularization, and is not required for Schema modularization. It does, however, allow the developer to organize the modules in accordance with their hierarchical structure. The directories listed in Table 2 also correspond exactly to the module container modules in this framework.

2.2.4. Names for Data Structures

The consistent use of naming conventions is important for the maintenance and development of complex software applications.

Adhering to these conventions provides numerous benefits to developers:

Simplifies testing and debugging by managing complexity

Eases maintenance by allowing any developer to read and understand another developer's code

Provides self-documenting code by using descriptive names and predictable naming conventions

Enforces encapsulation by using consistent naming conventions for public and private knowledge.

With few exceptions, the naming conventions used in XHTML-MOD are preserved in this framework.

The naming convention in XHTML-MOD uses suffixing of object names to indicate functionality, as described below.

2.2.4.1. Attributes

Abstract attribute groups and attribute lists are suffixed with the ".attrib" and ".attlist" suffixes respectively.

2.2.4.2. Content models

Three different suffixes are used in content model names. They are ".content" for element content models, and ".class" or ".mix" for abstract content models.

2.2.4.3. Elements

Element names are not suffixed in XHTML-MOD. This document uses the notion of element types, which are complexTypes used to define elements and are suffixed with ".type". The ".type" suffix was used in XHTML-MOD for attribute data types. This is superfluous in XML Schema (since attribute types are arguments to the "type" attribute) and so the suffix is used in a different way in this framework.

2.2.5. Module Structure

This document establishes a convention for the internal structure of XHTML Schema modules. This convention provides a consistent and predictable way of organizing schema module files internally. This convention applies also to the hub document, which is itself simply a module of modules, albeit a somewhat specialized one.

Each schema module is composed of several components, some of which are required for functional reasons and some of which provide metadata as a convenience to the author. Not every component is included in every module.

2.2.5.1. Schema Element

Each file begins with a <xsd:schema> root element (after the optional xml declaration and DOCTYPE).

2.2.5.1.1. Use of Version Attribute

In the XHTML schema modules, the version number for the specific language being defined (e.g. "1.1") is used as the default value of the version attribute on the schema element.

2.2.5.1.2. Qualified names

This framework uses the value of "unqualified" for the value of the elementFormDefault attribute on the schema root element. Elements within the html namespace do not need to use a namespace prefix.

2.2.5.2. Annotation Block

After the root element each module contains an annotation element containing several documentation sections briefly describing the purpose of the module.

2.2.5.2.1. Module Description

This is an annotation element that contains a short description of the module and its purpose.

2.2.5.2.2. Versioning Block

An annotation element containing authoring and versioning information for the module should always be included.

2.2.5.2.3. Copyright

The standard W3C copyright statement is included in each module through the use of an include element. An exception is the hub document, which contains the full copyright text.

2.2.5.2.4. Documentation

This is a module specific documentation element providing detailed information about the file's contents, its organization, and any noteworthy items of interest to developers.

2.2.5.3. 3. Module elements

Module elements contain include statements, import statements, or other modules (or comments). They must precede any other definitions in the file.

2.2.5.4. 4. Content model groups

These include groups with names ending in ".content", ".class", or ".mix".

2.2.5.5. 5. Attributes and Attribute groups

These are suffixed with either ".attrib" or ".attlist".

2.2.5.6. 6. Element type definitions

These are complexType elements defining each element's type.

2.2.5.7. 7. Element definitions

These define individual elements in the module.

Additional constraints on the internal structure of schema modules are:

Each module must contain include statements for other modules or data structure definitions, but not both.

Each module must include at least sections 1 and 2 above, as well either section 3 or some combination of sections 4-7.

2.2.6. Namespace Conventions

The handling of namespaces in XML Schema is entirely different from that in XHTML-MOD. Namespaces are integral to XML Schema and their use in modularization arises naturally from the schema syntax.

One convention chosen for this framework is that the names of elements and attributes in the modules are unqualified i.e. no namespace prefix is required for XHTML elements.

This is set by using the value of "unqualified" on the elementFormDefault attribute of the xsd:schema element.

2.2.7. Documentation Conventions

A consistent commenting convention has been imposed on the modules described here. The purpose of a commenting convention is to allow for generating documentation from the comments (as well as general comprehension). Documentation elements containing Annotation-level comments are assumed to be of the highest importance and should be used to denote information about the module itself, and for important notes for developers.

File-level comments are denoted as usual with SGML comment delimiters "<!--" and "-->". By means of this convention, modules can become self-documenting. Tools for extracting these comments and formatting them suitably may (hopefully) be developed in the future.

3. XHTML Schema Modules

This chapter is *normative*.

3.1. XHTML Abstract Modules

The DTD modularization framework specification speaks at length on the subject of abstract modules. In brief, an "abstract" module is simply a set of objects, in this case objects within an ordered hierarchy of content objects, which encapsulates all of the features of the objects and assembles them into a coherent set. This set of objects and their properties is independent of its machine representation, and so is the same whether written in DTD module form, as a Schema module, or as a Java class.

The abstract modules described in XHTML-MOD are composed in a functional manner, and each "abstract module" contains data structures that are generally functionally similar. (There is no requirement that modules be created along functional lines; any other method that suits the author's purpose may be used instead.)

The framework described here makes use of the same abstract modules as in XHTML-MOD with few exceptions. In the case of the schema module representation, the relationship between the "abstract" modules and the schema modules is quite close. In each case there is a one-to-one relationship between the abstract and concrete modules (with one exception for the changes to the legacy module) and they share essentially the same names and data structures.

3.2. XHTML Schema Modules

3.2.1. Required Modules

These modules must be included in any document that uses the XHTML namespace. Each section below describes the purpose of the module and its contents.

Schema location	SCHEMA/req/xhtml-framework-1.xsd [p.50]
Use	Required
Type	Module Container
Description	Required XHTML modules
Contents	SCHEMA/req/xhtml-framework-1.xsd [p.50] SCHEMA/req/xhtml-core-1.xsd [p.127]
Redefinitions	No
Dependencies	None

These files should not be modified by developers; instead use <redefine> or a substitution group.

3.2.1.1. Framework Modules

This is a module container for XHTML language support modules.

Schema location	SCHEMA/req/xhtml-framework-1.xsd [p.50]
Use	Required
Type	Module Container
Description	Language support modules
Contents	SCHEMA/req/framework/xhtml-notations-1.xsd [p.52] SCHEMA/req/framework/xhtml-datatypes-1.xsd [p.54] SCHEMA/req/framework/xhtml-events-1.xsd [p.93] SCHEMA/req/framework/xhtml-attribs-1.xsd [p.55] SCHEMA/req/framework/xhtml11-model-1.xsd [p.129] SCHEMA/req/framework/xhtml-charent-1.xsd [p.57]
Redefinitions	No
Dependencies	None

3.2.1.1.1. Notations

Schema location	SCHEMA/req/framework/xhtml-notations-1.xsd [p.52]
Use	Required
Type	Language Support- SGML notations
Contents	SGML Notations- see the SGML catalog file
Redefinitions	No
Dependencies	None

3.2.1.1.2. Data types

Schema location	SCHEMA/req/framework/xhtml-datatypes-1.xsd [p.54]
Use	Required
Type	Language Support - common data types
Contents	XHTML data type definitions
Redefinitions	No
Dependencies	None

3.2.1.1.3. Events

Schema location	SCHEMA/req/framework/xhtml-events-1.xsd [p.93]
Use	Required
Type	Language Support - common events attributes
Contents	Common events attributes for XHTML
Redefinitions	Yes
Dependencies	Element definitions

3.2.1.1.4. Common Attributes

Schema location	SCHEMA/req/framework/xhtml-attribs-1.xsd [p.55]
Use	Required
Type	Language Support - common attribute groups
Contents	Abstract attribute groups
Redefinitions	No
Dependencies	Element definitions

3.2.1.1.5. Common Content Models

Schema location	SCHEMA/req/framework/xhtml11-model-1.xsd [p.129]
Use	Required
Type	Language Support - common content model groups
Contents	Abstract content models
Redefinitions	No
Dependencies	Element definitions

3.2.1.1.6. Character Entities

The character entities module includes three notation elements within an <appinfo> element, each referencing one of the required entity sets in XHTML: ISO Latin-1, Symbols, and Special characters.

Character entities are not fully supported in XML Schema, as described in Section 2.1.

Schema location	SCHEMA/req/framework/xhtml-charent-1.xsd [p.57]
Use	Required
Type	Language Support
Contents	Character Entities for XHTML
Redefinitions	No
Dependencies	None

3.2.1.2. Core Element Modules

These are the core element definitions for the required modules.

Schema location	SCHEMA/req/core/xhtml-core-1.xsd [p.127]
Use	Required
Type	Module Container
Description	Core element modules
Contents	SCHEMA/req/core/xhtml-text-1.xsd [p.59] SCHEMA/req/core/xhtml-hypertext-1.xsd [p.61] SCHEMA/req/core/xhtml-list-1.xsd [p.62] SCHEMA/req/core/xhtml-ruby-1.xsd [p.104] SCHEMA/req/core/xhtml-struct-1.xsd [p.58]
Redefinitions	No
Dependencies	None

3.2.1.2.1. Text Modules

Schema location	SCHEMA/req/core/xhtml-text-1.xsd [p.59]
Use	Required
Type	Module Container
Description	Text element modules
Contents	SCHEMA/req/core/text/xhtml-blkphras-1.xsd [p.107] SCHEMA/req/core/text/xhtml-blkstruct-1.xsd [p.109] SCHEMA/req/core/text/xhtml-inlphras-1.xsd [p.110] SCHEMA/req/core/text/xhtml-inlstruct-1.xsd [p.113]
Redefinitions	No
Dependencies	None

Block Phrasal

Schema location	SCHEMA/req/core/text/xhtml-blkphras-1.xsd [p.107]
Use	Required
Type	Element definitions
Redefinitions	No
Dependencies	None
Elements	address [p.108] blockquote [p.108] h1 [p.108] h2 [p.108] h3 [p.108] h4 [p.108] h5 [p.108] h6 [p.108] pre [p.108]
Redefinitions	No
Dependencies	None

Block Structural

Schema location	SCHEMA/req/core/text/xhtml-blkstruct-1.xsd [p.109]
Use	Required
Type	Element definitions
Elements	div [p.110] p [p.110]
Redefinitions	No
Dependencies	None

Inline Phrasal

Schema location	SCHEMA/req/core/text/xhtml-inlphras-1.xsd [p.110]
Use	Required
Type	Element definitions
Elements	abbr [p.111] acronym [p.111] cite [p.111] code [p.111] dfn [p.111] em [p.111] kbd [p.111] q [p.111] samp [p.111] strong [p.111] var [p.111]
Redefinitions	No
Dependencies	None

Inline Structural

Schema location	SCHEMA/req/core/text/xhtml-inlstruct-1.xsd [p.113]
Use	Required
Type	Element definitions
Elements	br [p.113] span [p.114]
Redefinitions	No
Dependencies	None

3.2.1.2.2. Hypertext

Schema location	SCHEMA/req/core/xhtml-hypertext-1.xsd [p.61]
Use	Required
Type	Element definitions
Elements	a [p.62]
Redefinitions	No
Dependencies	None

3.2.1.2.3. Lists

Schema location	SCHEMA/req/core/xhtml-list-1.xsd [p.62]
Use	Required
Type	Element definitions
Elements	dd [p.63] dl [p.63] dt [p.63] li [p.63] ol [p.63] ul [p.64]
Redefinitions	No
Dependencies	None

3.2.1.2.4. Ruby

Ruby elements denote annotations used in some Asian languages. [RUBY] [p.40]

The Ruby module has been moved into the core element definitions module because it is normatively required in XHTML 1.1

Schema location	SCHEMA/req/core/xhtml-ruby-1.xsd [p.104]
Use	Required
Type	Element definitions
Elements	rb [p.106] rbc [p.106] rp [p.107] rt [p.106] rtc [p.106] ruby [p.105]
Redefinitions	No
Dependencies	None

3.2.1.2.5. Structural

Schema location	SCHEMA/req/core/xhtml-struct-1.xsd [p.58]
Use	Required
Type	Element definitions
Elements	body [p.59] head [p.59] html [p.59] title [p.59]
Redefinitions	No
Dependencies	None

3.2.2. Optional Modules

These modules are (clearly) optional; they may be removed or combined arbitrarily (except for dependencies). Developers should not modify the contents of these files as they part of the XHTML definition. Instead, extension in the optional modules should be confined to redefinitions and derivations.

Schema location	SCHEMA/xhtml-optional-1.xsd [p.121]
Use	Required
Type	Module Container
Description	Optional modules
Contents	SCHEMA/opt/xhtml-edit-1.xsd [p.65] SCHEMA/opt/xhtml-bdo-1.xsd [p.66] SCHEMA/opt/xhtml-link-1.xsd [p.99] SCHEMA/opt/xhtml-meta-1.xsd [p.96] SCHEMA/opt/xhtml-base-1.xsd [p.101] SCHEMA/opt/xhtml-script-1.xsd [p.97] SCHEMA/opt/xhtml-style-1.xsd [p.98] SCHEMA/opt/xhtml-image-1.xsd [p.81] SCHEMA/opt/xhtml-csismap-1.xsd [p.82] SCHEMA/opt/xhtml-ssismap-1.xsd [p.85] SCHEMA/opt/xhtml-param-1.xsd [p.114] SCHEMA/opt/xhtml-applet-1.xsd [p.85] SCHEMA/opt/xhtml-object-1.xsd [p.87] SCHEMA/opt/xhtml-table-1.xsd [p.76] SCHEMA/opt/xhtml-form-1.xsd [p.70] SCHEMA/opt/xhtml-nameident-1.xsd [p.102] SCHEMA/opt/xhtml-legacy-1.xsd [p.103] SCHEMA/opt/frames/xhtml-frames-1.xsd [p.88] SCHEMA/opt/frames/xhtml-target-1.xsd [p.90] SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.92]
Redefinitions	No
Dependencies	None

3.2.2.1. Edit

Schema location	SCHEMA/opt/xhtml-edit-1.xsd [p.65]
Use	Optional
Type	Element definitions
Elements	del [p.66] ins [p.66]
Redefinitions	No
Dependencies	None

3.2.2.2. Bdo

Schema location	SCHEMA/opt/xhtml-bdo-1.xsd [p.66]
Use	Optional
Type	Element definitions
Elements	bdo [p.67]
Redefinitions	No
Dependencies	None

3.2.2.3. Presentational

Schema location	SCHEMA/opt/xhtml-pres-1.xsd [p.64]
Use	Optional
Type	Module Container
Description	Presentational element modules
Contents	SCHEMA/opt/pres/xhtml-blkpres-1.xsd [p.108] SCHEMA/opt/pres/xhtml-inlpres-1.xsd [p.112]

3.2.2.3.1. Inline Presentational

Schema location	SCHEMA/opt/pres/xhtml-inlpres-1.xsd [p.112]
Use	Optional
Type	Element definitions
Elements	b [p.112] big [p.112] i [p.113] small [p.113] sub [p.113] sup [p.113] tt [p.113]
Redefinitions	No
Dependencies	None

3.2.2.3.2. Block Presentational

Schema location	SCHEMA/opt/pres/xhtml-blkpres-1.xsd [p.108]
Use	Optional
Type	Element definitions
Elements	hr [p.109]
Redefinitions	No
Dependencies	None

3.2.2.4. Link

Schema location	SCHEMA/opt/xhtml-link-1.xsd [p.99]
Use	Optional
Type	Element definitions
Elements	link [p.100]
Redefinitions	No
Dependencies	None

3.2.2.5. Meta

Schema location	SCHEMA/opt/xhtml-meta-1.xsd [p.96]
Use	Optional
Type	Element definitions
Elements	meta [p.97]
Redefinitions	No
Dependencies	None

3.2.2.6. Base

Schema location	SCHEMA/opt/xhtml-base-1.xsd [p.101]
Use	Optional
Type	Element definitions
Elements	base [p.101]
Redefinitions	No
Dependencies	None

3.2.2.7. Scripting

Schema location	SCHEMA/opt/xhtml-script-1.xsd [p.97]
Use	Optional
Type	Element definitions
Elements	noscript [p.98] script [p.97]
Redefinitions	No
Dependencies	None

3.2.2.8. Style

Schema location	SCHEMA/opt/xhtml-style-1.xsd [p.98]
Use	Optional
Type	Element definitions
Elements	style [p.98]
Redefinitions	No
Dependencies	None

3.2.2.9. Image

Schema location	SCHEMA/opt/xhtml-image-1.xsd [p.81]
Use	Optional
Type	Element definitions
Elements	img [p.82]
Redefinitions	No
Dependencies	None

3.2.2.10. Client-side Image Maps

Schema location	SCHEMA/opt/xhtml-csismap-1.xsd [p.82]
Use	Optional
Type	Element definitions
Elements	area [p.83] map [p.84]
Redefinitions	No
Dependencies	None

3.2.2.11. Server-side Image Maps

Schema location	SCHEMA/opt/xhtml-ssismap-1.xsd [p.85]
Use	Optional
Type	Attribute definitions
Redefinitions	No
Dependencies	None

3.2.2.12. Param

Schema location	SCHEMA/opt/xhtml-param-1.xsd [p.114]
Use	Optional
Type	Element definitions
Elements	param [p.115]
Redefinitions	No
Dependencies	None

3.2.2.13. Applet

Schema location	SCHEMA/opt/xhtml-applet-1.xsd [p.85]
Use	Optional
Type	Element definitions
Elements	applet [p.87]
Redefinitions	No
Dependencies	Param::param

3.2.2.14. Object

Schema location	SCHEMA/opt/xhtml-object-1.xsd [p.87]
Use	Optional
Type	Element definitions
Elements	object [p.88]
Redefinitions	No
Dependencies	Param::param

3.2.2.15. Tables

Schema location	SCHEMA/opt/xhtml-table-1.xsd [p.76]
Use	Optional
Type	Element definitions
Elements	caption [p.81] col [p.79] colgroup [p.80] table [p.81] tbody [p.80] td [p.79] tfoot [p.80] th [p.79] thead [p.81] tr [p.79]
Redefinitions	No
Dependencies	None

3.2.2.16. Forms

Schema location	SCHEMA/opt/xhtml-form-1.xsd [p.70]
Use	Optional
Type	Element definitions
Elements	button [p.73] fieldset [p.73] form [p.71] input [p.72] label [p.71] legend [p.73] optgroup [p.72] option [p.72] select [p.72] textarea [p.73]
Redefinitions	No
Dependencies	None

3.2.2.17. Nameident

Schema location	SCHEMA/opt/xhtml-nameident-1.xsd [p.102]
Use	Optional
Type	Attribute definitions
Redefinitions	No
Dependencies	None

3.2.2.18. Legacy

This module has been reorganized to conform to the framework conventions used here. It has been divided here into two separate modules. The "misc" module contains everything in the DTD legacy model except frames. Frames are now in a separate module called framedefs. This allows the developer to easily separate the legacy features if desired.

Schema location	SCHEMA/opt/xhtml-legacy-1.xsd [p.103]
Use	Optional
Type	Module container
Contents	SCHEMA/opt/misc/xhtml-misc-1.xsd [p.115] SCHEMA/opt/xhtml-framedefs-1.xsd [p.120]
Redefinitions	No
Dependencies	None

3.2.2.18.1. Misc

Schema location	SCHEMA/opt/misc/xhtml-misc-1.xsd [p.115]
Use	Optional
Type	Element definitions
Elements	basefont [p.116] center [p.116] dir [p.117] font [p.116] isindex [p.117] menu [p.117] s [p.116] strike [p.116] u [p.116]
Redefinitions	Yes
Dependencies	Yes

3.2.2.18.2. Framedefs

Schema location	SCHEMA/opt/xhtml-framedefs-1.xsd [p.120]
Use	Optional
Type	Element definitions
Contents	SCHEMA/opt/xhtml-frames-1.xsd [p.120] SCHEMA/opt/frames/xhtml-target-1.xsd [p.90] SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.92]
Redefinitions	Yes
Dependencies	Yes

Frames

Schema location	SCHEMA/opt/frames/xhtml-frames-1.xsd [p.88]
Use	Optional
Type	Element definitions
Elements	frame [p.89] frameset [p.89] noframes [p.90]
Redefinitions	Yes
Dependencies	Target::target

Target

Schema location	SCHEMA/opt/frames/xhtml-target-1.xsd [p.90]
Use	Optional
Type	Attribute redefinitions
Redefinitions	Yes
Dependencies	Yes

Iframe

Schema location	SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.92]
Use	Optional
Type	Element definitions
Elements	iframe [p.93]
Redefinitions	Yes
Dependencies	Target::target

3.2.2.19. Basic Forms

Schema location	SCHEMA/opt/xhtml-basic-form-1.xsd [p.67]
Use	Optional
Type	Element definitions
Elements	form [p.68] input [p.69] label [p.68] option [p.69] select [p.69] textarea [p.69]
Redefinitions	No
Dependencies	Removal of Forms

3.2.2.20. Basic Tables

Schema location	SCHEMA/opt/xhtml-basic-table-1.xsd [p.74]
Use	Optional
Type	Element definitions
Elements	caption [p.75] table [p.75] td [p.76] th [p.76] tr [p.76]
Redefinitions	No
Dependencies	Removal of Tables

3.2.3. XHTML Hub Document (Non-normative)

This is an example base schema document that includes all the other modules to create the complete schema.

3.2.3.1. XHTML 1.1

The hub document included here intends to approximate XHTML 1.1 subject to the requirements given in Section 1.4. This schema should be fully equivalent to the DTD version except for schema-specific additions and changes. This hub document is non-normative and provided only as an example.

Schema location	SCHEMA/xhtml-1.1.xsd [p.47]
Use	Main schema document
Type	Module Container
Description	Hub document
Redefinitions	No
Dependencies	None
Contents	SCHEMA/req/xhtml-framework-1.xsd [p.50] SCHEMA/xhtml-optional-1.xsd [p.121]

3.3. Validity and Conformance

The purpose of any language definition, regardless of its basis on DTDs, XML Schema, or some other representation, is the same: to determine if a specific document instance conforms to the language definition. In XML Schema terms, this means that documents can be validated using the schema. The validation process attempts to determine the document's structural integrity, and the behavior of any XML processor in cases of validation errors is well-defined in the XML 1.0 specification. Therefore the real test of any modularization system for XHTML is whether the resulting schema can be used to determine if any particular XHTML document instance is valid.

This document does not attempt to define conformance beyond the ability to validate the structural integrity of documents. In particular it does not attempt to describe any level of user-agent conformance, as this is not a modularization issue, but an issue for the specification of the language semantics. Conformance to the XML Schema-based modularization framework is strictly defined in terms of document validation. Further levels of conformance are described in the published language specifications themselves.

3.3.1. XHTML Conformance

Schemas defining language variants within the XHTML namespace may be considered to be conformant if they:

- Conform to the existing published language definition for that language variant
- Include all of the modules that are required in this document
- Retain the semantic components of the required elements intact including but not limited to elements, attributes, and attribute values
- Properly identify external elements and attributes with the appropriate XML namespace
- Any combination of the XHTML Schema modules provided that complies with these rules can be considered to be a conformant language variant in the XHTML namespace.

3.3.2. Schema Modularization Conformance

An XML Schema or set of Schema modules can be considered to be conformant to this schema modularization framework if they follow the schema modularization framework conventions described in Section 2.2.

3.3.3. The XHTML Family of Documents

The XHTML Family of Documents is defined as the set of language variants that use the XHTML namespace as the namespace of the root element, which must be <html>.

In order to be a conformant member of the XHTML Family of Documents, an XML Schema or set of schema modules must:

- Define <html> as its root element, and use the XHTML namespace as the root element's namespace.
- Conform to the XHTML Conformance and Schema Modularization Conformance sections of this document.

This class of document definitions includes both XHTML language variants and compound document types using external modules.

3.3.4. Versioning

Versioning of modules that claim conformance to this specification is subject to the framework conventions in Section 2.2. Versioning information should be available in the version block section of each conformant module.

A. References

This appendix is *normative*.

A.1. Normative References

[APPROACH]
An Approach to the Modularization of XHTL using XML Schema, Rick Jelliffe, Academia Sinica Computing Center 12 December 2000.
See: http://www.ascc.net/~ricko/xhtml.htm

[CSS2]
"Cascading Style Sheets, level 2 (CSS2) Specification", B. Bos, H. W. Lie, C. Lilley, I. Jacobs, 12 May 1998.
Available at: http://www.w3.org/TR/1998/REC-CSS2-19980512

[DOM]
"Document Object Model (DOM) Level 1 Specification", Lauren Wood *et al.*, 1 October 1998.
Available at: http://www.w3.org/TR/REC-DOM-Level-1-19981001

[CATALOG]
Entity Management: OASIS Technical Resolution 9401:1997 (Amendment 2 to TR 9401) Paul Grosso, Chair, Entity Management Subcommittee, SGML Open, 10 September 1997.
See: http://www.oasis-open.org/html/a401.htm

[HTML4]
HTML 4.01 Specification: W3C Recommendation, Dave Raggett, Arnaud Le Hors, Ian Jacobs, 24 December 1999.
See: http://www.w3.org/TR/1999/REC-html401-19991224

[ISO10646]
"Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:2000. This reference refers to a set of codepoints that may evolve as new characters are assigned to them. Also, this reference assumes that the character sets defined by ISO 10646 and [Unicode] [p.40] remain character-by-character equivalent. This reference also includes future publications of other parts of 10646 (i.e., other than Part 1) that define characters in planes 1-16.

[MathML]
Mathematical Markup Language 2, David Carlisle, et al. 21 February 200.
See: http://www.w3.org/TR/2001/REC-MathML2-20010221

[RFC1808]
Relative Uniform Resource Locators, R. Fielding.
See: http://www.ietf.org/rfc/rfc1808.txt

[RFC2045]
"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[RFC2119]
"Key words for use in RFCs to indicate requirement levels", S. Bradner, March 1997.

[RFC3066]
"Tags for the Identification of Languages", H. Alvestrand, January 2001.
[RFC1808]
[RUBY]
Ruby Annotation, Marcin Sawicki, et. al., 16 February 2001
See: http://www.w3.org/TR/ruby
[SGML]
Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML), ISO 8879:1986.
Please consult http://www.iso.ch/cate/d16387.html for information about the standard, or http://www.oasis-open.org/cover/general.html#overview about SGML.
[SMIL]
Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, Philipp Hoschka, 15 June 1998.
See: http://www.w3.org/TR/1998/REC-smil-19980615
[SRGB]
"A Standard Default Color Space for the Internet", version 1.10, M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, 5 November 1996. This document is http://www.w3.org/Graphics/Color/sRGB
[UNICODE]
The Unicode Standard, Version 3.0, The Unicode Consortium, Reading Mass. Addison-Wesley Developers Press, 2000. ISBN 0-201-61633-5. See http://www.unicode.org.
[URI]
Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, August 1998.
See: http://www.ietf.org/rfc/rfc2396.txt. This RFC updates RFC 1738 [URL] [p.40] and [RFC1808] [p.40] .
[URL]
IETF RFC 1738, Uniform Resource Locators (URL), T. Berners-Lee, L. Masinter, M. McCahill.
See: http://www.ietf.org/rfc/rfc1738.txt
[XHTML1]
XHTML 1.0: The Extensible HyperText Markup Language, Steven Pemberton, et al., 26 January 2000.
See: http://www.w3.org/TR/2000/REC-xhtml1-20000126
[XHTML11]
XHTML 1.1 - Module-based XHTML, Murray Altheim, et. al., 5 January 2000.
See: http://www.w3.org/TR/2000/WD-xhtml11-20000105/
[XHTMLBASIC]
XHTML Basic, Mark Baker, et. al., 19 December 2000.
See: http://www.w3.org/TR/xhtml-basic/
[XHTMLMIME]
The application/xhtml+xml Media Type, Mark Baker, IETF, January 2001.
See: http://www.ietf.org/

[XHTMLMOD]
Modularization of XHTML, Murray Altheim, et. al., 22 February 2001
See: http://www.w3.org/TR/2001/PR-xhtml-modularization-20010222
[XML]
Extensible Markup Language (XML) 1.0: Second Edition, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, 6 October 2000.
See: http://www.w3.org/TR/2000/REC-xml-20001006
[XMLNAMES]
"Namespaces in XML", T. Bray, D. Hollander, A. Layman, 14 January 1999.
XML namespaces provide a simple method for qualifying names used in XML documents by associating them with namespaces identified by URI.
Available at: http://www.w3.org/TR/1999/REC-xml-names-19990114
[XMLSCHEMA]
XML Schema Part 1: Structures, Henry S. Thompson, et al., 16 March 2001
See: http://www.w3.org/TR/2001/PR-xmlschema-1-20010316

B. Changes

This appendix is *informative*.

B.1. Changes to Abstract Modules

- Moved Ruby to Optional

B.2. Changes from DTD Module Implementations

- Reorganized legacy with 2 new modules
- Changed Ruby content model
- Moved nameident into optional
- Modified meaning of ".type" objects
- Added file naming, file structure, module structure, and documentation conventions.
- Changed some attribute value definitions to more restricted types
- Added new "req" module

C. Acknowledgements

This appendix is *informative*.

The following people provided support and assistance:

- Steven Pemberton
- Rick Jelliffe
- Josef Dietl
- Sebastian Schnitzenbaumer
- Malte Wedel
- Bob Lojek
- Jack Herer

D. XHTML Schema Module Implementations

This appendix is *normative*.

This appendix contains implementations of the modules defined in XHTML Abstract Modules via XML Schema [XMLSCHEMA] [p.41] when the XML Schema becomes a W3C approved recommendation.

D.1. XHTML Abstract Modules and XML Schema

The DTD modularization framework specification speaks at length on the subject of abstract modules. In brief, an "abstract" module is simply a set of objects, in this case objects within an ordered hierarchy of content objects, which encapsulates all of the features of the objects and assembles them into a coherent set. This set of objects and their properties is independent of its machine representation, and so is the same whether written in DTD module form, as a Schema module, or as a Java class.

The abstract modules described in XHTML-MOD are composed in a functional manner, and each "abstract module" contains data structures that are generally functionally similar. (There is no requirement that modules be created along functional lines; any other method that suits the author's purpose may be used instead.)

The framework described here makes use of the same abstract modules as in XHTML-MOD with few exceptions. In the case of the schema module representation, the relationship between the "abstract" modules and the schema modules is quite close. In each case there is a one-to-one relationship between the abstract and concrete modules (with one exception for the changes to the legacy module) and they share essentially the same names and data structures.

D.2. XHTML Schema Modules

D.2.1. XHTML Hub Document

This is the base schema document that includes all the other modules to create the complete schema.

The hub document included here intends to approximate XHTML 1.1 subject to the requirements given in Requirements [p.6] . This schema should be fully equivalent to the DTD version except for schema-specific additions and changes. Any document instance considered valid using the XHTML 1.1 DTD must also be valid according to this schema.

```
<?xml version="1.0"?>

<schema targetNamespace="http://www.w3.org/1999/xhtml"
        xmlns="http://www.w3.org/2000/10/XMLSchema"
    elementFormDefault="unqualified"
    version="1.1">
```

```
    <annotation>
        <documentation>
/**
* This is the XML Schema driver for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//SCHEMA XHTML//EN"
* And this namespace for XHTML elements:
*        xmlns:xhtml="http://www.w3.org/1999/xhtml"
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
*   This is XHTML, a reformulation of HTML as a modular XML application.
*
*     The Extensible HyperText Markup Language (XHTML)
*     Copyright 1998-2000 World Wide Web Consortium
*     (Massachusetts Institute of Technology, Institut National de
*      Recherche en Informatique et en Automatique, Keio University).
*      All Rights Reserved.
*
*     Permission to use, copy, modify and distribute the XHTML DTD and its
*     accompanying documentation for any purpose and without fee is hereby
*     granted in perpetuity, provided that the above copyright notice and
*     this paragraph appear in all copies.  The copyright holders make no
*     representation about the suitability of the DTD for any purpose.
*
*     It is provided "as is" without expressed or implied warranty.
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
*
* This document is based on an approach to modular schemas originally suggested by
* Rick Jelliffe ricko@gate.sinica.edu.tw and members of the XML Schema Working Group at

* W3C.
```

```
* http://www.ascc.net/~ricko/xhtml.htm

*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
* In order to modify this schema, you can modify the files
* containing the optional modules and if necessary make changes
* to the common content models module. It is not necessary to modify this file.
*
*/
        </documentation>
    </annotation>

    <!-- required modules section begins here -->

    <annotation>
        <documentation>
/**
* These modules are required to be included in any member of the
* XHTML Family of Documents TM.
*
* In order to modify this schema, you can modify the files
* containing the optional modules and if necessary make changes
* to the common content models module. It is not necessary to modify this file.
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
* Framework module
*
*        +  notations
*        +  datatypes
*        +  xlink
*        +  events
*        +  common attributes
*        +  common content models
*        +  character entities
*
*/
        </documentation>
    </annotation>
    <include schemaLocation="xhtml-framework-1.xsd"/>
    <annotation>
        <documentation>
/**
* Core Required elements modules
*
*        +  text
*        +  hypertext
*        +  lists
*        +  ruby
*        +  structural
*
```

```
*/
        </documentation>
    </annotation>
    <include schemaLocation="xhtml-core-1.xsd"/>
    <!-- end of REQUIRED modules -->
    <annotation>
        <documentation>
/**
* Optional Element modules
*
* Comment out those you don't need.
* Note that you will also need (probably)
* to modify the content models file.
*
*        +  Edit
*        +  Bdo
*        +  Presentational
*        +  Link
*        +  Meta
*        +  Base
*        +  Scripting
*        +  Style
*        +  Image
*        +  Client side image maps
*        +  Server side image maps
*        +  Param
*        +  Applet
*        +  Object
*        +  Tables
*        +  Forms
*        +  Nameident
*        +  Legacy (includes frames)
*        +  Basic forms
*        +  Basic tables
*
*/
        </documentation>
    </annotation>
    <include schemaLocation="xhtml-optional-1.xsd"/>
</schema>
```

D.3. XHTML SCHEMA Modular Framework

In order to take advantage of the XHTML Schema Modules, Schema authors need to define the content model for their language. XHTML provides a variety of tools to ease this effort. They are defined in a set of support modules, instantiated by a main Framework module:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1">

    <xsd:annotation>
        <xsd:documentation>
/**
```

```
* This is the XML Schema Framework module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//XHTML Modular Framework 1.0//EN"
*
*/
    </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
    </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
        <xsd:documentation>
/**
* Modular Framework
*
*     This required module instantiates the modules needed
*     to support the XHTML modularization model, including:
*
*        +  notations
*        +  datatypes
*        +  events
*        +  common attributes
*        +  common content models
*        +  character entities
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <!-- Required declarations for common notations -->

    <xsd:annotation>
        <xsd:documentation>
/**
* Notations module
*
* Contains XHTML notations for data types
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml-notations-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Datatypes module
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml-datatypes-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Events module
*
* Events attributes
```

```
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml-events-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Common attributes module
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml-attribs-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Content models module
*
* Common content models
*
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml11-model-1.xsd"/>

    <!-- Entities are broken in XML Schema -->

    <xsd:annotation>
        <xsd:documentation>
/**
* Character entities module
*
* Notation declarations for Latin 1, Special, and Symbol character entity sets
*/
    </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="framework/xhtml-charent-1.xsd"/>

    <!-- end of framework modules    -->
</xsd:schema>
```

Note that the module above references a content model module. This module is defined on a per-document type basis in addition to the document type driver file. The Modular framework also relies upon the following component modules:

D.3.1. XHTML Notations

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
    <xsd:documentation>
/**
* This is the XML Schema module for data type notations for XHTML
*  Please use this formal public identifier to identify it:
```

```
*          "-//W3C//NOTATIONS XHTML Notations 1.0//EN"
* And this namespace for XHTML elements:
*        xmlns:xhtml="http://www.w3.org/1999/xhtml"
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>


     <xsd:documentation>
/**
* Notations
*
*    defines the following notations, many of these imported from
*    other specifications and standards. When an existing FPI is
*    known, it is incorporated here.
*
*/
  </xsd:documentation>
</xsd:annotation>

    <!-- W3C XML 1.0 Recommendation -->
    <xsd:notation name="w3c-xml" public="ISO 8879//NOTATION Extensible Markup Language (XML) 1.0//EN"/>
    <!-- XML 1.0 CDATA -->
    <xsd:notation name="cdata" public="-//W3C//NOTATION XML 1.0: CDATA//EN"/>
    <!-- SGML Formal Public Identifiers -->
    <xsd:notation name="fpi" public="ISO 8879:1986//NOTATION Formal Public Identifier//EN"/>
    <!-- XHTML Notations ... -->
    <!-- Length defined for cellpadding/cellspacing -->
    <!-- nn for pixels or nn% for percentage length -->
    <!-- a single character from [ISO10646] -->
    <xsd:notation name="character" public="-//W3C//NOTATION XHTML Datatype: Character//EN"/>
    <!-- a character encoding, as per [RFC2045] -->
    <xsd:notation name="charset" public="-//W3C//NOTATION XHTML Datatype: Charset//EN"/>
    <!-- a space separated list of character encodings, as per [RFC2045] -->
    <xsd:notation name="charsets" public="-//W3C//NOTATION XHTML Datatype: Charsets//EN"/>
    <!-- media type, as per [RFC2045] -->
    <xsd:notation name="contentType" public="-//W3C//NOTATION XHTML Datatype: ContentType//EN"/>
    <!-- comma-separated list of media types, as per [RFC2045] -->
    <xsd:notation name="contentTypes" public="-//W3C//NOTATION XHTML Datatype: ContentTypes//EN"/>
    <!-- date and time information. ISO date format -->
    <xsd:notation name="datetime" public="-//W3C//NOTATION XHTML Datatype: Datetime//EN"/>
    <!-- a language code, as per [RFC3066] -->
    <xsd:notation name="languageCode" public="-//W3C//NOTATION XHTML Datatype: LanguageCode//EN"/>
    <!-- nn for pixels or nn% for percentage length -->
    <xsd:notation name="length" public="-//W3C//NOTATION XHTML Datatype: Length//EN"/>
    <!-- space-separated list of link types -->
    <xsd:notation name="linkTypes" public="-//W3C//NOTATION XHTML Datatype: LinkTypes//EN"/>
    <!-- single or comma-separated list of media descriptors -->
    <xsd:notation name="mediaDesc" public="-//W3C//NOTATION XHTML Datatype: MediaDesc//EN"/>
    <!-- pixel, percentage, or relative -->
    <xsd:notation name="multiLength" public="-//W3C//NOTATION XHTML Datatype: MultiLength//EN"/>
    <!-- one or more digits (NUMBER) -->
    <xsd:notation name="number" public="-//W3C//NOTATION XHTML Datatype: Number//EN"/>
    <!-- one or more digits (NUMBER) -->
    <xsd:notation name="pixels" public="-//W3C//NOTATION XHTML Datatype: Pixels//EN"/>
    <!-- script expression -->
```

```
    <xsd:notation name="script" public="-//W3C//NOTATION XHTML Datatype: Script//EN"/>
    <!-- textual content -->
    <xsd:notation name="text" public="-//W3C//NOTATION XHTML Datatype: Text//EN"/>
    <!-- a Uniform Resource Identifier, see [URI] -->
    <xsd:notation name="uri" public="-//W3C//NOTATION XHTML Datatype: URI//EN"/>
    <!-- a space-separated list of Uniform Resource Identifiers, see [URI] -->
    <xsd:notation name="uris" public="-//W3C//NOTATION XHTML Datatype: URIs//EN"/>

</xsd:schema>
```

D.3.2. XHTML Datatypes

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema datatypes module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//XHTML Datatypes 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
        <xsd:documentation>
/**
* Datatypes
*
*     defines containers for the following datatypes, many of
*     these imported from other specifications and standards.
*
*/
  </xsd:documentation>
    </xsd:annotation>
    <!-- Length defined for cellpadding/cellspacing -->
    <!-- nn for pixels or nn% for percentage length -->
    <xsd:simpleType name="Length">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- space-separated list of link types -->
    <xsd:simpleType name="LinkTypes">
        <xsd:list itemType="xsd:NMTOKEN"/>
    </xsd:simpleType>
    <!-- single or comma-separated list of media descriptors -->
    <xsd:simpleType name="MediaDesc">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- pixel, percentage, or relative -->
    <xsd:simpleType name="MultiLength">
        <xsd:restriction base="xsd:string"/>
```

```
    </xsd:simpleType>
    <!-- one or more digits (NUMBER) -->
    <xsd:simpleType name="Number">
        <xsd:restriction base="xsd:nonNegativeInteger"/>
    </xsd:simpleType>
    <!-- integer representing length in pixels -->
    <xsd:simpleType name="Pixels">
        <xsd:restriction base="xsd:nonNegativeInteger"/>
    </xsd:simpleType>
    <!-- script expression -->
    <xsd:simpleType name="Script">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- RGB color expression - this needs a better definition -->
    <xsd:simpleType name="Color">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- textual content -->
    <xsd:simpleType name="Text">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- Imported Datatypes  -->
    <!-- a single character from [ISO10646] -->
    <xsd:simpleType name="Character">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- a character encoding, as per [RFC2045] -->
    <xsd:simpleType name="Charset">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- a space separated list of character encodings, as per [RFC2045] -->
    <xsd:simpleType name="Charsets">
        <xsd:list itemType="xsd:string"/>
    </xsd:simpleType>
    <!-- media type, as per [RFC2045] -->
    <xsd:simpleType name="ContentType">
        <xsd:list itemType="xsd:string"/>
    </xsd:simpleType>
    <!-- comma-separated list of media types, as per [RFC2045] -->
    <xsd:simpleType name="ContentTypes">
        <xsd:list itemType="xsd:string"/>
    </xsd:simpleType>
    <!-- date and time information. ISO date format -->
    <xsd:simpleType name="Datetime">
        <xsd:restriction base="xsd:CDATA"/>
    </xsd:simpleType>
    <!-- formal public identifier, as per [ISO8879] -->
    <xsd:simpleType name="FPI">
        <xsd:restriction base="xsd:CDATA"/>
    </xsd:simpleType>
    <!-- a language code, as per [RFC3066] -->
    <xsd:simpleType name="LanguageCode">
        <xsd:restriction base="xsd:NMTOKEN"/>
    </xsd:simpleType>
    <!-- a Uniform Resource Identifier, see [URI] -->
    <xsd:simpleType name="URI">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <!-- a space-separated list of Uniform Resource Identifiers, see [URI] -->
    <xsd:simpleType name="URIs">
        <xsd:list itemType="xsd:string"/>
    </xsd:simpleType>
</xsd:schema>
```

D.3.3. XHTML Common Attribute Definitions

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
```

```
            version="1.1"
>

    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema common attributes module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//XHTML Common Attributes 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
        <xsd:documentation>
/**
*  Common Attributes
*    This module declares many of the common attributes for the XHTML Schema
*
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:attributeGroup name="id">
        <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="class">
        <xsd:attribute name="class" type="xsd:NMTOKENS"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="title">
        <xsd:attribute name="title" type="xsd:string"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="Core.extra.attrib">
        <!-- add your attribute here -->
    </xsd:attributeGroup>
    <xsd:attributeGroup name="Core.attrib">
        <xsd:attributeGroup ref="id"/>
        <xsd:attributeGroup ref="class"/>
        <xsd:attributeGroup ref="title"/>
        <xsd:attributeGroup ref="Core.extra.attrib"/>
    </xsd:attributeGroup>
    <xsd:attributeGroup name="BIDI"/>
    <xsd:attribute name="dir">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="ltr"/>
                <xsd:enumeration value="rtl"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attributeGroup name="I18n.attrib">
        <xsd:attributeGroup ref="BIDI"/>
        <xsd:attribute name="xml:lang" type="LanguageCode"/>
    </xsd:attributeGroup>
    <!-- intrinsic event attributes declared previously -->
    <xsd:attributeGroup name="Common.extra">
        <!-- add your attributes here -->
    </xsd:attributeGroup>
    <xsd:attributeGroup name="Common.attrib">
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
```

```
        <xsd:attributeGroup ref="Events.attrib"/>
        <xsd:attributeGroup ref="Common.extra"/>
    </xsd:attributeGroup>
</xsd:schema>
```

D.3.4. XHTML Character Entities

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Character Entities module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ENTITIES XHTML Character Entities 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
    Character Entities for XHTML

     This module declares the set of character entities for XHTML,
     including the Latin 1, Symbol and Special character collections.
*/
    </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!-- These are the entity sets for ISO Latin 1 characters for the XHTML -->
    <xsd:notation name="XHTMLLatin1"
                  id="xhtml-lat1"
                  public="-//W3C//ENTITIES Latin 1 for XHTML//EN"
                  system="http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent"/>
    <!-- These are the entity sets for special characters for the XHTML -->
    <xsd:notation name="XHTMLSpecial"
                  id="xhtml-special"
                  public="-//W3C//ENTITIES Special for XHTML//EN"
                  system="http://www.w3.org/TR/xhtml1/DTD/xhtml-special.ent"/>
    <!-- These are the entity sets for symbol characters for the XHTML -->
    <xsd:notation name="XHTMLSymbol"
                  id="xhtml-symbol"
                  public="-//W3C//ENTITIES Symbols for XHTML//EN"
                  system="http://www.w3.org/TR/xhtml1/DTD/xhtml-symbol.ent"/>

</xsd:schema>
```

D.4. XHTML Module Implementations

This section contains the formal definition of each of the XHTML Abstract Modules as a DTD module.

D.4.1. XHTML Core Modules

D.4.1.1. Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Document Structure module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Document Structure 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

        <xsd:documentation>
/**
* Document Structure
*
*        title, head, body, html
*
*     The Structure Module defines the major structural elements and
*     their attributes.
*
*     Note that the content model of the head element type is redeclared
*     when the Base Module is included in the DTD.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

    </xsd:annotation>

    <!-- title: Document Title  -->
    <!--
        The title element is not considered part of the flow of text.
        It should be displayed, for example as the page header or
        window title. Exactly one title is required per document.
    -->
```

```
    <xsd:attributeGroup name="title.attlist">
        <xsd:attributeGroup ref="I18n.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="title.type" mixed="true">
        <xsd:attributeGroup ref="title.attlist"/>
    </xsd:complexType>
    <xsd:element name="title" type="title.type"/>
    <!-- head: Document Head -->
    <xsd:group name="head.content">
        <xsd:sequence>
            <xsd:group ref="HeadOpts.mix"/>
            <xsd:element ref="title" minOccurs="1"/>
            <xsd:group ref="HeadOpts.mix"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="head.attlist">
        <xsd:attribute name="profile" type="URI"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
    </xsd:attributeGroup>
    <xsd:complexType name="head.type">
        <xsd:group ref="head.content"/>
        <xsd:attributeGroup ref="head.attlist"/>
    </xsd:complexType>
    <xsd:element name="head" type="head.type"/>
    <!-- body: Document Body  -->
    <xsd:attributeGroup name="body.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="body.type" mixed="true">
        <xsd:group ref="Block.mix" minOccurs="1"/>
        <xsd:attributeGroup ref="body.attlist"/>
    </xsd:complexType>
    <xsd:element name="body" type="body.type"/>
    <!-- html: XHTML Document Element  -->
    <!-- check value -->
    <xsd:attributeGroup name="html.attlist">
        <xsd:attribute name="version" type="FPI" value="XHTML1.1"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
    </xsd:attributeGroup>
    <xsd:complexType name="html.type">
        <xsd:sequence>
            <xsd:element ref="head"/>
            <xsd:element ref="body"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="html.attlist"/>
    </xsd:complexType>
    <xsd:element name="html" type="html.type"/>
</xsd:schema>
```

D.4.1.2. Text

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Text  module for XHTML
* This is a REQUIRED module.
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Text 1.0//EN"
*
*/
        </xsd:documentation>
```

```
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
        </xsd:documentation>

        <xsd:documentation>
/**
* Textual Content
*
*    The Text module includes declarations for all core
*     text container elements and their attributes.
*
*        +  block phrasal
*        +  block structural
*        +  inline phrasal
*        +  inline structural
*
*/
        </xsd:documentation>

    <xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

<xsd:annotation>
        <xsd:documentation>
/**
* Block Phrasal module
* Elements defined here:
*        address, blockquote, pre, h1, h2, h3, h4, h5, h6
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="text/xhtml-blkphras-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Block Structural module
* Elements defined here:
*        div, p
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="text/xhtml-blkstruct-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Inline Phrasal module
* Elements defined here:
*        abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="text/xhtml-inlphras-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Inline Structural module
* Elements defined here:
```

```
*        br,span
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="text/xhtml-inlstruct-1.xsd"/>

</xsd:schema>
```

D.4.1.3. Hypertext

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Hypertext module for XHTML
*  This is a REQUIRED module.
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Hypertext 1.0//EN"
*
*/
  </xsd:documentation>

        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

        <xsd:documentation>
/**
*
* Hypertext
*
*        a
*
*     This module declares the anchor ('a') element type, which
*     defines the source of a hypertext link. The destination
*     (or link 'target') is identified via its 'id' attribute
*     rather than the 'name' attribute as was used in HTML.
*
*/
  </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

    </xsd:annotation>

    <!-- a -->
    <xsd:attributeGroup name="a.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="href" type="URI"/>
        <xsd:attribute name="charset" type="Charset"/>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:attribute name="hreflang" type="LanguageCode"/>
```

```
        <xsd:attribute name="rel" type="LinkTypes"/>
        <xsd:attribute name="rev" type="LinkTypes"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:attribute name="tabindex" type="Number"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="a.type" mixed="true">
        <xsd:group ref="InlNoAnchor.mix"/>
        <xsd:attributeGroup ref="a.attlist"/>
    </xsd:complexType>
    <xsd:element name="a" type="a.type"/>
</xsd:schema>
```

D.4.1.4. Lists

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Lists module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Lists 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Lists
*
*        dl, dt, dd, ol, ul, li
*
*     This module declares the list-oriented element types
*     and their attributes.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!-- definition lists -->
    <!-- dt: Definition Term  -->
    <xsd:attributeGroup name="dt.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
```

```
<xsd:complexType name="dt.type" mixed="true">
    <xsd:group ref="Inline.mix"/>
    <xsd:attributeGroup ref="dt.attlist"/>
</xsd:complexType>

<xsd:element name="dt" type="dt.type"/>

<!-- dd: Definition Description  -->
<xsd:attributeGroup name="dd.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="dd.type" mixed="true">
    <xsd:group ref="Flow.mix"/>
    <xsd:attributeGroup ref="dd.attlist"/>
</xsd:complexType>

<xsd:element name="dd" type="dd.type"/>

<!-- dl: Definition List  -->
<xsd:attributeGroup name="dl.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- content model? -->
<xsd:group name="dl.content">
    <xsd:sequence minOccurs="1" maxOccurs="unbounded">
        <xsd:element ref="dt"/>
        <xsd:element ref="dd"/>
    </xsd:sequence>
</xsd:group>

<xsd:complexType name="dl.type" mixed="true">
    <xsd:group ref="dl.content"/>
    <xsd:attributeGroup ref="dl.attlist"/>
</xsd:complexType>

<xsd:element name="dl" type="dl.type"/>

<!-- li: List Item -->
<xsd:attributeGroup name="li.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="li.type" mixed="true">
    <xsd:group ref="Flow.mix"/>
    <xsd:attributeGroup ref="li.attlist"/>
</xsd:complexType>

<xsd:element name="li" type="li.type"/>

<!-- ol: Ordered List (numbered styles) -->
<xsd:attributeGroup name="ol.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="ol.type">
    <xsd:sequence minOccurs="1">
        <xsd:element ref="li"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="ol.attlist"/>
</xsd:complexType>

<xsd:element name="ol" type="ol.type"/>

<!-- ul: Unordered List -->
<xsd:attributeGroup name="ul.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
```

```
    <xsd:complexType name="ul.type">
        <xsd:sequence minOccurs="1">
            <xsd:element ref="li"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="ul.attlist"/>
    </xsd:complexType>

    <xsd:element name="ul" type="ul.type"/>

</xsd:schema>
```

D.4.2. Text Modules

D.4.2.1. Presentation

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Presentation module for XHTML
* This is a REQUIRED module.
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Presentation 1.0//EN"
*
*/
        </xsd:documentation>

        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

        <xsd:documentation>
/**
* Presentational Elements
*
*    This module defines elements and their attributes for
*     simple presentation-related markup.
*Elements defined here:
* hr
* b, big, i, small, sub, sup, tt
*
*/
        </xsd:documentation>
    </xsd:annotation>

    <xsd:annotation>
        <xsd:documentation>
/**
* Block Presentational module
* Elements defined here:
```

```
*        hr
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="pres/xhtml-blkpres-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Inline Presentational module
* Elements defined here:
*        b, big, i, small, sub, sup, tt
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="pres/xhtml-inlpres-1.xsd"/>
</xsd:schema>
```

D.4.2.2. Edit

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Editing Markup module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ELEMENTS XHTML Editing Markup 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*   Editing Elements
*
*        ins, del
*
*     This module declares element types and attributes used to indicate
*     inserted and deleted content while editing a document.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <xsd:annotation>
        <xsd:documentation>
/**
* Both of these elements have the same content model and attribute list
*
*/
  </xsd:documentation>
```

```
    </xsd:annotation>
    <xsd:attributeGroup name="edit.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="cite" type="URI"/>
        <xsd:attribute name="datetime" type="Datetime"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="edit.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="Flow.mix"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="edit.type" mixed="true">
        <xsd:group ref="edit.content"/>
        <xsd:attributeGroup ref="edit.attlist"/>
    </xsd:complexType>
    <!-- ins: Inserted Text   -->
    <xsd:element name="ins" type="edit.type"/>
    <!-- del: Deleted Text   -->
    <xsd:element name="del" type="edit.type"/>
</xsd:schema>
```

D.4.2.3. Bi-directional Text

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema BDO Element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML BDO Element 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Bidirectional Override (bdo) Element
*
*     This modules declares the element 'bdo', used to override the
*     Unicode bidirectional algorithm for selected fragments of text.*
*
*     Bidirectional text support includes both the bdo
*     element and the 'dir' attribute.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <xsd:attributeGroup name="bdo.attlist">
        <xsd:attribute name="xml:lang" type="LanguageCode"/>
```

```
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:attribute name="dir" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="ltr"/>
                    <xsd:enumeration value="rtl"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="bdo.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="bdo.attlist"/>
    </xsd:complexType>
    <xsd:element name="bdo" type="bdo.type"/>
</xsd:schema>
```

D.4.3. Forms

D.4.3.1. Basic Forms

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Basic Forms module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Basic Forms 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Basic Forms
*
*     This forms module is based on the HTML 3.2 forms model, with
*     the WAI-requested addition of the label element. While this
*     module essentially mimics the content model and attributes of
*     HTML 3.2 forms, the element types declared herein also include
*     all HTML 4 common attributes.
*
*       Elements defined here:
*        form, label, input, select, option, textarea
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>
```

```
</xsd:annotation>
    <!--
        BlkNoForm.mix includes all non-form block elements,
        plus Misc.class
    -->
    <xsd:group name="BlkNoForm.mix">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="Heading.class"/>
            <xsd:group ref="List.class"/>
            <xsd:group ref="BlkStruct.class"/>
            <xsd:group ref="BlkPhras.class"/>
            <xsd:group ref="BlkPres.class"/>
            <xsd:group ref="Table.class"/>
            <xsd:group ref="Block.extra"/>
            <xsd:group ref="Misc.class"/>
        </xsd:choice>
    </xsd:group>
    <!-- form -->
    <xsd:attributeGroup name="form.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="action" type="URI"/>
        <xsd:attribute name="method" use="default" value="get">
            <xsd:simpleType>
                <xsd:restriction base="NMTOKEN">
                    <xsd:enumeration value="get"/>
                    <xsd:enumeration value="post"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="enctype" type="ContentType" use="fixed" value="application/x-www-form-urlencoded"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="form.type">
        <xsd:group ref="BlkNoForm.mix"/>
        <xsd:attributeGroup ref="form.attlist"/>
    </xsd:complexType>
    <xsd:element name="form" type="form.type"/>
    <!-- label: Form Field Label Text -->
    <!--
        Each label must not contain more than ONE field
    -->
    <xsd:group name="label.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="input"/>
            <xsd:element ref="select"/>
            <xsd:element ref="textarea"/>
            <xsd:group ref="InlStruct.class.class"/>
            <xsd:group ref="InlPhras.class"/>
            <xsd:group ref="I18n.class"/>
            <xsd:group ref="InlPres.class"/>
            <xsd:group ref="InlSpecial.class"/>
            <xsd:group ref="Misc.class"/>
        </xsd:choice>
    </xsd:group>
    <xsd:attributeGroup name="label.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="for" type="Text"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="label.type" mixed="true">
        <xsd:group ref="label.content"/>
        <xsd:attributeGroup ref="label.attlist"/>
    </xsd:complexType>
    <xsd:element name="label" type="label.type"/>
    <!-- input: Form Control -->
    <!--
        attribute name required for all but submit & reset
    -->
    <!--
        Basic Forms removes 'image' and 'file' input types.
    -->
    <xsd:attribute name="type" use="default" value="text">
        <xsd:simpleType>
            <xsd:restriction base="NMTOKEN">
```

```
                <xsd:enumeration value="text"/>
                <xsd:enumeration value="password"/>
                <xsd:enumeration value="checkbox"/>
                <xsd:enumeration value="radio"/>
                <xsd:enumeration value="submit"/>
                <xsd:enumeration value="reset"/>
                <xsd:enumeration value="hidden"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attributeGroup name="input.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute ref="type"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="value" type="Text"/>
        <xsd:attribute name="checked" type="Text"/>
        <xsd:attribute name="size" type="Number"/>
        <xsd:attribute name="maxlength" type="Number"/>
        <xsd:attribute name="src" type="URI"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="input.type">
        <xsd:attributeGroup ref="input.attlist"/>
    </xsd:complexType>
    <xsd:element name="input" type="input.type"/>
    <!-- select: Option Selector  -->
    <xsd:attributeGroup name="select.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="size" type="Number"/>
        <xsd:attribute name="multiple" type="Text"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="select.type">
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="option"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="select.attlist"/>
    </xsd:complexType>
    <xsd:element name="select" type="select.type"/>
    <!-- option: Selectable Choice  -->
    <xsd:attributeGroup name="option.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="selected" type="Text"/>
        <xsd:attribute name="value" type="Text"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="option.type" mixed="true">
        <xsd:attributeGroup ref="option.attlist"/>
    </xsd:complexType>
    <xsd:element name="option" type="option.type"/>
    <!-- textarea: Multi-Line Text Field -->
    <xsd:attributeGroup name="textarea.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="rows" type="Number"/>
        <xsd:attribute name="cols" type="Number"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="textarea.type" mixed="true">
        <xsd:attributeGroup ref="textarea.attlist"/>
    </xsd:complexType>
    <xsd:element name="textarea" type="textarea.type"/>
</xsd:schema>
```

D.4.3.2. Forms

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Forms module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Forms 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Forms
*
*        form, label, input, select, optgroup, option,
*        textarea, fieldset, legend, button
*
*     This module declares markup to provide support for online
*     forms, based on the features found in HTML 4.0 forms.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- note that table.class was omitted from the DTD -->
    <xsd:group name="Table.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="table"/>
        </xsd:choice>
    </xsd:group>
    <!--
        BlkNoForm.mix includes all non-form block elements,
        plus Misc.class
    -->
    <xsd:group name="BlkNoForm.mix">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="Heading.class"/>
            <xsd:group ref="List.class"/>
            <xsd:group ref="BlkStruct.class"/>
            <xsd:group ref="BlkPhras.class"/>
            <xsd:group ref="BlkPres.class"/>
            <xsd:group ref="Table.class"/>
            <xsd:group ref="Block.extra"/>
            <xsd:group ref="Misc.class"/>
        </xsd:choice>
    </xsd:group>
    <!-- form: Form Element -->
    <xsd:attributeGroup name="form.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="action" type="URI"/>
        <xsd:attribute name="method" use="default" value="get">
```

```
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="get"/>
                <xsd:enumeration value="post"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="enctype" type="ContentType" use="fixed" value="application/x-www-form-urlencoded"/>
    <xsd:attribute name="accept-charset" type="Charsets"/>
    <xsd:attribute name="accept" type="ContentTypes"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="form.content">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:group ref="BlkNoForm.mix"/>
        <xsd:element ref="fieldset"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="form.type">
    <xsd:group ref="form.content"/>
    <xsd:attributeGroup ref="form.attlist"/>
</xsd:complexType>
<xsd:element name="form" type="form.type"/>
<!-- label: Form Field Label Text  -->
<!--
    Each label must not contain more than ONE field
-->
<xsd:group name="label.content">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="input"/>
        <xsd:element ref="select"/>
        <xsd:element ref="textarea"/>
        <xsd:element ref="button"/>
        <xsd:group ref="InlStruct.class"/>
        <xsd:group ref="InlPhras.class"/>
        <xsd:group ref="I18n.class"/>
        <xsd:group ref="InlPres.class"/>
        <xsd:group ref="InlSpecial.class"/>
        <xsd:group ref="Inline.extra"/>
        <xsd:group ref="Misc.class"/>
    </xsd:choice>
</xsd:group>
<xsd:attributeGroup name="label.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="for" type="xsd:IDREF"/>
    <xsd:attribute name="accesskey" type="Character"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="label.type" mixed="true">
    <xsd:group ref="label.content"/>
    <xsd:attributeGroup ref="label.attlist"/>
</xsd:complexType>
<xsd:element name="label" type="label.type"/>
<!-- input: Form Control -->
<xsd:attribute name="type" use="default" value="text">
    <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="password"/>
            <xsd:enumeration value="checkbox"/>
            <xsd:enumeration value="radio"/>
            <xsd:enumeration value="submit"/>
            <xsd:enumeration value="reset"/>
            <xsd:enumeration value="hidden"/>
            <xsd:enumeration value="image"/>
            <xsd:enumeration value="button"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<!--
    attribute 'name' required for all but submit & reset
-->
<xsd:attributeGroup name="input.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute ref="type"/>
    <xsd:attribute name="name" type="Text"/>
```

```
    <xsd:attribute name="value" type="Text"/>
    <xsd:attribute name="checked" type="Text"/>
    <xsd:attribute name="disabled" type="Text"/>
    <xsd:attribute name="readonly" type="Text"/>
    <xsd:attribute name="size" type="Number"/>
    <xsd:attribute name="maxlength" type="Number"/>
    <xsd:attribute name="src" type="URI"/>
    <xsd:attribute name="alt" type="Text"/>
    <xsd:attribute name="tabindex" type="Number"/>
    <xsd:attribute name="accesskey" type="Character"/>
    <xsd:attribute name="accept" type="ContentTypes"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- input is EMPTY -->
<xsd:complexType name="input.type">
    <xsd:attributeGroup ref="input.attlist"/>
</xsd:complexType>
<xsd:element name="input" type="input.type"/>
<!-- select: Option Selector  -->
<xsd:attributeGroup name="select.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="name" type="Text"/>
    <xsd:attribute name="size" type="Number"/>
    <xsd:attribute name="multiple" type="Text"/>
    <xsd:attribute name="disabled" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="select.content">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element ref="optgroup"/>
        <xsd:element ref="option"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="select.type">
    <xsd:group ref="select.content"/>
    <xsd:attributeGroup ref="select.attlist"/>
</xsd:complexType>
<xsd:element name="select" type="select.type"/>
<!-- optgroup: Option Group  -->
<xsd:attributeGroup name="optgroup.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="disabled" type="Text"/>
    <xsd:attribute name="label" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="optgroup.content">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element ref="option"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="optgroup.type">
    <xsd:group ref="optgroup.content"/>
    <xsd:attributeGroup ref="select.attlist"/>
</xsd:complexType>
<xsd:element name="optgroup" type="optgroup.type"/>
<!-- option: Selectable Choice  -->
<xsd:attributeGroup name="option.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="selected" type="Text"/>
    <xsd:attribute name="disabled" type="Text"/>
    <xsd:attribute name="label" type="Text"/>
    <xsd:attribute name="value" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="option.type" mixed="true">
    <xsd:attributeGroup ref="option.attlist"/>
</xsd:complexType>
<xsd:element name="option" type="option.type"/>
<!-- textarea: Multi-Line Text Field  -->
<xsd:attributeGroup name="textarea.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="name" type="Text"/>
    <xsd:attribute name="rows" type="Number"/>
    <xsd:attribute name="cols" type="Number"/>
    <xsd:attribute name="disabled" type="Text"/>
    <xsd:attribute name="readonly" type="Text"/>
```

```
        <xsd:attribute name="tabindex" type="Number"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="textarea.type" mixed="true">
        <xsd:attributeGroup ref="textarea.attlist"/>
    </xsd:complexType>
    <xsd:element name="textarea" type="textarea.type"/>
    <!-- fieldset: Form Control Group  -->
    <!--
        #PCDATA is to solve the mixed content problem,
        per specification only whitespace is allowed
    -->
    <xsd:attributeGroup name="fieldset.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="fieldset.type" mixed="true">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="legend"/>
            <xsd:group ref="Flow.mix"/>
        </xsd:choice>
        <xsd:attributeGroup ref="fieldset.attlist"/>
    </xsd:complexType>
    <xsd:element name="fieldset" type="fieldset.type"/>
    <!-- legend: Fieldset Legend  -->
    <xsd:attributeGroup name="legend.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="legend.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="legend.attlist"/>
    </xsd:complexType>
    <xsd:element name="legend" type="legend.type"/>
    <!-- button: Push Button  -->
    <xsd:attributeGroup name="button.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="value" type="Text"/>
        <xsd:attribute name="type" type="Number"/>
        <xsd:attribute name="type" use="default" value="submit">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="button"/>
                    <xsd:enumeration value="submit"/>
                    <xsd:enumeration value="reset"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="disabled" type="Text"/>
        <xsd:attribute name="tabindex" type="Number"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="button.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="BlkNoForm.mix"/>
            <xsd:group ref="InlStruct.class"/>
            <xsd:group ref="InlPhras.class"/>
            <xsd:group ref="InlPres.class"/>
            <xsd:group ref="BlkPres.class"/>
            <xsd:group ref="I18n.class"/>
            <xsd:group ref="InlSpecial.class"/>
            <xsd:group ref="Inline.extra"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="button.type" mixed="true">
        <xsd:group ref="button.content"/>
        <xsd:attributeGroup ref="button.attlist"/>
    </xsd:complexType>
    <xsd:element name="button" type="button.type"/>
</xsd:schema>
```

D.4.4. Tables

D.4.4.1. Basic Tables

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Basic Tables module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Basic Tables 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Basic Tables
*
*        table, caption, tr, th, td
*
*     This table module declares elements and attributes defining
*     a table model based fundamentally on features found in the
*     widely-deployed HTML 3.2 table model.  While this module
*     mimics the content model and table attributes of HTML 3.2
*     tables, the element types declared herein also includes all
*     HTML 4 common and most of the HTML 4 table attributes.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!--
        horizontal alignment attributes for cell contents
    -->

    <xsd:attributeGroup name="CellHAlign.attrib">
        <xsd:attribute name="align">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="left"/>
                    <xsd:enumeration value="center"/>
                    <xsd:enumeration value="right"/>
                </xsd:restriction>
            </xsd:simpleType>
```

```
    </xsd:attribute>
</xsd:attributeGroup>

<!--
    vertical alignment attributes for cell contents
-->

<xsd:attributeGroup name="CellVAlign.attrib">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="top"/>
                <xsd:enumeration value="middle"/>
                <xsd:enumeration value="bottom"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>

<!--
    scope is simpler than axes attribute for common tables
-->

<xsd:attributeGroup name="scope.attrib">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="row"/>
                <xsd:enumeration value="col"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>

<!-- table: Table Element  -->

<!-- table: Table Element  -->
<xsd:attributeGroup name="table.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="summary" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:group name="table.content">
    <xsd:sequence>
        <xsd:element ref="caption" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="tr" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:group>

<xsd:complexType name="table.type" >
    <xsd:group ref="table.content"/>
    <xsd:attributeGroup ref="table.attlist"/>
</xsd:complexType>
<xsd:element name="table" type="table.type"/>

<!-- caption: Table Caption -->

<xsd:attributeGroup name="caption.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="summary" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="caption.type" mixed="true" >
    <xsd:group ref="Inline.mix"/>
    <xsd:attributeGroup ref="caption.attlist"/>
</xsd:complexType>

<xsd:element name="caption" type="caption.type"/>

<!-- tr: Table Row  -->

<xsd:attributeGroup name="tr.attlist">
```

```
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attributeGroup ref="CellHAlign.attrib"/>
        <xsd:attributeGroup ref="CellVAlign.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="tr.content">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element ref="th"/>
            <xsd:element ref="td"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="tr.type">
        <xsd:group ref="tr.content"/>
        <xsd:attributeGroup ref="tr.attlist"/>
    </xsd:complexType>
    <xsd:element name="tr" type="tr.type"/>

    <!-- th: Table Header Cell  -->
    <!--
        th is for header cells, td for data,
        but for cells acting as both use td
    -->
    <xsd:attributeGroup name="th.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="abbr" type="Text"/>
        <xsd:attribute name="axis" type="Text"/>
        <xsd:attribute name="headers" type="xsd:IDREFS"/>
        <xsd:attributeGroup ref="scope.attrib"/>
        <xsd:attribute name="rowspan" type="Number" use="default" value="1"/>
        <xsd:attribute name="colspan" type="Number" use="default" value="1"/>
        <xsd:attributeGroup ref="CellHAlign.attrib"/>
        <xsd:attributeGroup ref="CellVAlign.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="th.type" mixed="true">
        <xsd:group ref="FlowNoTable.mix"/>
        <xsd:attributeGroup ref="th.attlist"/>
    </xsd:complexType>
    <xsd:element name="th" type="th.type"/>

    <!-- td: Table Data Cell ... -->

    <!-- td: Table Data Cell -->
    <xsd:attributeGroup name="td.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="abbr" type="Text"/>
        <xsd:attribute name="axis" type="Text"/>
        <xsd:attribute name="headers" type="xsd:IDREFS"/>
        <xsd:attributeGroup ref="scope.attrib"/>
        <xsd:attribute name="rowspan" type="Number" use="default" value="1"/>
        <xsd:attribute name="colspan" type="Number" use="default" value="1"/>
        <xsd:attributeGroup ref="CellHAlign.attrib"/>
        <xsd:attributeGroup ref="CellVAlign.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="td.type" mixed="true">
        <xsd:group ref="FlowNoTable.mix"/>
        <xsd:attributeGroup ref="td.attlist"/>
    </xsd:complexType>
    <xsd:element name="td" type="td.type"/>

</xsd:schema>
```

D.4.4.2. Tables

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
```

```
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Tables module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Tables 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Tables
*
*        table, caption, thead, tfoot, tbody, colgroup, col, tr, th, td
*
*     This module declares element types and attributes used to provide
*     table markup similar to HTML 4.0, including features that enable
*     better accessibility for non-visual user agents.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- attribute groups -->
    <!--
        The frame attribute specifies which parts of the frame around
        the table should be rendered. The values are not the same as
        CALS to avoid a name clash with the valign attribute.
    -->
    <xsd:attributeGroup name="frame.attrib">
        <xsd:attribute name="frame">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="void"/>
                    <xsd:enumeration value="above"/>
                    <xsd:enumeration value="below"/>
                    <xsd:enumeration value="hsides"/>
                    <xsd:enumeration value="lhs"/>
                    <xsd:enumeration value="rhs"/>
                    <xsd:enumeration value="vsides"/>
                    <xsd:enumeration value="box"/>
                    <xsd:enumeration value="border"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:attributeGroup>
    <!--
        The rules attribute defines which rules to draw between cells:
        If rules is absent then assume:
       "none" if border is absent or border="0" otherwise "all"
    -->
    <xsd:attributeGroup name="rules.attrib">
        <xsd:attribute name="rules">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="none"/>
                    <xsd:enumeration value="groups"/>
                    <xsd:enumeration value="rows"/>
                    <xsd:enumeration value="cols"/>
```

```
                <xsd:enumeration value="all"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>
<!--
    vertical alignment attribute for cell contents
-->
<xsd:attributeGroup name="CellVAlign.attrib">
    <xsd:attribute name="valign">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="top"/>
                <xsd:enumeration value="middle"/>
                <xsd:enumeration value="bottom"/>
                <xsd:enumeration value="baseline"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>
<!--
    horizontal alignment attributes for cell contents
-->
<xsd:attributeGroup name="CellHAlign.attrib">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="left"/>
                <xsd:enumeration value="center"/>
                <xsd:enumeration value="right"/>
                <xsd:enumeration value="justify"/>
                <xsd:enumeration value="char"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="char" type="Character"/>
    <xsd:attribute name="charoff" type="Length"/>
</xsd:attributeGroup>
<!--
    scope is simpler than axes attribute for common tables
-->
<xsd:attributeGroup name="scope.attrib">
    <xsd:attribute name="scope">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="row"/>
                <xsd:enumeration value="col"/>
                <xsd:enumeration value="rowgroup"/>
                <xsd:enumeration value="colgroup"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>
<!--
    Note that these elements are not defined in the same order
    as in the DTD
-->
<!-- td: Table Data Cell -->
<xsd:attributeGroup name="td.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="abbr" type="Text"/>
    <xsd:attribute name="axis" type="Text"/>
    <xsd:attribute name="headers" type="xsd:IDREFS"/>
    <xsd:attributeGroup ref="scope.attrib"/>
    <xsd:attribute name="rowspan" type="Number" use="default" value="1"/>
    <xsd:attribute name="colspan" type="Number" use="default" value="1"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="td.type" mixed="true">
    <xsd:group ref="Flow.mix"/>
    <xsd:attributeGroup ref="td.attlist"/>
</xsd:complexType>
```

```
<xsd:element name="td" type="td.type"/>
<!-- th: Table Header Cell  -->
<!--
    th is for header cells, td for data,
    but for cells acting as both use td
-->
<xsd:attributeGroup name="th.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="abbr" type="Text"/>
    <xsd:attribute name="axis" type="Text"/>
    <xsd:attribute name="headers" type="xsd:IDREFS"/>
    <xsd:attributeGroup ref="scope.attrib"/>
    <xsd:attribute name="rowspan" type="Number" use="default" value="1"/>
    <xsd:attribute name="colspan" type="Number" use="default" value="1"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="th.type" mixed="true">
    <xsd:group ref="Flow.mix"/>
    <xsd:attributeGroup ref="th.attlist"/>
</xsd:complexType>
<xsd:element name="th" type="th.type"/>
<!-- tr: Table Row  -->
<xsd:attributeGroup name="tr.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="tr.content">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element ref="th"/>
        <xsd:element ref="td"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="tr.type">
    <xsd:group ref="tr.content"/>
    <xsd:attributeGroup ref="tr.attlist"/>
</xsd:complexType>
<xsd:element name="tr" type="tr.type"/>
<!-- col: Table Column  -->
<!--
    col elements define the alignment properties for
     cells in one or more columns.

     The width attribute specifies the width of the
     columns, e.g.

       width="64"        width in screen pixels
       width="0.5*"      relative width of 0.5

     The span attribute causes the attributes of one
     col element to apply to more than one column.
-->
<xsd:attributeGroup name="col.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="span" type="Number" use="default" value="1"/>
    <xsd:attribute name="width" type="MultiLength"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- col is EMPTY -->
<xsd:complexType name="col.type">
    <xsd:attributeGroup ref="col.attlist"/>
</xsd:complexType>
<xsd:element name="col" type="col.type"/>
<!-- colgroup: Table Column Group -->
<!--
    colgroup groups a set of col elements. It allows you
    to group several semantically-related columns together.
-->
<xsd:attributeGroup name="colgroup.attlist">
```

```
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="span" type="Number" use="default" value="1"/>
    <xsd:attribute name="width" type="MultiLength"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="colgroup.content">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="col"/>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="colgroup.type">
    <xsd:group ref="colgroup.content"/>
    <xsd:attributeGroup ref="colgroup.attlist"/>
</xsd:complexType>
<xsd:element name="colgroup" type="colgroup.type"/>
<!-- tbody: Table Body  -->
<!--
    Use multiple tbody sections when rules are needed
    between groups of table rows.
-->
<xsd:attributeGroup name="tbody.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="tbody.content">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="tr"/>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="tbody.type">
    <xsd:group ref="tbody.content"/>
    <xsd:attributeGroup ref="tbody.attlist"/>
</xsd:complexType>
<xsd:element name="tbody" type="tbody.type"/>
<!-- tfoot: Table Footer  -->
<!--
    Use tfoot to duplicate footers when breaking table
    across page boundaries, or for static footers when
    tbody sections are rendered in scrolling panel.
-->
<xsd:attributeGroup name="tfoot.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="tfoot.content">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="tr"/>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="tfoot.type">
    <xsd:group ref="tfoot.content"/>
    <xsd:attributeGroup ref="tfoot.attlist"/>
</xsd:complexType>
<xsd:element name="tfoot" type="tfoot.type"/>
<!-- thead: Table Header  -->
<!--
    Use thead to duplicate headers when breaking table
    across page boundaries, or for static headers when
    tbody sections are rendered in scrolling panel.
-->
<xsd:attributeGroup name="thead.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attributeGroup ref="CellHAlign.attrib"/>
    <xsd:attributeGroup ref="CellVAlign.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="thead.content">
    <xsd:sequence maxOccurs="unbounded">
```

```
            <xsd:element ref="tr"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="thead.type">
        <xsd:group ref="thead.content"/>
        <xsd:attributeGroup ref="thead.attlist"/>
    </xsd:complexType>
    <xsd:element name="thead" type="thead.type"/>
    <!-- caption: Table Caption  -->
    <xsd:attributeGroup name="caption.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="caption.content">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element ref="tr"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="caption.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="caption.attlist"/>
    </xsd:complexType>
    <xsd:element name="caption" type="caption.type"/>
    <!-- table: Table Element  -->
    <xsd:attributeGroup name="table.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="summary" type="Text"/>
        <xsd:attribute name="width" type="Length"/>
        <xsd:attribute name="border" type="Pixels"/>
        <xsd:attributeGroup ref="frame.attrib"/>
        <xsd:attributeGroup ref="rules.attrib"/>
        <xsd:attribute name="cellspacing" type="Length"/>
        <xsd:attribute name="cellpadding" type="Length"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="table.content">
        <xsd:sequence>
            <xsd:element ref="caption" minOccurs="0"/>
            <xsd:choice>
                <xsd:element ref="col" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="colgroup" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:choice>
            <xsd:choice>
                <xsd:sequence>
                    <xsd:element ref="thead" minOccurs="0"/>
                    <xsd:element ref="tfoot" minOccurs="0"/>
                    <xsd:element ref="tbody" maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:choice>
                    <xsd:element ref="tr" maxOccurs="unbounded"/>
                </xsd:choice>
            </xsd:choice>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="table.type" mixed="true">
        <xsd:group ref="table.content"/>
        <xsd:attributeGroup ref="table.attlist"/>
    </xsd:complexType>
    <xsd:element name="table" type="table.type"/>
</xsd:schema>
```

D.4.5. Image

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
```

```
     <xsd:documentation>
/**
* This is the XML Schema Images module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Images 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Images
*
*        img
*
*     This module provides markup to support basic image embedding.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!--
        To avoid problems with text-only UAs as well as to make
        image content understandable and navigable to users of
        non-visual UAs, you need to provide a description with
        the 'alt' attribute, and avoid server-side image maps.
    -->
    <xsd:attributeGroup name="image.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="src" type="URI" use="required"/>
        <xsd:attribute name="alt" type="Text" use="required"/>
        <xsd:attribute name="longdesc" type="URI"/>
        <xsd:attribute name="height" type="Length"/>
        <xsd:attribute name="width" type="Length"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>

    <!-- img is EMPTY -->
    <xsd:complexType name="image.type">
        <xsd:attributeGroup ref="image.attlist"/>
    </xsd:complexType>

    <xsd:element name="img" type="image.type"/>
</xsd:schema>
```

D.4.6. Client-side Image Map

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
```

```
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Client-side Image Maps module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ELEMENTS XHTML Client-side Image Maps 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Client-side Image Maps
*
*        area, map
*
*     This module declares elements and attributes to support client-side
*     image maps. This requires that the Image Module (or a module
*     declaring the img element type) be included in the DTD.
*
*     These can be placed in the same document or grouped in a
*     separate document, although the latter isn't widely supported
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- area -->
    <xsd:attribute name="shape" use="default" value="rect">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="rect"/>
                <xsd:enumeration value="circle"/>
                <xsd:enumeration value="poly"/>
                <xsd:enumeration value="default"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attributeGroup name="area.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="href" type="URI"/>
        <xsd:attribute ref="shape"/>
        <xsd:attribute name="coords" type="Text"/>
        <xsd:attribute name="nohref" type="Text"/>
        <xsd:attribute name="alt" type="Text"/>
        <xsd:attribute name="tabindex" type="Number"/>
        <xsd:attribute name="accesskey" type="Character"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <!-- area is EMPTY -->
    <xsd:complexType name="area.type">
        <xsd:attributeGroup ref="area.attlist"/>
    </xsd:complexType>
    <xsd:element name="area" type="area.type"/>
    <!-- map -->
    <xsd:attributeGroup name="map.attlist">
        <xsd:attributeGroup ref="id"/>
        <xsd:attributeGroup ref="class"/>
```

```
        <xsd:attributeGroup ref="title"/>
        <xsd:attributeGroup ref="Core.extra.attrib"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
        <xsd:attributeGroup ref="Events.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:group name="map.content">
        <xsd:choice maxOccurs="unbounded">
            <xsd:group ref="Block.mix"/>
            <xsd:element ref="area" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="map.type">
        <xsd:group ref="map.content"/>
        <xsd:attributeGroup ref="area.attlist"/>
    </xsd:complexType>
    <xsd:element name="map" type="map.type"/>

        <!-- modify anchor attribute definition list -->
    <xsd:attributeGroup name="a.csim.attlist">
        <xsd:attribute name="shape" type="Text" use="default" value="rect"/>
        <xsd:attribute name="coords" type="Text"/>
    </xsd:attributeGroup>

    <xsd:redefine schemaLocation="../core/xhtml-hypertext-1.xsd">
        <xsd:attributeGroup name="a.attlist">
            <xsd:extension base="a.attlist">
                <xsd:attributeGroup ref="a.csim.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>

    <!-- modify img attribute definition list -->
    <xsd:attributeGroup name="img.csim.attlist">
        <xsd:attribute name="usemap" type="xsd:IDREF"/>
    </xsd:attributeGroup>

    <xsd:redefine schemaLocation="xhtml-image-1.xsd">
        <xsd:attributeGroup name="img.attlist">
            <xsd:extension base="img.attlist">
                <xsd:attributeGroup ref="img.csim.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>

    <!-- modify form input attribute definition list -->
    <xsd:attributeGroup name="input.csim.attlist">
        <xsd:attribute name="usemap" type="xsd:IDREF"/>
    </xsd:attributeGroup>


    <xsd:redefine schemaLocation="xhtml-form-1.xsd">
        <xsd:attributeGroup name="input.attlist">
            <xsd:extension base="input.attlist">
                <xsd:attributeGroup ref="input.csim.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>

    <!-- modify object attribute definition list -->
    <xsd:attributeGroup name="object.csim.attlist">
        <xsd:attribute name="usemap" type="xsd:IDREF"/>
    </xsd:attributeGroup>


    <xsd:redefine schemaLocation="xhtml-object-1.xsd">
        <xsd:attributeGroup name="object.attlist">
            <xsd:extension base="object.attlist">
                <xsd:attributeGroup ref="object.csim.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>
</xsd:schema>
```

D.4.7. Server-side Image Map

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Server-side Image Maps module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Server-side Image Maps 1.0//EN"
*
*/
        </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>
        <xsd:documentation>
/**
* Server-side Image Maps
*
*     This adds the 'ismap' attribute to the img element to
*     support server-side processing of a user selection.
*
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:attributeGroup  name="img.ssimap.attlist" >
        <xsd:attribute name="ismap"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="xhtml-image-1.xsd">

        <xsd:attributeGroup name="img.attlist">

            <xsd:extension base="img.attlist">

                <xsd:attributeGroup ref="img.ssimap.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>
</xsd:schema>
```

D.4.8. Applet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
```

```
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema module for Java Applets in XHTML
*  Please use this formal public identifier to identify it:
*            "-//W3C//ELEMENTS XHTML Java Applets 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Java Applets
*
*        applet
*
*     This module declares the applet element type and its attributes,
*     used to provide support for Java applets. The 'alt' attribute
*     is now required (as it is on images). One of either code or
*     object attributes must be present. In the document, place param
*     elements before the object elements that require their content.
*
*     Note that use of this module requires instantiation of the
*     Param Element Module.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

    </xsd:annotation>

    <xsd:attributeGroup name="applet.attlist">
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:attribute name="alt" type="Text"/>
        <xsd:attribute name="archive" type="URI"/>
        <xsd:attribute name="code" type="URI"/>
        <xsd:attribute name="codebase" type="URI"/>
        <xsd:attribute name="object" type="URI"/>
        <xsd:attribute name="width" type="Length"/>
        <xsd:attribute name="height" type="Length"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>

    <xsd:group name="applet.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="param"/>
            <xsd:group ref="Flow.mix"/>
        </xsd:choice>
    </xsd:group>

    <xsd:complexType name="applet.type">
        <xsd:group ref="applet.content"/>
        <xsd:attributeGroup ref="applet.attlist"/>
```

```
    </xsd:complexType>

    <xsd:element name="applet" type="applet.type"/>
</xsd:schema>
```

D.4.9. Object

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Embedded Object module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Embedded Object 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Embedded Objects
*
*        object
*
*     This module declares the object element type and its attributes,
*     used to embed external objects as part of XHTML pages. In the
*     document, place param elements prior to the object elements
*     that require their content.
*
*     Note that use of this module requires instantiation of the
*     Param Element Module prior to this module.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <xsd:attributeGroup name="object.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="declare" type="Text"/>
        <xsd:attribute name="classid" type="URI"/>
        <xsd:attribute name="codebase" type="URI"/>
        <xsd:attribute name="data" type="URI"/>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:attribute name="codetype" type="ContentType"/>
        <xsd:attribute name="archive" type="URIs"/>
        <xsd:attribute name="standby" type="Text"/>
```

```
        <xsd:attribute name="height" type="Length"/>
        <xsd:attribute name="width" type="Length"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="tabindex" type="Number"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>


    <xsd:group name="object.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="Flow.mix"/>
            <xsd:element ref="param"/>
        </xsd:choice>
    </xsd:group>

    <xsd:complexType name="object.type" mixed="true">
        <xsd:group ref="object.content" />
        <xsd:attributeGroup ref="object.attlist"/>
    </xsd:complexType>

    <xsd:element name="object" type="object.type"/>
</xsd:schema>
```

D.4.10. Frames

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
targetNamespace="http://www.w3.org/1999/xhtml"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
elementFormDefault="unqualified" version="1.1">
     <xsd:annotation>
         <xsd:documentation>
/**
* This is the XML Schema Frames module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Frames 1.0//EN"
*
*/
  </xsd:documentation>
         <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
         <xsd:documentation>
/**
* Frames
*
*        frameset, frame, noframes
*
*     This module declares frame-related element types and attributes.
*
*/
         </xsd:documentation>
         <xsd:documentation source="xhtml-copyright-1.txt"/>
     </xsd:annotation>
```

```
<!-- comma-separated list of MultiLength -->
<xsd:simpleType name="MultiLengths">
    <xsd:restriction base="xsd:CDATA"/>
</xsd:simpleType>
<!--
    The content model for XHTML documents depends on whether
    the <head> is followed by a <frameset> or <body> element.
-->
<!-- frameset -->
<xsd:attributeGroup name="frameset.attlist">
    <xsd:attributeGroup ref="Core.attrib"/>
    <xsd:attribute name="rows" type="MultiLengths"/>
    <xsd:attribute name="cols" type="MultiLengths"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:group name="frameset.content">
    <xsd:sequence>
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="frameset"/>
            <xsd:element ref="frame"/>
        </xsd:choice>
        <xsd:element ref="noframes" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="frameset.type" mixed="true">
    <xsd:group ref="frameset.content"/>
    <xsd:attributeGroup ref="frameset.attlist"/>
</xsd:complexType>
<xsd:element name="frameset" type="frameset.type"/>
<!-- frame -->
<!--
    reserved frame names start with "_"
    otherwise starts with letter
-->
<xsd:attributeGroup name="frame.attlist">
    <xsd:attributeGroup ref="Core.attrib"/>
    <xsd:attribute name="longdesc" type="URI"/>
    <xsd:attribute name="src" type="URI"/>
    <xsd:attribute name="frameborder" use="default" value="1">
        <xsd:simpleType>
            <xsd:restriction base="xsd:nonNegativeInteger">
                <xsd:enumeration value="1"/>
                <xsd:enumeration value="0"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="marginwidth" type="Pixels"/>
    <xsd:attribute name="marginheight" type="Pixels"/>
    <xsd:attribute name="noresize" type="Text"/>
    <xsd:attribute name="scrolling" use="default" value="auto">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="yes"/>
                <xsd:enumeration value="no"/>
                <xsd:enumeration value="auto"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- frame is EMPTY -->
<xsd:complexType name="frame.type">
    <xsd:attributeGroup ref="frame.attlist"/>
</xsd:complexType>
<xsd:element name="frame" type="frame.type"/>
```

```
    <!-- changes to other declarations  -->
    <!-- redefine content model for html element,
     substituting frameset for body  -->
    <xsd:redefine schemaLocation="../../../core/xhtml-struct-1.xsd">
        <xsd:group name="html.content">
            <xsd:choice minOccurs="1" maxOccurs="unbounded">
                <xsd:element ref="head" minOccurs="0" maxOccurs="1"/>
                <xsd:element ref="frameset"/>
            </xsd:choice>
        </xsd:group>
    </xsd:redefine>
    <!-- noframes -->
    <!-- alternate content container for non frame-based rendering -->
    <xsd:group name="noframes.content">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element ref="body"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="noframes.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="noframes.type" mixed="true">
        <xsd:group ref="noframes.content"/>
        <xsd:attributeGroup ref="noframes.attlist"/>
    </xsd:complexType>
    <xsd:element name="noframes" type="noframes.type"/>
</xsd:schema>
```

D.4.11. Target

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Target module for XHTML
*  Please use this formal public identifier to identify it:
*            "-//W3C//ELEMENTS XHTML Target 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Target
*
*   target
*
```

```
*
*     This module declares the 'target' attribute used for opening windows
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>

    <!-- render in this frame -->
    <xsd:simpleType name="FrameTarget">
        <xsd:restriction base="xsd:CDATA"/>
    </xsd:simpleType>

    <!-- add 'target' attribute to 'a' element -->
    <xsd:attributeGroup name="a.target.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="../../req/core/xhtml-hypertext-1.xsd">

        <xsd:attributeGroup name="a.attlist">

            <xsd:extension base="a.attlist">

                <xsd:attributeGroup ref="a.target.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>
    <!-- add 'target' attribute to 'area' element -->
    <xsd:attributeGroup name="area.target.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="xhtml-csismap-1.xsd">

        <xsd:attributeGroup name="area.attlist">

            <xsd:extension base="area.attlist">

                <xsd:attributeGroup ref="area.target.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>

    <!-- add 'target' attribute to 'link' element -->
    <xsd:attributeGroup name="link.target.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="xhtml-link-1.xsd">

        <xsd:attributeGroup name="link.attlist">

            <xsd:extension base="link.attlist">

                <xsd:attributeGroup ref="link.target.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>

    <!-- add 'target' attribute to 'form' element -->
    <xsd:attributeGroup name="form.target.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="xhtml-form-1.xsd">

        <xsd:attributeGroup name="form.attlist">
```

```
            <xsd:extension base="form.attlist">

                <xsd:attributeGroup ref="form.target.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>
    <!-- add 'target' attribute to 'base' element -->
    <xsd:attributeGroup name="base.target.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>

    <xsd:redefine schemaLocation="xhtml-base-1.xsd">

        <xsd:attributeGroup name="base.attlist">

            <xsd:extension base="base.attlist">

                <xsd:attributeGroup ref="base.target.attlist" />

            </xsd:extension>

        </xsd:attributeGroup>

    </xsd:redefine>
</xsd:schema>
```

D.4.12. Iframe

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Inline Frame Element module for XHTML
*  Please use this formal public identifier to identify it:
*            "-//W3C//ELEMENTS XHTML Inline Frame Element 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Inline Frames
*
*        iframe
*
*     This module declares the iframe element type and its attributes,
*     used to create an inline frame within a document.
*
*/
```

```
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- Inline Frames  -->
    <xsd:attributeGroup name="iframe.attlist">
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:attribute name="longdesc" type="URI"/>
        <xsd:attribute name="src" type="URI"/>
        <xsd:attribute name="frameborder" use="default" value="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:nonNegativeInteger">
                    <xsd:enumeration value="1"/>
                    <xsd:enumeration value="0"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="marginwidth" type="Pixels"/>
        <xsd:attribute name="marginheight" type="Pixels"/>
        <xsd:attribute name="scrolling" use="default" value="auto">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="yes"/>
                    <xsd:enumeration value="no"/>
                    <xsd:enumeration value="auto"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="height" type="Length"/>
        <xsd:attribute name="width" type="Length"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="iframe.type" mixed="true">
        <xsd:group ref="Flow.mix"/>
        <xsd:attributeGroup ref="iframe.attlist"/>
    </xsd:complexType>
    <xsd:element name="iframe" type="iframe.type"/>
</xsd:schema>
```

D.4.13. Intrinsic Events

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
  <xsd:documentation>
/**
* This is the XML Schema Intrinsic Events module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ENTITIES XHTML Intrinsic Events 1.0//EN"
*
*/
  </xsd:documentation>
  <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
  <xsd:documentation>
```

```
/**
* Intrinsic Event Attributes
*
*     These are the event attributes defined in HTML 4.0,
*     Section 18.2.3 "Intrinsic Events".
*
*    "Note: Authors of HTML documents are advised that changes
*     are likely to occur in the realm of intrinsic events
*     (e.g., how scripts are bound to events). Research in
*     this realm is carried on by members of the W3C Document
*     Object Model Working Group (see the W3C Web site at
*     http://www.w3.org/ for more information)."
*
*/
  </xsd:documentation>
  <xsd:documentation source="xhtml-copyright-1.txt"/>
</xsd:annotation>
        <xsd:attributeGroup name="Events.attrib">
        <xsd:attribute name="onclick" type="Script"/>
        <xsd:attribute name="ondblclick" type="Script"/>
        <xsd:attribute name="onmousedown" type="Script"/>
        <xsd:attribute name="onmouseup" type="Script"/>
        <xsd:attribute name="onmouseover" type="Script"/>
        <xsd:attribute name="onmousemove" type="Script"/>
        <xsd:attribute name="onmouseout" type="Script"/>
        <xsd:attribute name="onkeypress" type="Script"/>
        <xsd:attribute name="onkeydown" type="Script"/>
        <xsd:attribute name="onkeyup" type="Script"/>
    </xsd:attributeGroup>
    <!--
        additional attributes on anchor element
    -->
    <xsd:attributeGroup name="a.events.attlist">
        <xsd:attribute name="onfocus" type="Script"/>
        <xsd:attribute name="onblur" type="Script"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="../core/xhtml-hypertext-1.xsd">
        <xsd:attributeGroup name="a.attlist">
            <xsd:extension base="a.attlist">
                <xsd:attributeGroup ref="a.events.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>
    <!--
        additional attributes on form element
    -->
    <xsd:attributeGroup name="form.events.attlist">
        <xsd:attribute name="onsubmit" type="Script"/>
        <xsd:attribute name="onreset" type="Script"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
        <xsd:attributeGroup name="form.attlist">
            <xsd:extension base="form.attlist">
                <xsd:attributeGroup ref="form.events.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>

    <!--
        additional attributes on label element
    -->
    <xsd:attributeGroup name="label.events.attlist">
        <xsd:attribute name="onfocus" type="Script"/>
        <xsd:attribute name="onblur" type="Script"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
        <xsd:attributeGroup name="label.attlist">
            <xsd:extension base="label.attlist">
                <xsd:attributeGroup ref="label.events.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>
    <!--
        additional attributes on input element
```

```
-->
<xsd:attributeGroup name="input.events.attlist">
    <xsd:attribute name="onfocus" type="Script"/>
    <xsd:attribute name="onblur" type="Script"/>
    <xsd:attribute name="onselect" type="Script"/>
    <xsd:attribute name="onchange" type="Script"/>
</xsd:attributeGroup>
<xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
    <xsd:attributeGroup name="input.attlist">
        <xsd:extension base="input.attlist">
            <xsd:attributeGroup ref="input.events.attlist"/>
        </xsd:extension>
    </xsd:attributeGroup>
</xsd:redefine>

<!--
    additional attributes on select element
-->
<xsd:attributeGroup name="select.events.attlist">
    <xsd:attribute name="onfocus" type="Script"/>
    <xsd:attribute name="onblur" type="Script"/>
    <xsd:attribute name="onchange" type="Script"/>
</xsd:attributeGroup>
<xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
    <xsd:attributeGroup name="select.attlist">
        <xsd:extension base="select.attlist">
            <xsd:attributeGroup ref="select.events.attlist"/>
        </xsd:extension>
    </xsd:attributeGroup>
</xsd:redefine>
<!--
    additional attributes on textarea element
-->
<xsd:attributeGroup name="textarea.events.attlist">
    <xsd:attribute name="onfocus" type="Script"/>
    <xsd:attribute name="onblur" type="Script"/>
    <xsd:attribute name="onselect" type="Script"/>
    <xsd:attribute name="onchange" type="Script"/>
</xsd:attributeGroup>
<xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
    <xsd:attributeGroup name="textarea.attlist">
        <xsd:extension base="textarea.attlist">
            <xsd:attributeGroup ref="textarea.events.attlist"/>
        </xsd:extension>
    </xsd:attributeGroup>
</xsd:redefine>
<!--
    additional attributes on button element
-->
<xsd:attributeGroup name="button.events.attlist">
    <xsd:attribute name="onfocus" type="Script"/>
    <xsd:attribute name="onblur" type="Script"/>
</xsd:attributeGroup>
<xsd:redefine schemaLocation="../../opt/xhtml-form-1.xsd">
    <xsd:attributeGroup name="button.attlist">
        <xsd:extension base="button.attlist">
            <xsd:attributeGroup ref="button.events.attlist"/>
        </xsd:extension>
    </xsd:attributeGroup>
</xsd:redefine>
<!--
    additional attributes on body element
-->
<xsd:attributeGroup name="body.events.attlist">
    <xsd:attribute name="onload" type="Script"/>
    <xsd:attribute name="onunload" type="Script"/>
</xsd:attributeGroup>
<xsd:redefine schemaLocation="../core/xhtml-struct-1.xsd">
    <xsd:attributeGroup name="body.attlist">
        <xsd:extension base="body.attlist">
            <xsd:attributeGroup ref="body.events.attlist"/>
        </xsd:extension>
    </xsd:attributeGroup>
</xsd:redefine>
```

```
    <!--
        additional attributes on area element
    -->
    <xsd:attributeGroup name="area.events.attlist">
        <xsd:attribute name="onfocus" type="Script"/>
        <xsd:attribute name="onblur" type="Script"/>
    </xsd:attributeGroup>
    <xsd:redefine schemaLocation="../../opt/xhtml-csismap-1.xsd">
        <xsd:attributeGroup name="area.attlist">
            <xsd:extension base="area.attlist">
                <xsd:attributeGroup ref="area.events.attlist"/>
            </xsd:extension>
        </xsd:attributeGroup>
    </xsd:redefine>
</xsd:schema>
```

D.4.14. Metainformation

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Metainformation module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ELEMENTS XHTML Metainformation 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Meta Information
*
*        meta
*
*     This module declares the meta element type and its attributes,
*     used to provide declarative document metainformation.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- meta: Generic Metainformation  -->
    <xsd:attributeGroup name="meta.attlist">
        <xsd:attributeGroup ref="I18n.attrib"/>
        <xsd:attribute name="http-equiv" type="xsd:NMTOKEN"/>
        <xsd:attribute name="name" type="xsd:NMTOKEN"/>
        <xsd:attribute name="content" type="Text" use="required"/>
        <xsd:attribute name="scheme" type="Text"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="meta.type">
```

```
        <xsd:attributeGroup ref="meta.attlist"/>
    </xsd:complexType>
    <xsd:element name="meta" type="meta.type"/>
</xsd:schema>
```

D.4.15. Scripting

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
        xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Scripting module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Scripting 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Scripting
*
*        script, noscript
*
*     This module declares element types and attributes used to provide
*     support for executable scripts as well as an alternate content
*     container where scripts are not supported.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- script: Scripting Statement -->
    <xsd:attributeGroup name="script.attlist">
        <xsd:attribute name="charset" type="Charset"/>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:attribute name="src" type="URI"/>
        <xsd:attribute name="defer" type="Text"/>
        <xsd:attribute name="xml:space" type="Text" use="fixed" value="preserve"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="script.type" mixed="true">
        <xsd:attributeGroup ref="script.attlist"/>
    </xsd:complexType>
    <xsd:element name="script" type="script.type"/>
    <!-- noscript: No-Script Alternate Content  -->
    <xsd:attributeGroup name="noscript.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
```

```
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="noscript.type">
        <xsd:group ref="Block.mix" minOccurs="1"/>
        <xsd:attributeGroup ref="noscript.attlist"/>
    </xsd:complexType>
    <xsd:element name="noscript" type="noscript.type"/>
</xsd:schema>
```

D.4.16. Stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Stylesheets module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//DTD XHTML Stylesheets 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Stylesheets
*
*        style
*
*     This module declares the style element type and its attributes,
*     used to embed stylesheet information in the document head element.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- style: Stylesheet Information -->
    <xsd:attributeGroup name="style.attlist">
        <xsd:attributeGroup ref="title"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:attribute name="media" type="MediaDesc"/>
        <xsd:attribute name="xml:space" type="Text" use="fixed" value="preserve"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="style.type" mixed="true">
        <xsd:attributeGroup ref="style.attlist"/>
    </xsd:complexType>
    <xsd:element name="style" type="style.type"/>
</xsd:schema>
```

D.4.17. Style Attribute

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Inline Style module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ENTITIES XHTML Inline Style 1.0//EN"
*
*/
  </xsd:documentation>
     <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
    <xsd:documentation>
/**
* Inline Style
*
*     This module declares the 'style' attribute, used to support inline
*     style markup.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>
</xsd:annotation>
    <xsd:attributeGroup name="style.inline.attlist">
        <xsd:attribute name="style" type="Text" />
    </xsd:attributeGroup>
<xsd:redefine schemaLocation="../../req/core/xhtml-attrib-1.xsd">
    <xsd:attributeGroup name="Core.extra.attrib">

        <xsd:extension base="object.attlist">

            <xsd:attributeGroup ref="style.inline.attlist" />

        </xsd:extension>

    </xsd:attributeGroup>
</xsd:redefine>
</xsd:schema>
```

D.4.18. Link

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
```

```
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Link Element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Link Element 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Link element
*
*        link
*
*     This module declares the link element type and its attributes,
*     which could (in principle) be used to define document-level links
*     to external resources such as:
*
*     a) for document specific toolbars/menus, e.g. start, contents,
*        previous, next, index, end, help
*     b) to link to a separate style sheet (rel="stylesheet")
*     c) to make a link to a script (rel="script")
*     d) by stylesheets to control how collections of html nodes are
*        rendered into printed documents
*     e) to make a link to a printable version of this document
*        e.g. a postscript or pdf version (rel="alternate" media="print")
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!-- link: Media-Independent Link -->
    <xsd:attributeGroup name="link.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="charset" type="Charset"/>
        <xsd:attribute name="href" type="URI"/>
        <xsd:attribute name="hreflang" type="LanguageCode"/>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:attribute name="rel" type="LinkTypes"/>
        <xsd:attribute name="rev" type="LinkTypes"/>
        <xsd:attribute name="media" type="MediaDesc"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <!-- link is EMPTY -->
    <xsd:complexType name="link.type">
        <xsd:attributeGroup ref="link.attlist"/>
    </xsd:complexType>
    <xsd:element name="link" type="link.type"/>
</xsd:schema>
```

D.4.19. Base

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Base Element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Base Element 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Base element
*
*        base
*
*     This module declares the base element type and its attributes,
*     used to define a base URI against which relative URIs in the
*     document will be resolved.
*
*     Note that this module also redeclares the content model for
*     the head element to include the base element.
*
*/

        </xsd:documentation>
<xsd:documentation source="xhtml-copyright-1.txt"/>
</xsd:annotation>


    <xsd:attributeGroup name="base.attlist">
        <xsd:attribute name="href" type="URI" use="required" />
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>

<!-- base is EMPTY -->
    <xsd:complexType name="base.type">
        <xsd:attributeGroup ref="base.attlist"/>
    </xsd:complexType>

    <xsd:element name="base" type="base.type"/>
</xsd:schema>
```

D.4.20. Name Identification

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Name Identifier module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Name Identifier 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Name Identifier
*
*       'name' attribute on form, img, a, map, applet, frame, iframe
*
*     This module declares the 'name' attribute on element types when
*     it is used as a node identifier for legacy linking and scripting
*     support. This does not include those instances when 'name' is used
*     as a container for form control, property or metainformation names.
*
*     This module should be instantiated following all modules it modifies.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <xsd:attributeGroup name="form.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

    <xsd:attributeGroup name="img.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

    <xsd:attributeGroup name="a.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

    <xsd:attributeGroup name="map.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>
```

```
    <xsd:attributeGroup name="applet.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

    <xsd:attributeGroup name="frame.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

    <xsd:attributeGroup name="iframe.name.attlist">
        <xsd:attribute name="name" type="Text"/>
    </xsd:attributeGroup>

</xsd:schema>
```

D.4.21. Legacy

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Legacy Markup module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Legacy Markup 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* HTML Legacy Markup
*
* this module includes modules for both frames and
* additional deprecated elements and attributes.
*
*/

</xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <xsd:annotation>
        <xsd:documentation>
/**
* Framedefs module
* Elements defined here:
*      frameset, frame, noframes, att:target, iframe
*
*/
</xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="xhtml-framedefs-1.xsd"/>
    <xsd:annotation>
```

```
        <xsd:documentation>
/**
* Miscellaneous module
* Attributes defined here:
*    font, basefont, center, s, strike, u, dir, menu, isindex
*    (plus additional datatypes and attributes)
*/
</xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="misc/xhtml-misc-1.xsd"/>
</xsd:schema>
```

D.4.22. Ruby

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Ruby module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Ruby 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

        <xsd:documentation>
/**
* Ruby Elements
*
*        ruby, rbc, rtc, rb, rt, rp
*
*     This module declares the elements and their attributes used to
*     support ruby annotation markup.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

    </xsd:annotation>

    <!-- Fragments for the content model of the ruby element -->

    <xsd:group name="Ruby.content.simple.fallback" >
        <xsd:sequence>
            <xsd:element ref="rb"  />
            <xsd:element ref="rp" minOccurs="0" maxOccurs="1" />
            <xsd:element ref="rt" />
            <xsd:element ref="rp" minOccurs="0" maxOccurs="1" />
```

```
        </xsd:sequence>
    </xsd:group>

    <xsd:group name="Ruby.content.simple" >
        <xsd:sequence>
            <xsd:element ref="rb"  />
            <xsd:element ref="rt" />
        </xsd:sequence>
    </xsd:group>

    <xsd:group name="Ruby.content.group" >
        <xsd:sequence>
            <xsd:element ref="rbc"  />
            <xsd:element ref="rtc" />
            <xsd:element ref="rtc" minOccurs="0" maxOccurs="1"  />
        </xsd:sequence>
    </xsd:group>

    <!--
        Content models of the rb and the rt elements are intended to
         allow other inline-level elements of its parent markup language,
         but it should not include ruby descendent elements. The following
         parameter entity Noruby.content can be used to redefine
         those content models with minimum effort.  It's defined as
         '( #PCDATA )' by default.
    -->

<!-- not used, mixed types instead -->
<xsd:complexType name="Noruby.content" mixed="true">
</xsd:complexType>

    <!-- ruby element  -->
    <!--
        Ruby Common Attributes

         The following optional ATTLIST declarations provide an easy way
         to define common attributes for ruby elements.  These declarations
         are ignored by default.

         Ruby elements are intended to have common attributes of its
         parent markup language.
    -->

    <xsd:attributeGroup name="Ruby.common.attrib">
    </xsd:attributeGroup>

    <!-- common attributes for ruby  -->

    <xsd:attributeGroup name="Ruby.common.attlist">
        <xsd:attributeGroup ref="Ruby.common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>

    <xsd:group name="Ruby.content">
        <xsd:sequence>
            <xsd:group ref="Ruby.content.simple"  />
            <xsd:group ref="Ruby.content.group"  />
        </xsd:sequence>
    </xsd:group>

    <xsd:complexType name="Ruby.type" >
        <xsd:group ref="Ruby.content"  />
        <xsd:attributeGroup ref="Ruby.common.attlist"/>
    </xsd:complexType>

    <xsd:element name="ruby" type="Ruby.type" />

    <!-- rbc (ruby base component) element -->

    <!-- common attributes for rbc  -->

    <xsd:attributeGroup name="Rbc.common.attlist">
        <xsd:attributeGroup ref="Ruby.common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
```

```
</xsd:attributeGroup>

<xsd:complexType name="Rbc.type" >
    <xsd:sequence minOccurs="1" maxOccurs="unbounded" >
        <xsd:element ref="rb" />
    </xsd:sequence>
    <xsd:attributeGroup ref="Rbc.common.attlist"/>
</xsd:complexType>

<xsd:element name="rbc" type="Rbc.type" />

<!-- rtc (ruby text component) element  -->

<!-- common attributes for rtc  -->

<xsd:attributeGroup name="Rtc.common.attlist">
    <xsd:attributeGroup ref="Ruby.common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="Rtc.type" >
    <xsd:sequence minOccurs="1" maxOccurs="unbounded" >
        <xsd:element ref="rt" />
    </xsd:sequence>
    <xsd:attributeGroup ref="Rtc.common.attlist"/>
</xsd:complexType>

<xsd:element name="rtc" type="Rtc.type" />

<!-- rb (ruby base) element -->

<!-- common attributes for rb -->
<xsd:attributeGroup name="Rb.common.attlist">
    <xsd:attributeGroup ref="Ruby.common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="Rb.type" mixed="true">
    <xsd:attributeGroup ref="Rb.common.attlist"/>
</xsd:complexType>

<xsd:element name="rb" type="Rb.type" />

<!-- rt (ruby text) element  -->

<!--
     Rt.content; uses Noruby.content; as its content model,
     which is '( #PCDATA )' by default. It may be overridden
     by other modules to allow other inline-level elements
     of its parent markup language, but it should not include
     ruby descendent elements.
-->

<!-- common attributes for rt  -->

<!-- rbspan attribute is used for group ruby only .. -->
<xsd:attributeGroup name="Rt.common.attlist">
    <xsd:attribute name="rbspan" use="fixed" value="1" />
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>

<xsd:complexType name="Rt.type" mixed="true">
    <xsd:attributeGroup ref="Rt.common.attlist"/>
</xsd:complexType>

<xsd:element name="rt" type="Rt.type" />

<!-- rp (ruby parenthesis) element  -->

<!-- common attributes for rp  -->
<xsd:attributeGroup name="Rp.common.attlist">
    <xsd:attributeGroup ref="Ruby.common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
```

```
    <xsd:complexType name="Rp.type" mixed="true" >
        <xsd:attributeGroup ref="Rp.common.attlist"/>
    </xsd:complexType>

    <xsd:element name="rp" type="Rp.type" />

</xsd:schema>
```

D.5. XHTML Schema Support Modules

The modules in this section are elements of the XHTML DTD implementation that, while hidden from casual users, are important to understand when creating derivative markup languages using the Modularization architecture.

D.5.1. Block Phrasal

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema block phrasal element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Block Phrasal 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*Block Phrasal
*
*        address, blockquote, pre, h1, h2, h3, h4, h5, h6
*
*     This module declares the elements and their attributes used to
*     support block-level phrasal markup.
*
*/
  </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- address -->
    <xsd:attributeGroup name="address.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="address.type" mixed="true">
```

```
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="address.attlist"/>
    </xsd:complexType>
    <xsd:element name="address" type="address.type"/>
    <!-- blockquote -->
    <xsd:attributeGroup name="blockquote.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="cite" type="URI"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="blockquote.type">
        <xsd:group ref="Block.mix"/>
        <xsd:attributeGroup ref="blockquote.attlist"/>
    </xsd:complexType>
    <xsd:element name="blockquote" type="blockquote.type"/>
    <!-- pre -->
    <xsd:group name="pre.content">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="InlStruct.class"/>
            <xsd:group ref="InlPhras.class"/>
            <xsd:element ref="tt"/>
            <xsd:element ref="i"/>
            <xsd:element ref="b"/>
            <xsd:group ref="I18n.class"/>
            <xsd:group ref="Anchor.class"/>
            <xsd:element ref="script"/>
            <xsd:element ref="map"/>
            <xsd:group ref="Inline.extra"/>
        </xsd:choice>
    </xsd:group>
    <xsd:attributeGroup name="pre.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:attribute name="xml:space" type="Text" use="fixed" value="preserve"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="pre.type" mixed="true">
        <xsd:group ref="pre.content"/>
        <xsd:attributeGroup ref="pre.attlist"/>
    </xsd:complexType>
    <xsd:element name="pre" type="pre.type"/>
    <!-- Heading Elements  -->
    <xsd:attributeGroup name="heading.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="heading.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="heading.attlist"/>
    </xsd:complexType>
    <xsd:element name="h1" type="heading.type"/>
    <xsd:element name="h2" type="heading.type"/>
    <xsd:element name="h3" type="heading.type"/>
    <xsd:element name="h4" type="heading.type"/>
    <xsd:element name="h5" type="heading.type"/>
    <xsd:element name="h6" type="heading.type"/>
</xsd:schema>
```

D.5.2. Block Presentational

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML SchemaBlock presentation element module for XHTML
```

```
*  Please use this formal public identifier to identify it:
*            "-//W3C//ELEMENTS XHTML Block Presentation 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Block Presentational Elements
*
*       hr
*
*     This module declares the elements and their attributes used to
*     support block-level presentational markup.
*
*/
  </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

    <!-- hr is EMPTY -->
    <xsd:attributeGroup name="hr.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="hr.type">
        <xsd:attributeGroup ref="hr.attlist"/>
    </xsd:complexType>
    <xsd:element name="hr" type="hr.type"/>
</xsd:schema>
```

D.5.3. Block Structural

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Block Structural module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Block Structural 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
```

```
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Block Structural
*
*        div, p
*
*     This module declares the elements and their attributes used to
*     support block-level structural markup.
*
*/
  </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- div -->
    <xsd:attributeGroup name="div.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="div.type" mixed="true">
        <xsd:group ref="Flow.mix"/>
        <xsd:attributeGroup ref="div.attlist"/>
    </xsd:complexType>
    <xsd:element name="div" type="div.type"/>
    <!-- p -->
    <xsd:attributeGroup name="p.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="p.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="p.attlist"/>
    </xsd:complexType>
    <xsd:element name="p" type="p.type"/>
</xsd:schema>
```

D.5.4. Inline Phrasal

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Inline Phrasal module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Inline Phrasal 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
```

```
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Inline Phrasal
*
*        abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
*
*     This module declares the elements and their attributes used to
*     support inline-level phrasal markup.
*
*/
  </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

<xsd:annotation>
    <xsd:documentation>
/**
* With the exception of the q element, all of these elements have exactly the same content
* models and attribute lists.
*
*/
  </xsd:documentation>
</xsd:annotation>

    <xsd:attributeGroup name="InlPhras.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="InlPhras.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="InlPhras.attlist"/>
    </xsd:complexType>
    <xsd:element name="abbr" type="InlPhras.type"/>
    <xsd:element name="acronym" type="InlPhras.type"/>
    <xsd:element name="cite" type="InlPhras.type"/>
    <xsd:element name="code" type="InlPhras.type"/>
    <xsd:element name="dfn" type="InlPhras.type"/>
    <xsd:element name="em" type="InlPhras.type"/>
    <xsd:element name="kbd" type="InlPhras.type"/>
    <xsd:element name="samp" type="InlPhras.type"/>
    <xsd:element name="strong" type="InlPhras.type"/>
    <xsd:element name="var" type="InlPhras.type"/>
    <xsd:attributeGroup name="q.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <!-- note the additional attribute here -->
        <xsd:attribute name="cite" type="URI"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="q.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="InlPhras.attlist"/>
    </xsd:complexType>
    <xsd:element name="q" type="q.type"/>
</xsd:schema>
```

D.5.5. Inline Presentational

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Inline Presentation element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Inline Presentation 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
* Inline Presentational Elements
*
*        b, big, i, small, sub, sup, tt
*
*     This module declares the elements and their attributes used to
*     support inline-level presentational markup.
*
*/
        </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>

</xsd:annotation>

<xsd:annotation>
    <xsd:documentation>
/**
* All of these elements have exactly the same content
* models and attribute lists.
*
*/
  </xsd:documentation>
</xsd:annotation>

    <xsd:attributeGroup name="InlPres.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="InlPres.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="InlPres.attlist"/>
    </xsd:complexType>
    <xsd:element name="b" type="InlPres.type"/>
    <xsd:element name="big" type="InlPres.type"/>
```

```
    <xsd:element name="i" type="InlPres.type"/>
    <xsd:element name="small" type="InlPres.type"/>
    <xsd:element name="sub" type="InlPres.type"/>
    <xsd:element name="sup" type="InlPres.type"/>
    <xsd:element name="tt" type="InlPres.type"/>
</xsd:schema>
```

D.5.6. Inline Structural

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Inline Structural element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Inline Structural 1.0//EN"
*
*/
  </xsd:documentation>
</xsd:annotation>

<xsd:annotation>
     <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
</xsd:annotation>
<xsd:annotation>
    <xsd:documentation>
/**
* Inline Structural
*
*        br, span
*
*     This module declares the elements and their attributes
*     used to support inline-level structural markup.
*
*/
        </xsd:documentation>
<xsd:documentation source="xhtml-copyright-1.txt"/>
</xsd:annotation>

    <!-- br is EMPTY -->
    <xsd:attributeGroup name="br.attlist">
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="br.type">
        <xsd:attributeGroup ref="br.attlist"/>
    </xsd:complexType>
    <xsd:element name="br" type="br.type"/>
    <!-- span -->
    <xsd:attributeGroup name="span.attlist">
        <xsd:attributeGroup ref="Common.attrib"/>
```

```
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
    <xsd:complexType name="span.type" mixed="true">
        <xsd:group ref="Inline.mix"/>
        <xsd:attributeGroup ref="span.attlist"/>
    </xsd:complexType>
    <xsd:element name="span" type="span.type"/>
</xsd:schema>
```

D.5.7. Param

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Param Element module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Param Element 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* Parameters for Java Applets and Embedded Objects
*
*        param
*
*     This module provides declarations for the param element,
*     used to provide named property values for the applet
*     and object elements.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <xsd:attributeGroup name="param.attlist">
        <xsd:attributeGroup ref="id"/>
        <xsd:attribute name="name" type="Text"/>
        <xsd:attribute name="value" type="Text"/>
        <xsd:attribute name="valuetype" use="default" value="data">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="data"/>
                    <xsd:enumeration value="ref"/>
                    <xsd:enumeration value="object"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="type" type="ContentType"/>
        <xsd:anyAttribute namespace="##other"/>
    </xsd:attributeGroup>
```

```
    <!-- param is EMPTY -->
    <xsd:complexType name="param.type" mixed="true">
        <xsd:attributeGroup ref="param.attlist"/>
    </xsd:complexType>
    <xsd:element name="param" type="param.type"/>
</xsd:schema>
```

D.5.8. Miscellaneous Legacy

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema Miscellaneous Legacy Markup module for
* XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Miscellaneous Legacy Markup 1.0//EN"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
* HTML Miscellaneous Legacy Markup
*
*        font, basefont, center, s, strike, u, dir, menu, isindex
*
*          (plus additional datatypes and attributes)
*
*     This optional module declares additional markup for simple
*     presentation-related markup based on features found in the
*     HTML 4.0 Transitional and Frameset DTDs. This relies on
*     inclusion of the Legacy Redeclarations module. This module
*     also declares the frames, inline frames and object modules.
*
*     This is to allow XHTML documents to be transformed for
*     display on HTML browsers where CSS support is inconsistent
*     or unavailable.
*
*/
        </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
    </xsd:annotation>
    <!-- Additional Element Types  -->
    <!-- font -->
    <xsd:attributeGroup name="font.attlist">
        <xsd:attributeGroup ref="Core.attrib"/>
        <xsd:attributeGroup ref="I18n.attrib"/>
        <xsd:attribute name="size" type="Text"/>
        <xsd:attribute name="color" type="Color"/>
        <xsd:attribute name="face" type="Text"/>
        <xsd:anyAttribute namespace="##other"/>
```

```
</xsd:attributeGroup>
<xsd:group name="font.content">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="Inline.mix"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="font.type" mixed="true">
    <xsd:group ref="font.content"/>
    <xsd:attributeGroup ref="font.attlist"/>
</xsd:complexType>
<xsd:element name="font" type="font.type"/>
<!-- basefont -->
<xsd:attributeGroup name="basefont.attlist">
    <xsd:attributeGroup ref="id"/>
    <xsd:attribute name="size" type="Text"/>
    <xsd:attribute name="color" type="Color"/>
    <xsd:attribute name="face" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- basefont is EMPTY -->
<xsd:complexType name="basefont.type">
    <xsd:attributeGroup ref="basefont.attlist"/>
</xsd:complexType>
<xsd:element name="basefont" type="basefont.type"/>
<!-- center -->
<xsd:attributeGroup name="center.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="center.type" mixed="true">
    <xsd:group ref="Flow.mix"/>
    <xsd:attributeGroup ref="center.attlist"/>
</xsd:complexType>
<xsd:element name="center" type="center.type"/>
<!-- s -->
<xsd:attributeGroup name="s.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="s.type" mixed="true">
    <xsd:group ref="Inline.mix"/>
    <xsd:attributeGroup ref="s.attlist"/>
</xsd:complexType>
<xsd:element name="s" type="s.type"/>
<!-- strike -->
<xsd:attributeGroup name="strike.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="strike.type" mixed="true">
    <xsd:group ref="Inline.mix"/>
    <xsd:attributeGroup ref="strike.attlist"/>
</xsd:complexType>
<xsd:element name="strike" type="strike.type"/>
<!-- u -->
<xsd:attributeGroup name="u.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="u.type" mixed="true">
    <xsd:group ref="Inline.mix"/>
    <xsd:attributeGroup ref="u.attlist"/>
</xsd:complexType>
<xsd:element name="u" type="u.type"/>
<!-- dir -->
<!--
    NOTE: the content model for <dir> in HTML 4 excluded %Block.mix;
-->
<xsd:attributeGroup name="dir.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="compact" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="dir.type" mixed="true">
```

```
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="li"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="dir.attlist"/>
</xsd:complexType>
<xsd:element name="dir" type="dir.type"/>
<!-- menu -->
<!--
    NOTE: the content model for <menu> in HTML 4 excluded %Block.mix;
-->
<xsd:attributeGroup name="menu.attlist">
    <xsd:attributeGroup ref="Common.attrib"/>
    <xsd:attribute name="compact" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<xsd:complexType name="menu.type" mixed="true">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="li"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="menu.attlist"/>
</xsd:complexType>
<xsd:element name="menu" type="menu.type"/>
<!-- isindex -->
<xsd:attributeGroup name="isindex.attlist">
    <xsd:attributeGroup ref="Core.attrib"/>
    <xsd:attributeGroup ref="I18n.attrib"/>
    <xsd:attribute name="prompt" type="Text"/>
    <xsd:anyAttribute namespace="##other"/>
</xsd:attributeGroup>
<!-- isindex is EMPTY -->
<xsd:complexType name="isindex.type">
    <xsd:attributeGroup ref="isindex.attlist"/>
</xsd:complexType>
<xsd:element name="isindex" type="isindex.type"/>


<!-- Additional Attributes -->
<!-- !!! these should be redefines to add the attributes -->

<!--
    Alignment attribute for Transitional use in HTML browsers
    (this functionality is generally well-supported in CSS,
    except within some contexts)
-->


    <xsd:attributeGroup name="align.legacy.attlist">
        <xsd:attribute name="align">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="left"/>
                    <xsd:enumeration value="center"/>
                    <xsd:enumeration value="right"/>
                    <xsd:enumeration value="justify"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:attributeGroup>


    <!-- add 'target' attribute to 'a' element -->
    <xsd:attributeGroup name="a.legacy.attlist">
        <xsd:attribute name="target" type="FrameTarget"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="applet.legacy.attlist">
        <xsd:attribute name="align">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="top"/>
                    <xsd:enumeration value="middle"/>
                    <xsd:enumeration value="bottom"/>
                    <xsd:enumeration value="left"/>
```

```
                    <xsd:enumeration value="right"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="hspace" type="Pixels"/>
        <xsd:attribute name="vspace" type="Pixels"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="body.legacy.attlist">
        <xsd:attribute name="background" type="URI"/>
        <xsd:attribute name="bgcolor" type="Color"/>
        <xsd:attribute name="text" type="Color"/>
        <xsd:attribute name="link" type="Color"/>
        <xsd:attribute name="vlinke" type="Color"/>
        <xsd:attribute name="alink" type="Color"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="br.legacy.attlist">
        <xsd:attribute name="clear" use="default" value="none">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="left"/>
                    <xsd:enumeration value="all"/>
                    <xsd:enumeration value="right"/>
                    <xsd:enumeration value="none"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:attributeGroup>

<!-- its nice to be able to reference abstract modules rather than each invidual one. -->

    <xsd:attributeGroup name="caption.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="h1.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="h3.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="h4.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="h5.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="h6.legacy.attlist">
        <xsd:attributeGroup ref="align.legacy.attlist"/>
    </xsd:attributeGroup>


    <xsd:attributeGroup name="hr.legacy.attlist">
        <xsd:attribute name="align">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="left"/>
                    <xsd:enumeration value="center"/>
                    <xsd:enumeration value="right"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
```

```
    <xsd:attribute name="noshade" use="default" value="none"/>
    <xsd:attribute name="size" type="Pixels"/>
    <xsd:attribute name="width" type="Length"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="img.legacy.attlist">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="top"/>
                <xsd:enumeration value="middle"/>
                <xsd:enumeration value="bottom"/>
                <xsd:enumeration value="left"/>
                <xsd:enumeration value="right"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="border" type="Pixels"/>
    <xsd:attribute name="hspace" type="Length"/>
    <xsd:attribute name="vspace" type="Length"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="input.legacy.attlist">
    <xsd:attributeGroup ref="align.legacy.attlist"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="legend.legacy.attlist">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="top"/>
                <xsd:enumeration value="bottom"/>
                <xsd:enumeration value="left"/>
                <xsd:enumeration value="right"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:attributeGroup>

<xsd:attributeGroup name="li.legacy.attlist">
    <xsd:attribute name="type" type="Text"/>
    <xsd:attribute name="value" type="Text"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="object.legacy.attlist">
    <xsd:attribute name="align">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="top"/>
                <xsd:enumeration value="middle"/>
                <xsd:enumeration value="bottom"/>
                <xsd:enumeration value="left"/>
                <xsd:enumeration value="right"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="border" type="Pixels"/>
    <xsd:attribute name="hspace" type="Length"/>
    <xsd:attribute name="vspace" type="Length"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="ol.legacy.attlist">
    <xsd:attribute name="type" type="Text"/>
    <xsd:attribute name="compact" use="default" value="none"/>
    <xsd:attribute name="start" type="Text"/>
</xsd:attributeGroup>


<xsd:attributeGroup name="p.legacy.attlist">
```

```
            <xsd:attributeGroup ref="align.legacy.attlist"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="pre.legacy.attlist">
            <xsd:attribute name="width" type="Length"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="script.legacy.attlist">
            <xsd:attribute name="language" type="ContentType"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="table.legacy.attlist">
            <xsd:attributeGroup ref="align.legacy.attlist"/>
            <xsd:attribute name="bgcolor" type="Color"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="tr.legacy.attlist">
            <xsd:attribute name="bgcolor" type="Color"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="th.legacy.attlist">
            <xsd:attribute name="nowrap" use="default" value="none"/>
            <xsd:attribute name="bgcolor" type="Color"/>
            <xsd:attribute name="width" type="Length"/>
            <xsd:attribute name="height" type="Length"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="td.legacy.attlist">
            <xsd:attribute name="nowrap" use="default" value="none"/>
            <xsd:attribute name="bgcolor" type="Color"/>
            <xsd:attribute name="width" type="Length"/>
            <xsd:attribute name="height" type="Length"/>
        </xsd:attributeGroup>


        <xsd:attributeGroup name="ul.legacy.attlist">
            <xsd:attribute name="type" type="Text"/>
            <xsd:attribute name="compact" use="default" value="none"/>
        </xsd:attributeGroup>

</xsd:schema>
```

D.5.9. Legacy Frames

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
     <xsd:documentation>
/**
* This is the XML Schema Frames Definition module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ELEMENTS XHTML Frames Definitions 1.0//EN"
*
*/
  </xsd:documentation>

     <xsd:documentation>
/**
*
```

```
* Versioning block
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>

    <xsd:documentation>
/**
* Frames Definitions
*
*      frameset, frame, noframes, att:target, iframe
*   This module includes 3 additional modules, for frames,
*   target attribute, and iframe.
*
*/
    </xsd:documentation>

<xsd:documentation source="xhtml-copyright-1.txt"/>
</xsd:annotation>

    <xsd:annotation>
        <xsd:documentation>
/**
* Frames module
* Elements defined here:
*      frameset, frame, noframes
*
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="frames/xhtml-frames-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Target module
* Attributes defined here:
*        target
*/
</xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="frames/xhtml-target-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Iframe module
* Elements defined here:
*        iframe
*/
</xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="frames/xhtml-iframe-1.xsd"/>

</xsd:schema>
```

D.5.10. Optional Module Hub

```
<?xml version="1.0"?>
<schema targetNamespace="http://www.w3.org/1999/xhtml"
         xmlns="http://www.w3.org/2000/10/XMLSchema"
         elementFormDefault="unqualified"
         version="1.1"
>
     <annotation>
```

```
        <documentation>
/**
* This is the XML Schema Optional Elementsr module for XHTML
*  Please use this formal public identifier to identify it:
*           "-//W3C//ELEMENTS XHTML Optional  1.0//EN"
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
            <include schemaLocation="xhtml-copyright-1.xsd"/>
        </documentation>
    </annotation>

    <annotation>
        <documentation>
./**
* Optional Element modules
*
* Comment out those you don't need.
* Note that you will also need (probably)
* to modify the content models file.
*
*        +  Edit
*        +  Bdo
*        +  Presentational
*        +  Link
*        +  Meta
*        +  Base
*        +  Scripting
*        +  Style
*        +  Image
*        +  Client side image maps
*        +  Server side image maps
*        +  Param
*        +  Applet
*        +  Object
*        +  Tables
*        +  Forms
```

```
*         +  Nameident
*         +  Legacy
*         +  Basic forms
*         +  Basic tables
*
*/
        </documentation>
    </annotation>
    <annotation>
        <documentation>
/**
* Edit module
*.
*Elements defined here:
* ins,del
*
*/
    </documentation>
    </annotation>
    <include schemaLocation="opt/xhtml-edit-1.xsd"/>

    <annotation>
        <documentation>
/**
* Bidirectional element module
*.
*Elements defined here:
* bdo
*
*/
    </documentation>
    </annotation>
    <include schemaLocation="opt/xhtml-bdo-1.xsd"/>
    <annotation>
        <documentation>
/**
* Presentational module
*.
* Elements defined here:
* hr
* b, big, i, small, sub, sup, tt
*
*/
    </documentation>
    </annotation>
    <include schemaLocation="opt/xhtml-pres-1.xsd"/>
<annotation>
<documentation>
/**
* Link module
* Elements defined here:
*         link
*
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-link-1.xsd"/>
```

```
<annotation>
<documentation>
/**
* Meta module
* Elements defined here:
*        meta
*
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-meta-1.xsd"/>
<annotation>
<documentation>
/**
* Base module
* Elements defined here:
*        base
*
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-base-1.xsd"/>
<annotation>
<documentation>
/**
* Scripting module
* Elements defined here:
*        script, noscript
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-script-1.xsd"/>
<annotation>
<documentation>
/**
* Style module
* Elements defined here:
*        style
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-style-1.xsd"/>
<annotation>
<documentation>
/**
* Image module
* Elements defined here:
*        img
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-image-1.xsd"/>
<annotation>
<documentation>
/**
* Client-side mage maps module
* Elements defined here:
```

```
*        area, map
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-csismap-1.xsd"/>
<annotation>
<documentation>
/**
* Server-side image maps module
* Attributes defined here:
*        ismap->img
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-ssismap-1.xsd"/>
<annotation>
<documentation>
/**
* Param module
* Elements defined here:
*        param
*
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-param-1.xsd"/>
    <annotation>
        <documentation>
/**
* Applet module
* Elements defined here:
*        applet
*/
    </documentation>
    </annotation>
    <include schemaLocation="opt/xhtml-applet-1.xsd"/>
<annotation>
<documentation>
/**
* Object module
* Elements defined here:
*        object
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-object-1.xsd"/>
<annotation>
<documentation>
/**
* Tables module
* Elements defined here:
*        table, caption, thead, tfoot, tbody, colgroup, col, tr, th, td
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-table-1.xsd"/>
<annotation>
```

```
<documentation>
/**
* Forms module
* Elements defined here:
*       form, label, input, select, optgroup, option,
*       textarea, fieldset, legend, button
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-form-1.xsd"/>

<annotation>
<documentation>
/**
* Nameident module
* Attributes defined here:
* name (for retro browsers)
*
*/
</documentation>
</annotation>
    <include schemaLocation="opt/xhtml-nameident-1.xsd"/>

<annotation>
<documentation>
/**
* Legacy module
* Elements defined here:
*     font, basefont, center, s, strike, u, dir, menu, isindex
*     frame, frameset, noframes, iframe
*          (plus additional datatypes and attributes)
*
*/
</documentation>
</annotation>

<include  schemaLocation="opt/xhtml-legacy-1.xsd" />
    <annotation>
        <documentation>
/**
* Basic modules for forms and tables
*
* Note that to use these modules you must comment out
* the XHTML Forms and Tables modules.
*
*/
</documentation>
</annotation>
    <annotation>
        <documentation>
/**
* Basic Forms module
*
* Note that this module is not used in XHTML. It is designed
* for use with XHTML-Basic.
*
*Elements defined here:
```

```
* form, label, input, select, option, textarea
*
*/
    </documentation>
    </annotation>
<!-- include  schemaLocation="opt/basic/xhtml-basic-form-1.xsd" / -->
<annotation>
<documentation>
/**
* Basic Tables module
*
* Note that this module is not used in XHTML. It is designed
* for use with XHTML-Basic.
*
*Elements defined here:
* form, label, input, select, option, textarea
*
*/
</documentation>
</annotation>
<!-- include  schemaLocation="opt/basic/xhtml-basic-table-1.xsd" / -->

</schema>
```

D.5.11. Core Hub Module

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>

<xsd:annotation>
    <xsd:documentation>
/**
* This is the XML Schema Core Elements module for XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ELEMENTS XHTML Core 1.0//EN"
*
*/
    </xsd:documentation>

    <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
    </xsd:documentation>

    <xsd:documentation source="xhtml-copyright-1.txt"/>

    <xsd:documentation>
/**
```

```
* Required core modules for elements
*
*        +  text
*        +  hypertext
*        +  lists
*        +  ruby
*        +  structural
*
* This module includes only other modules and is required.
*
*
*/
    </xsd:documentation>
</xsd:annotation>

    <xsd:annotation>
        <xsd:documentation>
/**
* Text module
*
*    The Text module includes declarations for all core
*     text container elements and their attributes.
*
*        +  block phrasal
*        +  block structural
*        +  inline phrasal
*        +  inline structural
*
* Elements defined here:
*        address, blockquote, pre, h1, h2, h3, h4, h5, h6
*        div, p
*        abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
*        br,span
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="core/xhtml-text-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Hypertext module
* Elements defined here:
*        a
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="core/xhtml-hypertext-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Lists module
* Elements defined here:
*        dt, dd, dl, ol, ul, li
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="core/xhtml-list-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
* Ruby module
* Elements defined here:
*                ruby, rbc, rtc, rb, rt, rp
*/
        </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="core/xhtml-ruby-1.xsd"/>

    <xsd:annotation>
        <xsd:documentation>
/**
```

```
* Structural module
* Elements defined here:
*        title, head, body, html
*
*/
    </xsd:documentation>
    </xsd:annotation>
    <xsd:include schemaLocation="core/xhtml-struct-1.xsd"/>

</xsd:schema>
```

D.5.12. XHTML 1.1 Content Model

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
            targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
            xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema http://www.w3.org/2000/10/XMLSchema.xsd"
            elementFormDefault="unqualified"
            version="1.1"
>
    <xsd:annotation>
        <xsd:documentation>
/**
* This is the XML Schema module common content models in XHTML
*  Please use this formal public identifier to identify it:
*          "-//W3C//ENTITIES XHTML Document Model 1.0//EN"
* And this namespace for XHTML elements:
*        xmlns:xhtml="http://www.w3.org/1999/xhtml"
*
*/
  </xsd:documentation>
        <xsd:documentation>
/**
*
* Versioning block

* Author: Daniel Austin
* $RCSfile: xhtml-m12n-schema.ps,v $
* $Revision: 1.2 $
* $Date: 2001/03/22 16:14:55 $
* $Author: shane $
* (remove the NO below to see the full revision log)
* Log: $NOLog: $
*
*/
  </xsd:documentation>
        <xsd:documentation source="xhtml-copyright-1.txt"/>
        <xsd:documentation>
/**
*   XHTML Document Model
*
*     This module describes the groupings of elements that make up
*     common content models for XHTML elements.
*
*     XHTML has three basic content models:
*
*         Inline.mix;  character-level elements
*         Block.mix;   block-like elements, eg., paragraphs and lists
*         Flow.mix;    any block or inline elements
*
*    Any groups declared in this module may be used
*     to create element content models, but the above three are
*     considered 'global' (insofar as that term applies here).
*
*/
  </xsd:documentation>
    </xsd:annotation>
    <xsd:annotation>
        <xsd:documentation>
/**
```

```
* Extending the Model
*
*    While in some cases this module may need to be rewritten to
*     accommodate changes to the document model, minor extensions
*     may be accomplished by redeclaring any of the three *.extra;
*     groups to contain extension element types as follows:
*
*         Misc.extra    whose parent may be any block or inline element.
*
*         Inline.extra  whose parent may be any inline element.
*
*         Block.extra   whose parent may be any block element.
*
*/
  </xsd:documentation>
    </xsd:annotation>
    <!--   Optional Elements in head -->
    <xsd:group name="HeadOpts.mix">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="script" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="style"/>
            <xsd:element ref="meta"/>
            <xsd:element ref="link"/>
            <xsd:element ref="object"/>
        </xsd:choice>
    </xsd:group>
    <!--  Miscellaneous Elements-->
    <!--
        ins and del are used to denote editing changes
    -->
    <xsd:group name="Edit.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="ins"/>
            <xsd:element ref="del"/>
        </xsd:choice>
    </xsd:group>
    <!--
        script and noscript are used to contain scripts
        and alternative content
    -->
    <xsd:group name="Script.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="script"/>
            <xsd:element ref="noscript"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="Misc.extra">
        <xsd:choice minOccurs="0" maxOccurs="unbounded"/>
    </xsd:group>
    <!--
        These elements are neither block nor inline, and can
        essentially be used anywhere in the document body.
    -->
    <xsd:group name="Misc.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="Edit.class"/>
            <xsd:group ref="Script.class"/>
            <xsd:group ref="Misc.extra"/>
        </xsd:choice>
    </xsd:group>
    <!--  Inline Elements -->
    <xsd:group name="InlStruct.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="br"/>
            <xsd:element ref="span"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="InlPhras.class">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="em"/>
            <xsd:element ref="strong"/>
            <xsd:element ref="dfn"/>
            <xsd:element ref="code"/>
            <xsd:element ref="samp"/>
```

```
        <xsd:element ref="kbd"/>
        <xsd:element ref="var"/>
        <xsd:element ref="cite"/>
        <xsd:element ref="abbr"/>
        <xsd:element ref="acronym"/>
        <xsd:element ref="q"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="InlPres.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="tt"/>
        <xsd:element ref="i"/>
        <xsd:element ref="b"/>
        <xsd:element ref="big"/>
        <xsd:element ref="small"/>
        <xsd:element ref="sub"/>
        <xsd:element ref="sup"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="I18n.class">
    <xsd:sequence>
        <xsd:element ref="bdo"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="Anchor.class">
    <xsd:sequence>
        <xsd:element ref="a"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="InlSpecial.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="img"/>
        <xsd:element ref="map"/>
        <xsd:element ref="applet"/>
        <xsd:element ref="object"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="InlForm.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="input"/>
        <xsd:element ref="select"/>
        <xsd:element ref="textarea"/>
        <xsd:element ref="label"/>
        <xsd:element ref="button"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="Inline.extra">
    <xsd:choice minOccurs="0" maxOccurs="unbounded"/>
</xsd:group>
<!-- ruby is required in XHTML -->
<xsd:group name="Ruby.class">
    <xsd:sequence>
        <xsd:element ref="ruby"/>
    </xsd:sequence>
</xsd:group>
<!--
    Inline.class includes all inline elements,
    used as a component in mixes
-->
<xsd:group name="Inline.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="InlPhras.class"/>
        <xsd:group ref="InlPres.class"/>
        <xsd:group ref="I18n.class"/>
        <xsd:group ref="Anchor.class"/>
        <xsd:group ref="InlSpecial.class"/>
        <xsd:group ref="InlForm.class"/>
        <xsd:group ref="Ruby.class"/>
        <xsd:group ref="Inline.extra"/>
    </xsd:choice>
</xsd:group>
<!--
    InlNoRuby.class includes all inline elements
    except ruby, used as a component in mixes
```

```
-->
<xsd:group name="InlNoRuby.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="InlPhras.class"/>
        <xsd:group ref="InlPres.class"/>
        <xsd:group ref="I18n.class"/>
        <xsd:group ref="Anchor.class"/>
        <xsd:group ref="InlSpecial.class"/>
        <xsd:group ref="InlForm.class"/>
        <xsd:group ref="Inline.extra"/>
    </xsd:choice>
</xsd:group>

<!--
    NoRuby.content includes all inlines except ruby
-->
<xsd:complexType name="NoRuby.content" mixed="true">
    <xsd:sequence>
        <xsd:group ref="InlNoRuby.class"/>
        <xsd:group ref="Misc.class"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    InlNoAnchor.class includes all non-anchor inlines,
    used as a component in mixes
-->
<xsd:group name="InlNoAnchor.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="InlPhras.class"/>
        <xsd:group ref="InlPres.class"/>
        <xsd:group ref="I18n.class"/>
        <xsd:group ref="InlSpecial.class"/>
        <xsd:group ref="InlForm.class"/>
        <xsd:group ref="Ruby.class"/>
        <xsd:group ref="Inline.extra"/>
    </xsd:choice>
</xsd:group>
<!--
    InlNoAnchor.mix includes all non-anchor inlines
-->
<xsd:group name="InlNoAnchor.mix">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="InlNoAnchor.class"/>
        <xsd:group ref="Misc.class"/>
    </xsd:choice>
</xsd:group>
<!--
    Inline.mix includes all inline elements, including Misc.class
-->
<xsd:group name="Inline.mix">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="Inline.class"/>
        <xsd:group ref="Misc.class"/>
    </xsd:choice>
</xsd:group>
<!--  Block Elements  -->
<!--
    In the HTML 4.0 DTD, heading and list elements were included
    in the block group. The Heading.class and
    List.class groups must now be included explicitly
    on element declarations where desired.
-->
<xsd:group name="Heading.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="h1"/>
        <xsd:element ref="h2"/>
        <xsd:element ref="h3"/>
        <xsd:element ref="h4"/>
        <xsd:element ref="h5"/>
        <xsd:element ref="h6"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="List.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```
        <xsd:element ref="ul"/>
        <xsd:element ref="ol"/>
        <xsd:element ref="dl"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="BlkStruct.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="p"/>
        <xsd:element ref="div"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="BlkPhras.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="pre"/>
        <xsd:element ref="blockquote"/>
        <xsd:element ref="address"/>
    </xsd:choice>
</xsd:group>
<!-- this is omitted in the DTD -->
<xsd:group name="BlkPres.class">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="hr"/>
    </xsd:sequence>
</xsd:group>
<xsd:group name="BlkSpecial.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="table"/>
        <xsd:element ref="form"/>
        <xsd:element ref="fieldset"/>
    </xsd:choice>
</xsd:group>
<!-- this is a substitution group -->
<xsd:group name="Block.extra">
    <xsd:choice minOccurs="0" maxOccurs="unbounded"/>
</xsd:group>
<!--
    Block.class includes all block elements,
    used as an component in mixes
-->
<xsd:group name="Block.class">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="BlkStruct.class"/>
        <xsd:group ref="BlkPhras.class"/>
        <xsd:group ref="BlkPres.class"/>
        <xsd:group ref="BlkSpecial.class"/>
        <xsd:group ref="Block.extra"/>
    </xsd:choice>
</xsd:group>
<!--
    Block.mix includes all block elements plus %Misc.class;
-->
<xsd:group name="Block.mix">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="List.class"/>
        <xsd:group ref="Block.class"/>
        <xsd:group ref="Misc.class"/>
    </xsd:choice>
</xsd:group>
<!--   All Content Elements -->
<!--
    Flow.mix includes all text content, block and inline
    Note that the "any" element included here allows us
    to add data from any other namespace, a necessity
    for compound document creation.
    Note however that it is not possible to add
    to any head level element without further
    modification. To add RDF metadata to the head
    of a document, modify the struct module.
-->
<xsd:group name="Flow.mix">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="List.class"/>
        <xsd:group ref="Block.class"/>
        <xsd:group ref="Inline.class"/>
```

```
            <xsd:group ref="Misc.class"/>
            <xsd:any namespace="##other"/>
        </xsd:choice>
    </xsd:group>
</xsd:schema>
```