Modularization of XHTML™ in XML Schema

W3C Working Draft 9 December 2002

This version:
http://www.w3.org/TR/2002/WD-xhtml-m12n-schema-20021209
Latest version:
http://www.w3.org/TR/xhtml-m12n-schema
Previous version:
http://www.w3.org/TR/2002/WD-xhtml-m12n-schema-20020815
Diff-marked version:
xhtml-m12n-schema-diff.html
Editors:
Daniel Austin, W. W. Grainger, Inc.
Shane McCarron, Applied Testing and Technology, Inc.
Masayasu Ishikawa, W3C

This document is also available in these non-normative formats: Single XHTML file [p.1] , PostScript version, PDF version, ZIP archive, and Gzip'd TAR archive.

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 Last Call 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. After going through Last Call, this document will be integrated into a new edition of *Modularization of XHTML* [XHTMLMOD [p.42]]. The Last Call review period ends 31 January 2003.

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document 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.

Patent disclosures relevant to this specification may be found on the Working Group's public patent disclosure page.

Please send review comments to www-html-editor@w3.org (archive). Public discussion on XHTML 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.

Quick Table of Contents

Full Table of Contents

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.42]]. There are currently several public language variants in the XHTML namespace, including XHTML 1.0 [XHTML1 [p.41]] (which includes variants corresponding to the definitions of "Strict", "Transitional", and "Frameset") and XHTML Basic [XHTMLBASIC [p.41]]. The development of DTD-based modularization for XHTML made it possible to refashion XHTML 1.0 in a modularized way [XHTMLMOD [p.42]], resulting in XHTML Basic and XHTML 1.1 [XHTML11 [p.41]]. (Here and throughout this document, the term "XHTML-MOD" is used to refer to [XHTMLMOD [p.42]].)

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.41]].

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

Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules. [Booch94 [p.41]]

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.42]]. 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 [XMLNAMES [p.42]], 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. In the XML Schema modules described here, named character entities for XHTML are included using a DTD.

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="Heading.class"/> <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.41]]. Notations for XHTML are preserved in the Schema modules using the notation element in a straightforward way.

Example 8 - Notations

<table>
<tr><th>DTD</th><th>Schema</th></tr>
<tr><td><pre><!NOTATION character
   PUBLIC "-//W3C//NOTATION
   XHTML Datatype: Character//EN"></pre></td><td><pre><notation name="charset"
   public="-//W3C//NOTATION
   XHTML Datatype: Charset//EN"/></pre></td></tr>
</table>

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

<table>
<tr><th>DTD</th><th>Schema</th></tr>
<tr><td><pre><!ENTITY % Length.datatype "CDATA" ></pre></td><td><pre><simpleType name="Length">
   <restriction base="string"/>
</simpleType></pre></td></tr>
</table>

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 using a DTD reference to three individual DTD modules that define them.

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	DTD 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" [XMLSCHEMA_COMPOSITION [p.42]] 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 Schemas

The modularized XHTML schema uses three types of modules, which when combined comprise the entire XHTML 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 modules. The hub document corresponds to the DTD "driver" module 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) example hub document described here contains <include> elements for two modules, named "required" and "optional". Each of these included modules is itself a container module.

2.2.1.2. Container Modules

Module containers, 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 containers.

2.2.1.3. Element modules

In addition to the module containers listed above, there are around forty schema modules which contain only element definitions and their associated attribute and content model definitions. By convention, Schema modularizations 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, it is useful to provide here a module naming convention syntax. This syntax also provides a simple means of distinguishing modules 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 modules
- Identify the language version of the module

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

Example 10 - 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 content 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.41]].

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 modules 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 module 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 XHTML 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 module'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 module.

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.

ModuleF-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.

None of the modules defined here should be modified by developers; instead use <redefine> or a substitution group.

Schema location	SCHEMA/req/xhtml-framework-1.xsd [p.49]
Use	Required
Type	Module Container
Description	Required XHTML modules
Contents	SCHEMA/req/xhtml-framework-1.xsd [p.49] SCHEMA/req/xhtml-core-1.xsd [p.131]
Redefinitions	No
Dependencies	None

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.49]
Use	Required
Type	Module Container
Description	Language support modules
Contents	SCHEMA/req/framework/xhtml-notations-1.xsd [p.51] SCHEMA/req/framework/xhtml-datatypes-1.xsd [p.53] SCHEMA/req/framework/xhtml-attribs-1.xsd [p.55] SCHEMA/req/framework/xhtml11-model-1.xsd [p.133] 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.51]
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.53]
Use	Required
Type	Language Support - common data types
Contents	XHTML data type definitions
Redefinitions	No
Dependencies	None

3.2.1.1.3. 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

3.2.1.1.4. Common Content Models

Schema location	SCHEMA/req/framework/xhtml11-model-1.xsd [p.133]
Use	Required
Type	Language Support - common content model groups
Contents	Abstract content models
Redefinitions	No
Dependencies	Element definitions

3.2.1.1.5. Character Entities

The character entities module includes three DTD modules, 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.131]
Use	Required
Type	Module Container
Description	Core element modules
Contents	SCHEMA/req/core/xhtml-text-1.xsd [p.60] SCHEMA/req/core/xhtml-hypertext-1.xsd [p.61] SCHEMA/req/core/xhtml-list-1.xsd [p.62] 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.60]
Use	Required
Type	Module Container
Description	Text element modules
Contents	SCHEMA/req/core/text/xhtml-blkphras-1.xsd [p.109] SCHEMA/req/core/text/xhtml-blkstruct-1.xsd [p.112] SCHEMA/req/core/text/xhtml-inlphras-1.xsd [p.113] SCHEMA/req/core/text/xhtml-inlstruct-1.xsd [p.116]
Redefinitions	No
Dependencies	None

Block Phrasal

Schema location	SCHEMA/req/core/text/xhtml-blkphras-1.xsd [p.109]
Use	Required
Type	Element definitions
Redefinitions	No
Dependencies	None
Elements	address [p.110] blockquote [p.110] h1 [p.111] h2 [p.111] h3 [p.111] h4 [p.111] h5 [p.111] h6 [p.111] pre [p.111]
Redefinitions	No
Dependencies	None

Block Structural

Schema location	SCHEMA/req/core/text/xhtml-blkstruct-1.xsd [p.112]
Use	Required
Type	Element definitions
Elements	div [p.113] p [p.113]
Redefinitions	No
Dependencies	None

Inline Phrasal

Schema location	SCHEMA/req/core/text/xhtml-inlphras-1.xsd [p.113]
Use	Required
Type	Element definitions
Elements	abbr [p.114] acronym [p.114] cite [p.114] code [p.114] dfn [p.114] em [p.114] kbd [p.114] q [p.114] samp [p.114] strong [p.114] var [p.114]
Redefinitions	No
Dependencies	None

Inline Structural

Schema location	SCHEMA/req/core/text/xhtml-inlstruct-1.xsd [p.116]
Use	Required
Type	Element definitions
Elements	br [p.116] span [p.116]
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.64] dt [p.63] li [p.64] ol [p.64] ul [p.64]
Redefinitions	No
Dependencies	None

3.2.1.2.4. Structural

Schema location	SCHEMA/req/core/xhtml-struct-1.xsd [p.58]
Use	Required
Type	Element definitions
Elements	body [p.60] head [p.59] html [p.60] 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.125]
Use	Required
Type	Module Container
Description	Optional modules
Contents	SCHEMA/opt/xhtml-edit-1.xsd [p.66] SCHEMA/opt/xhtml-bdo-1.xsd [p.67] SCHEMA/opt/xhtml-link-1.xsd [p.102] SCHEMA/opt/xhtml-meta-1.xsd [p.98] SCHEMA/opt/xhtml-base-1.xsd [p.103] SCHEMA/opt/xhtml-script-1.xsd [p.99] SCHEMA/opt/xhtml-style-1.xsd [p.100] SCHEMA/opt/xhtml-image-1.xsd [p.86] SCHEMA/opt/xhtml-csismap-1.xsd [p.87] SCHEMA/opt/xhtml-ssismap-1.xsd [p.90] SCHEMA/opt/xhtml-param-1.xsd [p.117] SCHEMA/opt/xhtml-applet-1.xsd [p.91] SCHEMA/opt/xhtml-object-1.xsd [p.92] SCHEMA/opt/xhtml-table-1.xsd [p.81] SCHEMA/opt/xhtml-form-1.xsd [p.71] SCHEMA/opt/xhtml-nameident-1.xsd [p.104] SCHEMA/opt/xhtml-legacy-1.xsd [p.105] SCHEMA/opt/frames/xhtml-frames-1.xsd [p.93] SCHEMA/opt/frames/xhtml-target-1.xsd [p.95] SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.97] SCHEMA/req/framework/xhtml-events-1.xsd [p.98] SCHEMA/opt/xhtml-ruby-1.xsd [p.106]
Redefinitions	No
Dependencies	None

3.2.2.1. Edit

Schema location	SCHEMA/opt/xhtml-edit-1.xsd [p.66]
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.67]
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.111] SCHEMA/opt/pres/xhtml-inlpres-1.xsd [p.115]

3.2.2.3.1. Inline Presentational

Schema location	SCHEMA/opt/pres/xhtml-inlpres-1.xsd [p.115]
Use	Optional
Type	Element definitions
Elements	b [p.115] big [p.115] i [p.115] small [p.115] sub [p.115] sup [p.115] tt [p.115]
Redefinitions	No
Dependencies	None

3.2.2.3.2. Block Presentational

Schema location	SCHEMA/opt/pres/xhtml-blkpres-1.xsd [p.111]
Use	Optional
Type	Element definitions
Elements	hr [p.112]
Redefinitions	No
Dependencies	None

3.2.2.4. Link

Schema location	SCHEMA/opt/xhtml-link-1.xsd [p.102]
Use	Optional
Type	Element definitions
Elements	link [p.103]
Redefinitions	No
Dependencies	None

3.2.2.5. Meta

Schema location	SCHEMA/opt/xhtml-meta-1.xsd [p.98]
Use	Optional
Type	Element definitions
Elements	meta [p.99]
Redefinitions	No
Dependencies	None

3.2.2.6. Base

Schema location	SCHEMA/opt/xhtml-base-1.xsd [p.103]
Use	Optional
Type	Element definitions
Elements	base [p.103]
Redefinitions	No
Dependencies	None

3.2.2.7. Scripting

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

3.2.2.8. Style

Schema location	SCHEMA/opt/xhtml-style-1.xsd [p.100]
Use	Optional
Type	Element definitions
Elements	style [p.101]
Redefinitions	No
Dependencies	None

3.2.2.9. Image

Schema location	SCHEMA/opt/xhtml-image-1.xsd [p.86]
Use	Optional
Type	Element definitions
Elements	img [p.87]
Redefinitions	No
Dependencies	None

3.2.2.10. Client-side Image Maps

Schema location	SCHEMA/opt/xhtml-csismap-1.xsd [p.87]
Use	Optional
Type	Element definitions
Elements	area [p.89] map [p.90]
Redefinitions	No
Dependencies	None

3.2.2.11. Server-side Image Maps

Schema location	SCHEMA/opt/xhtml-ssismap-1.xsd [p.90]
Use	Optional
Type	Attribute definitions
Redefinitions	No
Dependencies	None

3.2.2.12. Param

Schema location	SCHEMA/opt/xhtml-param-1.xsd [p.117]
Use	Optional
Type	Element definitions
Elements	param [p.117]
Redefinitions	No
Dependencies	None

3.2.2.13. Applet

Schema location	SCHEMA/opt/xhtml-applet-1.xsd [p.91]
Use	Optional
Type	Element definitions
Elements	applet [p.92]
Redefinitions	No
Dependencies	Param::param

3.2.2.14. Object

Schema location	SCHEMA/opt/xhtml-object-1.xsd [p.92]
Use	Optional
Type	Element definitions
Elements	object [p.93]
Redefinitions	No
Dependencies	Param::param

3.2.2.15. Tables

Schema location	SCHEMA/opt/xhtml-table-1.xsd [p.81]
Use	Optional
Type	Element definitions
Elements	caption [p.85] col [p.84] colgroup [p.84] table [p.86] tbody [p.84] td [p.83] tfoot [p.85] th [p.83] thead [p.85] tr [p.83]
Redefinitions	No
Dependencies	None

3.2.2.16. Forms

Schema location	SCHEMA/opt/xhtml-form-1.xsd [p.71]
Use	Optional
Type	Element definitions
Elements	button [p.78] fieldset [p.77] form [p.72] input [p.74] label [p.73] legend [p.77] optgroup [p.75] option [p.76] select [p.75] textarea [p.76]
Redefinitions	No
Dependencies	None

3.2.2.17. Nameident

Schema location	SCHEMA/opt/xhtml-nameident-1.xsd [p.104]
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.105]
Use	Optional
Type	Module container
Contents	SCHEMA/opt/misc/xhtml-misc-1.xsd [p.118] SCHEMA/opt/xhtml-framedefs-1.xsd [p.124]
Redefinitions	No
Dependencies	None

3.2.2.18.1. Misc

Schema location	SCHEMA/opt/misc/xhtml-misc-1.xsd [p.118]
Use	Optional
Type	Element definitions
Elements	basefont [p.119] center [p.119] dir [p.120] font [p.119] isindex [p.120] menu [p.120] s [p.119] strike [p.119] u [p.120]
Redefinitions	Yes
Dependencies	Yes

3.2.2.18.2. Framedefs

Schema location	SCHEMA/opt/xhtml-framedefs-1.xsd [p.124]
Use	Optional
Type	Element definitions
Contents	SCHEMA/opt/xhtml-frames-1.xsd [p.124] SCHEMA/opt/frames/xhtml-target-1.xsd [p.95] SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.97]
Redefinitions	Yes
Dependencies	Yes

Frames

Schema location	SCHEMA/opt/frames/xhtml-frames-1.xsd [p.93]
Use	Optional
Type	Element definitions
Elements	frame [p.95] frameset [p.94] noframes [p.95]
Redefinitions	Yes
Dependencies	Target::target

Target

Schema location	SCHEMA/opt/frames/xhtml-target-1.xsd [p.95]
Use	Optional
Type	Attribute redefinitions
Redefinitions	Yes
Dependencies	Yes

Iframe

Schema location	SCHEMA/opt/frames/xhtml-iframe-1.xsd [p.97]
Use	Optional
Type	Element definitions
Elements	iframe [p.98]
Redefinitions	Yes
Dependencies	Target::target

3.2.2.19. Basic Forms

Schema location	SCHEMA/opt/xhtml-basic-form-1.xsd [p.68]
Use	Optional
Type	Element definitions
Elements	form [p.69] input [p.70] label [p.69] option [p.71] select [p.70] textarea [p.71]
Redefinitions	No
Dependencies	Removal of Forms

3.2.2.20. Basic Tables

Schema location	SCHEMA/opt/xhtml-basic-table-1.xsd [p.78]
Use	Optional
Type	Element definitions
Elements	caption [p.79] table [p.79] td [p.80] th [p.80] tr [p.80]
Redefinitions	No
Dependencies	Removal of Tables

3.2.2.21. Events

Schema location	SCHEMA/req/framework/xhtml-events-1.xsd [p.98]
Use	Required
Type	Language Support - common events attributes
Contents	Common events attributes for XHTML
Redefinitions	Yes
Dependencies	Element definitions

3.2.3. Ruby

Ruby elements denote annotations used in some Asian languages [RUBY [p.41]].

The Ruby module has been moved into the optional element definitions module. Note that it is normatively required in XHTML 1.1

Schema location	SCHEMA/opt/xhtml-ruby-1.xsd [p.106]
Use	Required
Type	Element definitions
Elements	rb [p.109] rbc [p.108] rp [p.109] rt [p.109] rtc [p.108] ruby [p.108]
Redefinitions	No
Dependencies	None

3.2.4. XHTML Hub Document (Non-normative)

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

3.2.4.1. 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.49] SCHEMA/xhtml-optional-1.xsd [p.125]

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.

A. References

This appendix is *normative*.

A.1. Normative References

[APPROACH]
: *An Approach to the Modularization of XHTML using XML Schema*, R. Jelliffe, Academia Sinica Computing Center, 19 December 2000.
Available at: http://www.ascc.net/~ricko/xhtml.htm

[Booch94]
: *Object-Oriented Analysis and Design*, G. Booch, 1994

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

[RFC2045]
: *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, N. Freed *et al.*, November 1996.
Available at: http://www.rfc-editor.org/rfc/rfc2045.txt
Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[RFC3066]
: *Tags for the Identification of Languages*, H. Alvestrand, January 2001.
Available at: http://www.rfc-editor.org/rfc/rfc3066.txt

[RUBY]
: *Ruby Annotation*, M. Sawicki *et al.*, 31 May 2001
Available at: http://www.w3.org/TR/2001/REC-ruby-20010531

[URI]
: *Uniform Resource Identifiers (URI): Generic Syntax*, T. Berners-Lee *et al.*, IETF, August 1998.
Available at: http://www.rfc-editor.org/rfc/rfc2396.txt.

[XHTML1]
: *XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)*, W3C Recommendation, S. Pemberton, *et al.*, 26 January 2000, revised 1 August 2002.
Available at: http://www.w3.org/TR/2002/REC-xhtml1-20020801

[XHTML11]
: *XHTML 1.1 - Module-based XHTML*, W3C Recommendation, M. Altheim, *et al.*, 31 May 2001.
Available at: http://www.w3.org/TR/2001/REC-xhtml11-20010531

[XHTMLBASIC]
: *XHTML Basic*, W3C Recommendation, M. Baker, *et al.*, 19 December 2000.
Available at: http://www.w3.org/TR/2000/REC-xhtml-basic-20001219

[XHTMLMIME]
: *The application/xhtml+xml Media Type*, M. Baker, P. Stark, IETF, January 2002.
Available at: http://www.rfc-editor.org/rfc/rfc3236.txt

[XHTMLMOD]
Modularization of XHTML, W3C Recommendation, M. Altheim, *et al.*, 10 April 2001
Available at: http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410
[XML]
Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, T. Bray *et al.*, 6 October 2000.
Available at: http://www.w3.org/TR/2000/REC-xml-20001006
[XMLNAMES]
Namespaces in XML, W3C Recommendation, T. Bray *et al.*, 14 January 1999.
Available at: http://www.w3.org/TR/1999/REC-xml-names-19990114
[XMLSCHEMA]
XML Schema Part 1: Structures, W3C Recommendation, H. S. Thompson, *et al.*, 2 May 2001
Available at: http://www.w3.org/TR/2001/REC-xmlschema-1-20010502
[XMLSCHEMA_COMPOSITION]
XML Schema Part 1: Structures, "4.2 Layer 2: Schema Documents, Namespaces and Composition", W3C Recommendation, H. S. Thompson, *et al.*, 2 May 2001
Available at: http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/#layer2

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
- Removed default = "get" from form.action
- Removed spurious form.button.type attribute
- Removed default value rect from csismap module
- Removed default value "data" from param.valuetype

C. Acknowledgements

This appendix is *informative*.

The following people provided support and assistance:

- Peter Stark
- Steven Pemberton
- Rick Jelliffe
- Josef Dietl
- Sebastian Schnitzenbaumer
- Malte Wedel
- 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.42]] 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" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
```

```
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema driver for XHTML 1.1.
      Please use this namespace for XHTML elements:

      "http://www.w3.org/1999/xhtml"

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      This is XHTML, a reformulation of HTML as a modular XML application
      The Extensible HyperText Markup Language (XHTML)
      Copyright &#169;1998-2002 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 Schema
      modules and their accompanying xs: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
      these XML Schema modules for any purpose.

      They are provided "as is" without expressed or implied warranty.
    </xs:documentation>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      This schema includes two modules, one
      for the XHTML required elements and attributes,
      and one for the optional elements and attributes.
      In order to modify this schema, you can modify the files
      containing the optional modules. It is not necessary to
      modify this file. All of the required modules must be
      included without change for conformance.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="xhtml-required-1.xsd">
    <xs:annotation>
      <xs:documentation>
        These modules are required to be included unchanged for
        XHTML conformance.

        Framework modules:
            +  notations
            +  datatypes
            +  xlink
            +  events
```

```
            +   common attributes
            +   common content models
            +   character entities

        Core required elements modules:
            +   text
            +   hypertext
            +   lists
            +   structure
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="xhtml-optional-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Optional Element modules:
            +   Edit
            +   Bdo
            +   Presentational
            +   Link
            +   Meta
            +   Base
            +   Scripting
            +   Style
            +   Image
            +   Client-side image maps
            +   Server-side image maps
            +   Param
            +   Applet
            +   Object
            +   Basic tables
            +   Tables
            +   Basic forms
            +   Forms
            +   Basic Ruby
            +   Ruby
            +   Nameident
            +   Legacy (includes frames)
      </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs: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"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
```

```
          elementFormDefault="unqualified"
          attributeFormDefault="unqualified"
          blockDefault="#all"
          finalDefault="#all">

<xs:annotation>
  <xs:documentation>
    This is the XML Schema Framework module for XHTML

    @author: Daniel Austin austin.d@ic.grainger.com
    $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
  </xs:documentation>
  <xs:documentation source="../xhtml-copyright-1.xsd"/>
</xs:annotation>

<xs:annotation>
  <xs:documentation>
    Modular Framework

    This required module instantiates the modules needed
    to support the XHTML modularization model, including:

      +  notations
      +  datatypes
      +  common attributes
      +  common content models
      +  character entities
  </xs:documentation>
</xs:annotation>

<xs:include schemaLocation="framework/xhtml-notations-1.xsd">
  <xs:annotation>
    <xs:documentation>
       Notations module
       Contains XHTML notations for data types
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="framework/xhtml-datatypes-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Datatypes module
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="framework/xhtml-attribs-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Common attributes module
    </xs:documentation>
  </xs:annotation>
</xs:include>

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

```
      <xs:documentation>
        Common content models module
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="framework/xhtml-charent-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Character entities module

        Notation declarations for Latin 1,
        Special, and Symbol character entity sets

        Entities are not supported in XML Schema
    </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs: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"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema module for data type notations for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Notations module

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

```
<!-- W3C XML 1.0 Recommendation -->
<xs:notation name="w3c-xml"
  public="ISO 8879//NOTATION Extensible Markup Language (XML) 1.0//EN"/>
<!-- XML 1.0 CDATA -->
<xs:notation name="cdata" public="-//W3C//NOTATION XML 1.0: CDATA//EN"/>
<!-- SGML Formal Public Identifiers -->
<xs: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, as per section 2.2 of [XML] -->
<xs:notation name="character"
  public="-//W3C//NOTATION XHTML Datatype: Character//EN"/>

<!-- a character encoding, as per [RFC2045] -->
<xs:notation name="charset"
  public="-//W3C//NOTATION XHTML Datatype: Charset//EN"/>

<!-- a space separated list of character encodings, as per [RFC2045] -->
<xs:notation name="charsets"
  public="-//W3C//NOTATION XHTML Datatype: Charsets//EN"/>

<!-- media type, as per [RFC2045] -->
<xs:notation name="contentType"
  public="-//W3C//NOTATION XHTML Datatype: ContentType//EN"/>

<!-- comma-separated list of media types, as per [RFC2045] -->
<xs:notation name="contentTypes"
  public="-//W3C//NOTATION XHTML Datatype: ContentTypes//EN"/>

<!-- date and time information. ISO date format -->
<xs:notation name="datetime"
  public="-//W3C//NOTATION XHTML Datatype: Datetime//EN"/>

<!-- a language code, as per [RFC3066] -->
<xs:notation name="languageCode"
  public="-//W3C//NOTATION XHTML Datatype: LanguageCode//EN"/>

<!-- nn for pixels or nn% for percentage length -->
<xs:notation name="length"
  public="-//W3C//NOTATION XHTML Datatype: Length//EN"/>

<!-- space-separated list of link types -->
<xs:notation name="linkTypes"
  public="-//W3C//NOTATION XHTML Datatype: LinkTypes//EN"/>

<!-- single or comma-separated list of media descriptors -->
<xs:notation name="mediaDesc"
  public="-//W3C//NOTATION XHTML Datatype: MediaDesc//EN"/>

<!-- pixel, percentage, or relative -->
<xs:notation name="multiLength"
```

```
    public="-//W3C//NOTATION XHTML Datatype: MultiLength//EN"/>

  <!-- one or more digits (NUMBER) -->
  <xs:notation name="number"
    public="-//W3C//NOTATION XHTML Datatype: Number//EN"/>

  <!-- one or more digits (NUMBER) -->
  <xs:notation name="pixels"
    public="-//W3C//NOTATION XHTML Datatype: Pixels//EN"/>

  <!-- script expression -->
  <xs:notation name="script"
    public="-//W3C//NOTATION XHTML Datatype: Script//EN"/>

  <!-- textual content -->
  <xs:notation name="text" public="-//W3C//NOTATION XHTML Datatype: Text//EN"/>

  <!-- a Uniform Resource Identifier, see [URI] -->
  <xs:notation name="uri" public="-//W3C//NOTATION XHTML Datatype: URI//EN"/>

  <!-- a space-separated list of Uniform Resource Identifiers, see [URI] -->
  <xs:notation name="uris" public="-//W3C//NOTATION XHTML Datatype: URIs//EN"/>

</xs:schema>
```

D.3.2. XHTML Datatypes

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">


  <xs:annotation>
    <xs:documentation>
      This is the XML Schema datatypes module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
   Datatypes
   defines containers for the following datatypes, many of
   these imported from other specifications and standards.
  </xs:documentation>
  </xs:annotation>

  <!-- nn for pixels or nn% for percentage length -->
```

```
<xs:simpleType name="Length">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- space-separated list of link types -->
<xs:simpleType name="LinkTypes">
  <xs:list itemType="xs:NMTOKEN"/>
</xs:simpleType>

<!-- single or comma-separated list of media descriptors -->
<xs:simpleType name="MediaDesc">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- pixel, percentage, or relative -->
<xs:simpleType name="MultiLength">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- one or more digits (NUMBER) -->
<xs:simpleType name="Number">
  <xs:restriction base="xs:nonNegativeInteger"/>
</xs:simpleType>

<!-- integer representing length in pixels -->
<xs:simpleType name="Pixels">
  <xs:restriction base="xs:nonNegativeInteger"/>
</xs:simpleType>

<!-- script expression -->
<xs:simpleType name="Script">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- !!! RGB color expression - this needs a better definition -->
<xs:simpleType name="Color">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- textual content -->
<xs:simpleType name="Text">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- Imported Datatypes  -->
<!-- a single character, as per section 2.2 of [XML] -->
<xs:simpleType name="Character">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- a character encoding, as per [RFC2045] -->
<xs:simpleType name="Charset">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- a space separated list of character encodings, as per [RFC2045] -->
<xs:simpleType name="Charsets">
```

```
    <xs:list itemType="xs:string"/>
  </xs:simpleType>

  <!-- media type, as per [RFC2045] -->
  <xs:simpleType name="ContentType">
    <xs:list itemType="xs:string"/>
  </xs:simpleType>

  <!-- comma-separated list of media types, as per [RFC2045] -->
  <xs:simpleType name="ContentTypes">
    <xs:list itemType="xs:string"/>
  </xs:simpleType>

  <!-- date and time information. ISO date format -->
  <xs:simpleType name="Datetime">
    <xs:restriction base="xs:normalizedString"/>
  </xs:simpleType>

  <!-- formal public identifier, as per [ISO8879] -->
  <xs:simpleType name="FPI">
    <xs:restriction base="xs:normalizedString"/>
  </xs:simpleType>

  <!-- a language code, as per [RFC3066] -->
  <xs:simpleType name="LanguageCode">
    <xs:restriction base="xs:NMTOKEN"/>
  </xs:simpleType>

  <!-- a Uniform Resource Identifier, see [URI] -->
  <xs:simpleType name="URI">
    <xs:restriction base="xs:anyURI"/>
  </xs:simpleType>

  <!-- a space-separated list of Uniform Resource Identifiers, see [URI] -->
  <xs:simpleType name="URIs">
    <xs:list itemType="xs:string"/>
  </xs:simpleType>

  <!-- comma-separated list of MultiLength -->
  <xs:simpleType name="MultiLengths">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

</xs:schema>
```

D.3.3. XHTML Common Attribute Definitions

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">
```

```
  <xs:annotation>
    <xs:documentation>
      This is the XML Schema common attributes module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

<!--
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="http://www.w3.org/2001/xml.xsd"/>
-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
             schemaLocation="xml-attribs-1.xsd">
    <xs:annotation>
      <xs:documentation>
        This import brings in the attributes xml:lang and xml:space in
        the XML namespace.
      </xs:documentation>
     </xs:annotation>
  </xs:import>

  <xs:annotation>
    <xs:documentation>
      Common Attributes

      This module declares many of the common attributes for the XHTML Schema
  </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="id">
    <xs:attribute name="id" type="xs:ID"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="class">
    <xs:attribute name="class" type="xs:NMTOKENS"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="title">
    <xs:attribute name="title" type="xs:string"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="Core.extra.attrib">
    <xs:annotation>
      <xs:documentation>
        add your attribute here
        <!-- xs:anyAttribute namespace="##other"/ -->
      </xs:documentation>
    </xs:annotation>
  </xs:attributeGroup>

  <xs:attributeGroup name="Core.attrib">
    <xs:attributeGroup ref="id"/>
    <xs:attributeGroup ref="class"/>
    <xs:attributeGroup ref="title"/>
```

```
    <xs:attributeGroup ref="Core.extra.attrib"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="BIDI">
    <xs:attribute name="dir">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="ltr"/>
          <xs:enumeration value="rtl"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="I18n.attrib">
    <xs:attributeGroup ref="BIDI"/>
    <xs:attribute ref="xml:lang"/>
  </xs:attributeGroup>

  <!-- intrinsic event attributes declared previously -->
  <xs:attributeGroup name="Common.extra">
    <xs:annotation>
      <xs:documentation>
        add your attributes here
      </xs:documentation>
    </xs:annotation>
  </xs:attributeGroup>

  <xs:attributeGroup name="Common.attrib">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attributeGroup ref="I18n.attrib"/>
    <!-- !!! xs:attributeGroup ref="Events.attrib"/ -->
    <xs:attributeGroup ref="Common.extra"/>
  </xs:attributeGroup>

</xs:schema>
```

D.3.4. XHTML Character Entities

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
This schema module includes three named character entity files.
-->
<!DOCTYPE xs:schema [
<!-- These are the entity sets for ISO Latin 1 characters for the XHTML -->
<!ENTITY % HTMLlat1 PUBLIC
   "-//W3C//ENTITIES Latin 1 for XHTML//EN"
   "xhtml-lat1.ent">
%HTMLlat1;
<!-- These are the entity sets for special characters for the XHTML -->
<!ENTITY % HTMLsymbol PUBLIC
   "-//W3C//ENTITIES Symbols for XHTML//EN"
   "xhtml-symbol.ent">
%HTMLsymbol;
<!-- These are the entity sets for symbol characters for the XHTML -->
<!ENTITY % HTMLspecial PUBLIC
```

```
    "-//W3C//ENTITIES Special for XHTML//EN"
    "xhtml-special.ent">
%HTMLspecial;
]>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Character Entities module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Character Entities for XHTML

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

</xs: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"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">
```

```
<xs:annotation>
  <xs:documentation>
    This is the XML Schema Document Structure module for XHTML

    @author: Daniel Austin austin.d@ic.grainger.com
    $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
</xs:documentation>
  <xs:documentation source="xhtml-copyright-1.xsd"/>
</xs:annotation>

<xs:annotation>
  <xs: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.
  </xs:documentation>
</xs:annotation>

<xs:attributeGroup name="title.attlist">
  <xs:attributeGroup ref="I18n.attrib"/>
</xs:attributeGroup>

<xs:complexType name="title.type" mixed="true">
  <xs:attributeGroup ref="title.attlist"/>
</xs:complexType>

<xs:element name="title" type="title.type"/>

<xs:group name="head.content">
  <xs:sequence>
    <xs:group ref="HeadOpts.mix"/>
    <xs:element ref="title" minOccurs="1"/>
    <xs:group ref="HeadOpts.mix"/>
  </xs:sequence>
</xs:group>

<xs:attributeGroup name="head.attlist">
  <xs:attribute name="profile" type="URI"/>
  <xs:attributeGroup ref="I18n.attrib"/>
</xs:attributeGroup>

<xs:complexType name="head.type">
  <xs:group ref="head.content"/>
  <xs:attributeGroup ref="head.attlist"/>
</xs:complexType>

<xs:element name="head" type="head.type"/>

<xs:attributeGroup name="body.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>
```

```
  <xs:complexType name="body.type" mixed="true">
    <xs:group ref="Block.mix" minOccurs="1" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="body.attlist"/>
  </xs:complexType>

  <xs:element name="body" type="body.type"/>

  <xs:attributeGroup name="html.attlist">
    <xs:attribute name="version" type="FPI" fixed="-//W3C//DTD XHTML 1.1//EN"/>
    <xs:attributeGroup ref="I18n.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="html.type">
    <xs:sequence>
      <xs:element ref="head"/>
      <xs:element ref="body"/>
    </xs:sequence>
    <xs:attributeGroup ref="html.attlist"/>
  </xs:complexType>

  <xs:element name="html" type="html.type"/>

</xs:schema>
```

D.4.1.2. Text

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Text  module for XHTML
      This is a REQUIRED module.

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="text/xhtml-blkphras-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Block Phrasal module
        Elements defined here:

          * address, blockquote, pre, h1, h2, h3, h4, h5, h6
    </xs:documentation>
    </xs:annotation>
  </xs:include>

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

          * div, p
    </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="text/xhtml-inlphras-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Inline Phrasal module
        Elements defined here:

          * abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
    </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="text/xhtml-inlstruct-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Inline Structural module
        Elements defined here:

          * br,span
    </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

D.4.1.3. Hypertext

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
```

```
          attributeFormDefault="unqualified"
          blockDefault="#all"
          finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Hypertext module for XHTML
      This is a REQUIRED module.

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
      </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="a.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="href" type="URI"/>
    <xs:attribute name="charset" type="Charset"/>
    <xs:attribute name="type" type="ContentType"/>
    <xs:attribute name="hreflang" type="LanguageCode"/>
    <xs:attribute name="rel" type="LinkTypes"/>
    <xs:attribute name="rev" type="LinkTypes"/>
    <xs:attribute name="accesskey" type="Character"/>
    <xs:attribute name="tabindex" type="Number"/>
  </xs:attributeGroup>

  <xs:complexType name="a.type" mixed="true">
    <xs:group ref="InlNoAnchor.mix"/>
    <xs:attributeGroup ref="a.attlist"/>
  </xs:complexType>

  <xs:element name="a" type="a.type"/>

</xs:schema>
```

D.4.1.4. Lists

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
```

```
          blockDefault="#all"
          finalDefault="#all">

<xs:annotation>
  <xs:documentation>
    This is the XML Schema Lists module for XHTML

    @author: Daniel Austin austin.d@ic.grainger.com
    $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
  </xs:documentation>
  <xs:documentation source="xhtml-copyright-1.xsd"/>
</xs:annotation>

<xs:annotation>
  <xs:documentation>
    Lists

      * dl, dt, dd, ol, ul, li

    This module declares the list-oriented element types
    and their attributes.
</xs:documentation>
</xs:annotation>

<xs:attributeGroup name="dt.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

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

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

<xs:attributeGroup name="dd.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

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

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

<xs:attributeGroup name="dl.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

<xs:group name="dl.content">
  <xs:choice minOccurs="1" maxOccurs="unbounded">
    <xs:element ref="dt"/>
    <xs:element ref="dd"/>
  </xs:choice>
</xs:group>
```

```
  <xs:complexType name="dl.type" mixed="true">
    <xs:group ref="dl.content" minOccurs="1" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="dl.attlist"/>
  </xs:complexType>

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

  <xs:attributeGroup name="li.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

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

  <xs:attributeGroup name="ol.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

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

  <xs:attributeGroup name="ul.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="ul.type">
    <xs:sequence>
      <xs:element ref="li" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="ul.attlist"/>
  </xs:complexType>

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

</xs:schema>
```

D.4.2. Text Modules

D.4.2.1. Presentation

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
```

```
          blockDefault="#all"
          finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Presentation module for XHTML
      This is a REQUIRED module.

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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
    </xs:documentation>
  </xs:annotation>

 <xs:include schemaLocation="pres/xhtml-blkpres-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Block Presentational module
        Elements defined here:

         * hr
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="pres/xhtml-inlpres-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Inline Presentational module
        Elements defined here:

          * b, big, i, small, sub, sup, tt
    </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

D.4.2.2. Edit

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Editing Markup module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Editing Elements

        * ins, del

      This module declares element types and attributes used to indicate
      inserted and deleted content while editing a document.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="edit.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="cite" type="URI"/>
    <xs:attribute name="datetime" type="Datetime"/>
  </xs:attributeGroup>

  <xs:group name="edit.content">
    <xs:choice>
      <xs:group ref="Flow.mix"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="edit.type" mixed="true">
    <xs:group ref="edit.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="edit.attlist"/>
  </xs:complexType>

  <!-- ins: Inserted Text   -->
  <xs:element name="ins" type="edit.type"/>

  <!-- del: Deleted Text   -->
  <xs:element name="del" type="edit.type"/>

</xs:schema>
```

D.4.2.3. Bi-directional Text

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           xmlns:xml="http://www.w3.org/XML/1998/namespace"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema BDO Element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="bdo.attlist">
    <xs:attribute ref="xml:lang"/>
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attribute name="dir" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="ltr"/>
          <xs:enumeration value="rtl"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

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

  <xs:element name="bdo" type="bdo.type"/>
</xs:schema>
```

D.4.3. Forms

D.4.3.1. Basic Forms

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Basic Forms module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  <xs:annotation>

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

<!--
BlkNoForm.mix includes all non-form block elements,
plus Misc.class
-->
  <xs:group name="BlkNoForm.mix">
    <xs:choice>
      <xs:group ref="Heading.class"/>
      <xs:group ref="List.class"/>
      <xs:group ref="BlkStruct.class"/>
      <xs:group ref="BlkPhras.class"/>
      <xs:group ref="BlkPres.class"/>
      <xs:group ref="Table.class"/>
      <xs:group ref="Block.extra"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>
```

```
    <xs:attributeGroup name="form.attlist">
      <xs:attributeGroup ref="Common.attrib"/>
      <xs:attribute name="action" type="URI"/>
      <xs:attribute name="method" use="optional" default="get">
        <xs:simpleType>
          <xs:restriction base="NMTOKEN">
            <xs:enumeration value="get"/>
            <xs:enumeration value="post"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="enctype" type="ContentType" use="fixed"
        value="application/x-www-form-urlencoded"/>
    </xs:attributeGroup>

    <xs:complexType name="form.type">
      <xs:group ref="BlkNoForm.mix" minOccurs="0" maxOccurs="unbounded"/>
      <xs:attributeGroup ref="form.attlist"/>
    </xs:complexType>

    <xs:element name="form" type="form.type"/>

    <xs:group name="label.content">
      <xs:choice>
        <xs:element ref="input"/>
        <xs:element ref="select"/>
        <xs:element ref="textarea"/>
        <xs:group ref="InlStruct.class.class"/>
        <xs:group ref="InlPhras.class"/>
        <xs:group ref="I18n.class"/>
        <xs:group ref="InlPres.class"/>
        <xs:group ref="InlSpecial.class"/>
        <xs:group ref="Misc.class"/>
      </xs:choice>
    </xs:group>

    <xs:attributeGroup name="label.attlist">
      <xs:attributeGroup ref="Common.attrib"/>
      <xs:attribute name="for" type="Text"/>
      <xs:attribute name="accesskey" type="Character"/>
    </xs:attributeGroup>

    <xs:complexType name="label.type" mixed="true">
      <xs:group ref="label.content" minOccurs="0" maxOccurs="unbounded"/>
      <xs:attributeGroup ref="label.attlist"/>
    </xs:complexType>

    <xs:element name="label" type="label.type"/>

  <!--
  Basic Forms removes 'image' and 'file' input types.
  -->
    <xs:attribute name="type" use="optional" default="text">
      <xs:simpleType>
        <xs:restriction base="NMTOKEN">
          <xs:enumeration value="text"/>
```

```
      <xs:enumeration value="password"/>
      <xs:enumeration value="checkbox"/>
      <xs:enumeration value="radio"/>
      <xs:enumeration value="submit"/>
      <xs:enumeration value="reset"/>
      <xs:enumeration value="hidden"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>

<xs:attributeGroup name="input.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute ref="type"/>
  <xs:attribute name="name" type="Text"/>
  <xs:attribute name="value" type="Text"/>
  <xs:attribute name="checked">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="checked"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="size" type="Number"/>
  <xs:attribute name="maxlength" type="Number"/>
  <xs:attribute name="src" type="URI"/>
  <xs:attribute name="accesskey" type="Character"/>
</xs:attributeGroup>

<xs:complexType name="input.type">
  <xs:attributeGroup ref="input.attlist"/>
</xs:complexType>

<xs:element name="input" type="input.type"/>

<xs:attributeGroup name="select.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="name" type="Text"/>
  <xs:attribute name="size" type="Number"/>
  <xs:attribute name="multiple">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="multiple"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

<xs:complexType name="select.type">
  <xs:sequence>
    <xs:element ref="option" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="select.attlist"/>
</xs:complexType>

<xs:element name="select" type="select.type"/>

<xs:attributeGroup name="option.attlist">
```

```
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="selected" type="Text"/>
    <xs:attribute name="value" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="option.type" mixed="true">
    <xs:attributeGroup ref="option.attlist"/>
  </xs:complexType>

  <xs:element name="option" type="option.type"/>

  <xs:attributeGroup name="textarea.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="name" type="Text"/>
    <xs:attribute name="rows" type="Number"/>
    <xs:attribute name="cols" type="Number"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:complexType name="textarea.type" mixed="true">
    <xs:attributeGroup ref="textarea.attlist"/>
  </xs:complexType>

  <xs:element name="textarea" type="textarea.type"/>

</xs:schema>
```

D.4.3.2. Forms

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Forms module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
```

```
    </xs:documentation>
  </xs:annotation>

  <xs:group name="Table.class">
    <xs:choice>
      <xs:element ref="table"/>
    </xs:choice>
  </xs:group>

<!--
    BlkNoForm.mix includes all non-form block elements,
       plus Misc.class
-->
  <xs:group name="BlkNoForm.mix">
    <xs:choice>
      <xs:group ref="Heading.class"/>
      <xs:group ref="List.class"/>
      <xs:group ref="BlkStruct.class"/>
      <xs:group ref="BlkPhras.class"/>
      <xs:group ref="BlkPres.class"/>
      <xs:group ref="Table.class"/>
      <xs:group ref="Block.extra"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>

  <!-- form: Form Element -->
  <xs:attributeGroup name="form.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="action" type="URI"/>
    <xs:attribute name="method" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="get"/>
          <xs:enumeration value="post"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="enctype" type="ContentType"
      fixed="application/x-www-form-urlencoded"/>
    <xs:attribute name="accept-charset" type="Charsets"/>
    <xs:attribute name="accept" type="ContentTypes"/>
  </xs:attributeGroup>

  <xs:group name="form.content">
    <xs:choice>
      <xs:group ref="BlkNoForm.mix" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="fieldset"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="form.type">
    <xs:group ref="form.content" minOccurs="1" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="form.attlist"/>
  </xs:complexType>

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

```
<!--
    label: Form Field Label Text
    Note: Each label must not contain more than ONE field
-->
  <xs:group name="label.content">
    <xs:choice>
      <xs:element ref="input"/>
      <xs:element ref="select"/>
      <xs:element ref="textarea"/>
      <xs:element ref="button"/>
      <xs:group ref="InlStruct.class"/>
      <xs:group ref="InlPhras.class"/>
      <xs:group ref="I18n.class"/>
      <xs:group ref="InlPres.class"/>
      <xs:group ref="InlSpecial.class"/>
      <xs:group ref="Inline.extra"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>

  <xs:attributeGroup name="label.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="for" type="xs:IDREF"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:complexType name="label.type" mixed="true">
    <xs:group ref="label.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="label.attlist"/>
  </xs:complexType>

  <xs:element name="label" type="label.type"/>

  <!-- input: Form Control -->
  <xs:attributeGroup name="type.attrib">
  <xs:attribute name="type" use="optional" default="text">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="text"/>
        <xs:enumeration value="password"/>
        <xs:enumeration value="checkbox"/>
        <xs:enumeration value="radio"/>
        <xs:enumeration value="submit"/>
        <xs:enumeration value="reset"/>
        <xs:enumeration value="hidden"/>
        <xs:enumeration value="image"/>
        <xs:enumeration value="button"/>
        <xs:enumeration value="file"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  </xs:attributeGroup>

<!--
    attribute 'name' required for all but submit & reset
-->
```

```
<xs:attributeGroup name="input.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attributeGroup ref="type.attrib"/>
  <xs:attribute name="name" type="Text"/>
  <xs:attribute name="value" type="Text"/>
  <xs:attribute name="checked">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="checked"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="readonly">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="readonly"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="size" type="Number"/>
  <xs:attribute name="maxlength" type="Number"/>
  <xs:attribute name="src" type="URI"/>
  <xs:attribute name="alt" type="Text"/>
  <xs:attribute name="tabindex" type="Number"/>
  <xs:attribute name="accesskey" type="Character"/>
  <xs:attribute name="accept" type="ContentTypes"/>
</xs:attributeGroup>

<xs:complexType name="input.type">
  <xs:attributeGroup ref="input.attlist"/>
</xs:complexType>

<xs:element name="input" type="input.type"/>

<!-- select: Option Selector  -->
<xs:attributeGroup name="select.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="name" type="Text"/>
  <xs:attribute name="size" type="Number"/>
  <xs:attribute name="multiple">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="multiple"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="disabled"/>
```

```
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

<xs:group name="select.content">
  <xs:choice>
    <xs:element ref="optgroup"/>
    <xs:element ref="option"/>
  </xs:choice>
</xs:group>

<xs:complexType name="select.type">
  <xs:group ref="select.content" minOccurs="1" maxOccurs="unbounded"/>
  <xs:attributeGroup ref="select.attlist"/>
</xs:complexType>

<xs:element name="select" type="select.type"/>

<!-- optgroup: Option Group  -->
<xs:attributeGroup name="optgroup.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="label" type="Text"/>
</xs:attributeGroup>

<xs:group name="optgroup.content">
  <xs:choice>
    <xs:element ref="option"/>
  </xs:choice>
</xs:group>

<xs:complexType name="optgroup.type">
  <xs:group ref="optgroup.content" minOccurs="1" maxOccurs="unbounded"/>
  <xs:attributeGroup ref="optgroup.attlist"/>
</xs:complexType>

<xs:element name="optgroup" type="optgroup.type"/>

<!-- option: Selectable Choice  -->
<xs:attributeGroup name="option.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="selected">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="selected"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
```

```
        <xs:restriction base="xs:token">
          <xs:enumeration value="disabled"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="label" type="Text"/>
    <xs:attribute name="value" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="option.type" mixed="true">
    <xs:attributeGroup ref="option.attlist"/>
  </xs:complexType>

  <xs:element name="option" type="option.type"/>

  <!-- textarea: Multi-Line Text Field  -->
  <xs:attributeGroup name="textarea.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="name" type="Text"/>
    <xs:attribute name="rows" type="Number"/>
    <xs:attribute name="cols" type="Number"/>
    <xs:attribute name="disabled">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="disabled"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="readonly">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="readonly"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="tabindex" type="Number"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:complexType name="textarea.type" mixed="true">
    <xs:attributeGroup ref="textarea.attlist"/>
  </xs:complexType>

  <xs:element name="textarea" type="textarea.type"/>

  <!-- fieldset: Form Control Group  -->
  <xs:attributeGroup name="fieldset.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="fieldset.type" mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="legend"/>
      <xs:group ref="Flow.mix"/>
    </xs:choice>
    <xs:attributeGroup ref="fieldset.attlist"/>
  </xs:complexType>
```

```
<xs:element name="fieldset" type="fieldset.type"/>

<!-- legend: Fieldset Legend  -->
<xs:attributeGroup name="legend.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="accesskey" type="Character"/>
</xs:attributeGroup>

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

<xs:element name="legend" type="legend.type"/>

<!-- button: Push Button  -->
<xs:attributeGroup name="button.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="name" type="Text"/>
  <xs:attribute name="value" type="Text"/>
  <!-- !!! xs:attribute name="type" type="Number"/ 07302002 DBA-->
  <xs:attribute name="type" use="optional" default="submit">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="button"/>
        <xs:enumeration value="submit"/>
        <xs:enumeration value="reset"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="tabindex" type="Number"/>
  <xs:attribute name="accesskey" type="Character"/>
</xs:attributeGroup>

<xs:group name="button.content">
  <xs:choice>
    <xs:group ref="BlkNoForm.mix"/>
    <xs:group ref="InlStruct.class"/>
    <xs:group ref="InlPhras.class"/>
    <xs:group ref="InlPres.class"/>
    <xs:group ref="BlkPres.class"/>
    <xs:group ref="I18n.class"/>
    <xs:group ref="InlSpecial.class"/>
    <xs:group ref="Inline.extra"/>
  </xs:choice>
</xs:group>

<xs:complexType name="button.type" mixed="true">
  <xs:group ref="button.content" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xs:attributeGroup ref="button.attlist"/>
  </xs:complexType>

  <xs:element name="button" type="button.type"/>

</xs:schema>
```

D.4.4. Tables

D.4.4.1. Basic Tables

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Basic Tables module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
     $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

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

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

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

<xs:attributeGroup name="table.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="summary" type="Text"/>
</xs:attributeGroup>

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

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

<xs:attributeGroup name="caption.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="summary" type="Text"/>
</xs:attributeGroup>

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

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

<xs:attributeGroup name="tr.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
```

```
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

  <xs:group name="tr.content">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="th"/>
      <xs:element ref="td"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="tr.type">
    <xs:group ref="tr.content"/>
    <xs:attributeGroup ref="tr.attlist"/>
  </xs:complexType>

  <xs:element name="tr" type="tr.type"/>

  <xs:attributeGroup name="th.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="abbr" type="Text"/>
    <xs:attribute name="axis" type="Text"/>
    <xs:attribute name="headers" type="xs:IDREFS"/>
    <xs:attributeGroup ref="scope.attrib"/>
    <xs:attribute name="rowspan" type="Number" use="optional" default="1"/>
    <xs:attribute name="colspan" type="Number" use="optional" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="th.type" mixed="true">
    <xs:group ref="FlowNoTable.mix"/>
    <xs:attributeGroup ref="th.attlist"/>
  </xs:complexType>

  <xs:element name="th" type="th.type"/>

  <xs:attributeGroup name="td.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="abbr" type="Text"/>
    <xs:attribute name="axis" type="Text"/>
    <xs:attribute name="headers" type="xs:IDREFS"/>
    <xs:attributeGroup ref="scope.attrib"/>
    <xs:attribute name="rowspan" type="Number" use="optional" default="1"/>
    <xs:attribute name="colspan" type="Number" use="optional" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="td.type" mixed="true">
    <xs:group ref="FlowNoTable.mix"/>
    <xs:attributeGroup ref="td.attlist"/>
  </xs:complexType>

  <xs:element name="td" type="td.type"/>

</xs:schema>
```

D.4.4.2. Tables

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Tables module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="frame.attrib">
    <xs:attribute name="frame">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="void"/>
          <xs:enumeration value="above"/>
          <xs:enumeration value="below"/>
          <xs:enumeration value="hsides"/>
          <xs:enumeration value="lhs"/>
          <xs:enumeration value="rhs"/>
          <xs:enumeration value="vsides"/>
          <xs:enumeration value="box"/>
          <xs:enumeration value="border"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="rules.attrib">
    <xs:attribute name="rules">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="none"/>
          <xs:enumeration value="groups"/>
```

```
        <xs:enumeration value="rows"/>
        <xs:enumeration value="cols"/>
        <xs:enumeration value="all"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

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

<xs:attributeGroup name="CellHAlign.attrib">
  <xs:attribute name="align">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="left"/>
        <xs:enumeration value="center"/>
        <xs:enumeration value="right"/>
        <xs:enumeration value="justify"/>
        <xs:enumeration value="char"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="char" type="Character"/>
  <xs:attribute name="charoff" type="Length"/>
</xs:attributeGroup>

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

<xs:attributeGroup name="td.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="abbr" type="Text"/>
  <xs:attribute name="axis" type="Text"/>
  <xs:attribute name="headers" type="xs:IDREFS"/>
  <xs:attributeGroup ref="scope.attrib"/>
  <xs:attribute name="rowspan" type="Number" use="optional" default="1"/>
```

```
    <xs:attribute name="colspan" type="Number" use="optional" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="td" type="td.type"/>

  <xs:attributeGroup name="th.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="abbr" type="Text"/>
    <xs:attribute name="axis" type="Text"/>
    <xs:attribute name="headers" type="xs:IDREFS"/>
    <xs:attributeGroup ref="scope.attrib"/>
    <xs:attribute name="rowspan" type="Number" use="optional" default="1"/>
    <xs:attribute name="colspan" type="Number" use="optional" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="th" type="th.type"/>

  <xs:attributeGroup name="tr.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

  <xs:group name="tr.content">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="th"/>
      <xs:element ref="td"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="tr.type">
    <xs:group ref="tr.content"/>
    <xs:attributeGroup ref="tr.attlist"/>
  </xs:complexType>

  <xs:element name="tr" type="tr.type"/>

  <xs:attributeGroup name="col.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="span" type="Number" use="optional" default="1"/>
    <xs:attribute name="width" type="MultiLength"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
```

```
</xs:attributeGroup>

<xs:complexType name="col.type">
  <xs:attributeGroup ref="col.attlist"/>
</xs:complexType>

<xs:element name="col" type="col.type"/>

<xs:attributeGroup name="colgroup.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="span" type="Number" use="optional" default="1"/>
  <xs:attribute name="width" type="MultiLength"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
  <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="colgroup.content">
  <xs:sequence>
    <xs:element ref="col" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="colgroup.type">
  <xs:group ref="colgroup.content"/>
  <xs:attributeGroup ref="colgroup.attlist"/>
</xs:complexType>

<xs:element name="colgroup" type="colgroup.type"/>

<xs:attributeGroup name="tbody.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
  <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="tbody.content">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="tr"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="tbody.type">
  <xs:group ref="tbody.content"/>
  <xs:attributeGroup ref="tbody.attlist"/>
</xs:complexType>

<xs:element name="tbody" type="tbody.type"/>

<xs:attributeGroup name="tfoot.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
  <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="tfoot.content">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="tr"/>
```

```
    </xs:sequence>
  </xs:group>

  <xs:complexType name="tfoot.type">
    <xs:group ref="tfoot.content"/>
    <xs:attributeGroup ref="tfoot.attlist"/>
  </xs:complexType>

  <xs:element name="tfoot" type="tfoot.type"/>

  <xs:attributeGroup name="thead.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
  </xs:attributeGroup>

  <xs:group name="thead.content">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tr"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="thead.type">
    <xs:group ref="thead.content"/>
    <xs:attributeGroup ref="thead.attlist"/>
  </xs:complexType>

  <xs:element name="thead" type="thead.type"/>

  <xs:attributeGroup name="caption.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:group name="caption.content">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tr"/>
    </xs:sequence>
  </xs:group>

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

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

  <xs:attributeGroup name="table.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="summary" type="Text"/>
    <xs:attribute name="width" type="Length"/>
    <xs:attribute name="border" type="Pixels"/>
    <xs:attributeGroup ref="frame.attrib"/>
    <xs:attributeGroup ref="rules.attrib"/>
    <xs:attribute name="cellspacing" type="Length"/>
    <xs:attribute name="cellpadding" type="Length"/>
  </xs:attributeGroup>
```

```
  <xs:group name="table.content">
    <xs:sequence>
      <xs:element ref="caption" minOccurs="0"/>
      <xs:choice>
        <xs:element ref="col" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="colgroup" minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
      <xs:choice>
        <xs:sequence>
          <xs:element ref="thead" minOccurs="0"/>
          <xs:element ref="tfoot" minOccurs="0"/>
          <xs:element ref="tbody" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:choice>
          <xs:element ref="tr" maxOccurs="unbounded"/>
        </xs:choice>
      </xs:choice>
    </xs:sequence>
  </xs:group>

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

  <xs:element name="table" type="table.type"/>

</xs:schema>
```

D.4.5. Image

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Images module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
     <xs:documentation>
       Images

         * img
```

```
        This module provides markup to support basic image embedding.
      </xs:documentation>
    </xs: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.
  -->
    <xs:attributeGroup name="img.attlist">
      <xs:attributeGroup ref="Common.attrib"/>
      <xs:attribute name="src" type="URI" use="required"/>
      <xs:attribute name="alt" type="Text" use="required"/>
      <xs:attribute name="longdesc" type="URI"/>
      <xs:attribute name="height" type="Length"/>
      <xs:attribute name="width" type="Length"/>
    </xs:attributeGroup>

    <xs:complexType name="img.type">
      <xs:attributeGroup ref="img.attlist"/>
    </xs:complexType>

    <xs:element name="img" type="img.type"/>

  </xs:schema>
```

D.4.6. Client-side Image Map

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <!--
This file included to incorporate events used in client side image maps 062302 DBA
-->
  <!--xs:include schemaLocation="xhtml-events-1.xsd"/-->
  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Client-side Image Maps module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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
  </xs:documentation>
</xs:annotation>

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

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

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

<xs:redefine schemaLocation="xhtml-object-1.xsd">
  <xs:attributeGroup name="object.attlist">
    <xs:attributeGroup ref="object.attlist"/>
    <xs:attributeGroup ref="object.csim.attlist"/>
  </xs:attributeGroup>
</xs:redefine>

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

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

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

<!-- modify object attribute definition list -->
<xs:attributeGroup name="object.csim.attlist">
```

```
    <xs:attribute name="usemap" type="xs:IDREF"/>
  </xs:attributeGroup>

  <!-- area -->
  <!-- !!!xs:attribute name="shape" use="required" default="rect" 7302002 DBA -->
  <xs:attributeGroup name="shape.attrib">
  <xs:attribute name="shape" default="rect">
  <!--xs:attribute name="shape" use="required"-->
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="rect"/>
        <xs:enumeration value="circle"/>
        <xs:enumeration value="poly"/>
        <xs:enumeration value="default"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  </xs:attributeGroup>

  <xs:attributeGroup name="area.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="href" type="URI"/>
    <xs:attributeGroup ref="shape.attrib"/>
    <xs:attribute name="coords" type="Text"/>
    <xs:attribute name="nohref" type="Text"/>
    <xs:attribute name="alt" type="Text"/>
    <xs:attribute name="tabindex" type="Number"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:complexType name="area.type">
    <xs:attributeGroup ref="area.attlist"/>
  </xs:complexType>

  <xs:element name="area" type="area.type"/>

  <!-- map -->
  <xs:attributeGroup name="map.attlist">
    <xs:attributeGroup ref="id"/>
    <xs:attributeGroup ref="class"/>
    <xs:attributeGroup ref="title"/>
    <xs:attributeGroup ref="Core.extra.attrib"/>
    <xs:attributeGroup ref="I18n.attrib"/>
    <!--xs:attributeGroup ref="Events.attrib"/-->
  </xs:attributeGroup>

  <xs:group name="map.content">
    <xs:choice maxOccurs="unbounded">
      <xs:group ref="Block.mix"/>
      <xs:element ref="area" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="map.type">
    <xs:group ref="map.content"/>
    <xs:attributeGroup ref="area.attlist"/>
```

```
  </xs:complexType>
  <xs:element name="map" type="map.type"/>

</xs:schema>
```

D.4.7. Server-side Image Map

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Server-side Image Maps module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

<!--
 Redefines only
-->
  <xs:annotation>
    <xs:documentation>
      Server-side Image Maps

      This adds the 'ismap' attribute to the img element to
      support server-side processing of a user selection.
    </xs:documentation>
  </xs:annotation>

  <xs:redefine schemaLocation="xhtml-image-1.xsd">
    <xs:attributeGroup name="img.attlist">
      <xs:attributeGroup ref="img.attlist"/>
      <xs:attributeGroup ref="img.ssimap.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:attributeGroup name="img.ssimap.attlist">
    <xs:attribute name="ismap">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="ismap"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

</xs:schema>
```

D.4.8. Applet

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema module for Java Applets in XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

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

  <xs:group name="applet.content">
    <xs:choice>
      <xs:element ref="param"/>
      <xs:group ref="Flow.mix"/>
    </xs:choice>
  </xs:group>
```

```
  <xs:complexType name="applet.type">
    <xs:group ref="applet.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="applet.attlist"/>
  </xs:complexType>

  <xs:element name="applet" type="applet.type"/>

</xs:schema>
```

D.4.9. Object

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Embedded Object module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      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.

      Elements defined here:

        * object
    </xs:documentation>
  </xs:annotation>

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

```
    <xs:attribute name="height" type="Length"/>
    <xs:attribute name="width" type="Length"/>
    <xs:attribute name="name" type="Text"/>
    <xs:attribute name="tabindex" type="Number"/>
  </xs:attributeGroup>

  <xs:group name="object.content">
    <xs:choice>
      <xs:element ref="param"/>
      <xs:group ref="Flow.mix"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="object.type" mixed="true">
    <xs:group ref="object.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="object.attlist"/>
  </xs:complexType>

  <xs:element name="object" type="object.type"/>

</xs:schema>
```

D.4.10. Frames

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Frames module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Frames

        * frameset, frame, noframes

      This module declares frame-related element types and attributes.
    </xs:documentation>
  </xs:annotation>

<!--
The content model for XHTML documents depends on whether
the <head> is followed by a <frameset> or <body> element.
```

```
-->
  <xs:redefine schemaLocation="../../req/core/xhtml-struct-1.xsd">
    <xs:group name="html.content">
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="head" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="frameset"/>
      </xs:choice>
    </xs:group>
  </xs:redefine>

  <xs:attributeGroup name="frameset.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attribute name="rows" type="MultiLengths"/>
    <xs:attribute name="cols" type="MultiLengths"/>
  </xs:attributeGroup>

  <xs:group name="frameset.content">
    <xs:sequence>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="frameset"/>
        <xs:element ref="frame"/>
      </xs:choice>
      <xs:element ref="noframes" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:group>

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

  <xs:element name="frameset" type="frameset.type"/>

<!--
    reserved frame names start with "_"
    otherwise starts with letter
-->
  <xs:attributeGroup name="frame.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attribute name="longdesc" type="URI"/>
    <xs:attribute name="src" type="URI"/>
    <xs:attribute name="frameborder" use="optional" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:nonNegativeInteger">
          <xs:enumeration value="1"/>
          <xs:enumeration value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="marginwidth" type="Pixels"/>
    <xs:attribute name="marginheight" type="Pixels"/>
    <xs:attribute name="noresize" type="Text"/>
    <xs:attribute name="scrolling" use="optional" default="auto">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="yes"/>
          <xs:enumeration value="no"/>
```

```
          <xs:enumeration value="auto"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:complexType name="frame.type">
    <xs:attributeGroup ref="frame.attlist"/>
  </xs:complexType>

  <xs:element name="frame" type="frame.type"/>

<!--
noframes
-->
  <xs:group name="noframes.content">
    <xs:sequence>
      <xs:element ref="body" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:group>

  <xs:attributeGroup name="noframes.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="noframes" type="noframes.type"/>

</xs:schema>
```

D.4.11. Target

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Target module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
```

```
    <xs:documentation>
      Target

        * target

      This module declares the 'target' attribute used for opening windows
    </xs:documentation>
  </xs:annotation>

  <xs:redefine schemaLocation="../../req/core/xhtml-hypertext-1.xsd">
    <xs:attributeGroup name="a.attlist">
        <xs:attributeGroup ref="a.attlist" />
        <xs:attributeGroup ref="a.target.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:redefine schemaLocation="../xhtml-csismap-1.xsd">
    <xs:attributeGroup name="area.attlist">
      <xs:attributeGroup ref="area.attlist" />
        <xs:attributeGroup ref="area.target.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:redefine schemaLocation="../xhtml-link-1.xsd">
    <xs:attributeGroup name="link.attlist">
      <xs:attributeGroup ref="link.attlist" />
        <xs:attributeGroup ref="link.target.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:redefine schemaLocation="../xhtml-form-1.xsd">
    <xs:attributeGroup name="form.attlist">
      <xs:attributeGroup ref="form.attlist" />
        <xs:attributeGroup ref="form.target.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:redefine schemaLocation="../xhtml-base-1.xsd">
    <xs:attributeGroup name="base.attlist">
      <xs:attributeGroup ref="base.attlist" />
        <xs:attributeGroup ref="base.target.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:attributeGroup name="base.target.attlist">
    <xs:attribute name="target" type="FrameTarget"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="form.target.attlist">
    <xs:attribute name="target" type="FrameTarget"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="link.target.attlist">
    <xs:attribute name="target" type="FrameTarget"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="area.target.attlist">
```

```
    <xs:attribute name="target" type="FrameTarget"/>
  </xs:attributeGroup>

  <xs:simpleType name="FrameTarget">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

  <xs:attributeGroup name="a.target.attlist">
    <xs:attribute name="target" type="FrameTarget"/>
  </xs:attributeGroup>

</xs:schema>
```

D.4.12. Iframe

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Frame Element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Inline Frames

        * iframe

      This module declares the iframe element type and its attributes,
      used to create an inline frame within a document.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="iframe.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attribute name="longdesc" type="URI"/>
    <xs:attribute name="src" type="URI"/>
    <xs:attribute name="frameborder" use="optional" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:nonNegativeInteger">
          <xs:enumeration value="1"/>
          <xs:enumeration value="0"/>
        </xs:restriction>
      </xs:simpleType>
```

```
    </xs:attribute>
    <xs:attribute name="marginwidth" type="Pixels"/>
    <xs:attribute name="marginheight" type="Pixels"/>
    <xs:attribute name="scrolling" use="optional" default="auto">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="yes"/>
          <xs:enumeration value="no"/>
          <xs:enumeration value="auto"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="height" type="Length"/>
    <xs:attribute name="width" type="Length"/>
  </xs:attributeGroup>

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

  <xs:element name="iframe" type="iframe.type"/>

</xs:schema>
```

D.4.13. Intrinsic Events

```
Module SCHEMA/req/framework/xhtml-events-1.xsd not found!
```

D.4.14. Metainformation

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Metainformation module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Meta Information

        * meta
```

```
      This module declares the meta element type and its attributes,
      used to provide declarative document metainformation.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="meta.attlist">
    <xs:attributeGroup ref="I18n.attrib"/>
    <xs:attribute name="http-equiv" type="xs:NMTOKEN"/>
    <xs:attribute name="name" type="xs:NMTOKEN"/>
    <xs:attribute name="content" type="Text" use="required"/>
    <xs:attribute name="scheme" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="meta.type">
    <xs:attributeGroup ref="meta.attlist"/>
  </xs:complexType>

  <xs:element name="meta" type="meta.type"/>

</xs:schema>
```

D.4.15. Scripting

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Scripting module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="../req/framework/xml-attribs-1.xsd"/>
```

```
  <xs:attributeGroup name="script.attlist">
    <xs:attribute name="charset" type="Charset"/>
    <xs:attribute name="type" type="ContentType"/>
    <xs:attribute name="src" type="URI"/>
    <xs:attribute name="defer" type="Text"/>
    <xs:attribute ref="xml:space" default="preserve"/>
  </xs:attributeGroup>

  <xs:complexType name="script.type" mixed="true">
    <xs:attributeGroup ref="script.attlist"/>
  </xs:complexType>

  <xs:element name="script" type="script.type"/>

  <xs:attributeGroup name="noscript.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="noscript.type">
    <xs:group ref="Block.mix"/>
    <xs:attributeGroup ref="noscript.attlist"/>
  </xs:complexType>

  <xs:element name="noscript" type="noscript.type"/>

</xs:schema>
```

D.4.16. Stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Stylesheets module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Stylesheets

        * style

      This module declares the style element type and its attributes,
```

```
      used to embed stylesheet information in the document head element.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="style.attlist">
    <xs:attributeGroup ref="title"/>
    <xs:attributeGroup ref="I18n.attrib"/>
    <xs:attribute name="type" type="ContentType"/>
    <xs:attribute name="media" type="MediaDesc"/>
    <xs:attribute ref="xml:space" default="preserve"/>
  </xs:attributeGroup>

  <xs:complexType name="style.type" mixed="true">
    <xs:attributeGroup ref="style.attlist"/>
  </xs:complexType>

  <xs:element name="style" type="style.type"/>

</xs:schema>
```

D.4.17. Style Attribute

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Style module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Inline Style module

      This module declares the 'style' attribute, used to support inline
      style markup.
    </xs:documentation>
  </xs:annotation>

<!--
Redefines only
-->
  <xs:redefine schemaLocation="../req/framework/xhtml-attribs-1.xsd">
    <xs:attributeGroup name="Common.extra">
      <!--xs:attributeGroup ref="Common.extra"/-->
```

```
      <xs:attributeGroup ref="style.inline.attlist"/>
    </xs:attributeGroup>
  </xs:redefine>

  <xs:attributeGroup name="style.inline.attlist">
    <xs:attribute name="style" type="Text"/>
  </xs:attributeGroup>

</xs:schema>
```

D.4.18. Link

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Link Element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="link.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="charset" type="Charset"/>
    <xs:attribute name="href" type="URI"/>
    <xs:attribute name="hreflang" type="LanguageCode"/>
    <xs:attribute name="type" type="ContentType"/>
    <xs:attribute name="rel" type="LinkTypes"/>
    <xs:attribute name="rev" type="LinkTypes"/>
    <xs:attribute name="media" type="MediaDesc"/>
  </xs:attributeGroup>

  <xs:complexType name="link.type">
    <xs:attributeGroup ref="link.attlist"/>
```

```
  </xs:complexType>

  <xs:element name="link" type="link.type"/>

</xs:schema>
```

D.4.19. Base

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Base Element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="base.attlist">
    <xs:attribute name="href" type="URI" use="required" />
  </xs:attributeGroup>

  <xs:complexType name="base.type">
    <xs:attributeGroup ref="base.attlist"/>
  </xs:complexType>

  <xs:element name="base" type="base.type"/>

  <xs:redefine schemaLocation="../req/core/xhtml-struct-1.xsd">
    <xs:group name="head.content">
      <xs:sequence>
        <xs:group ref="HeadOpts.mix"/>
```

```
        <xs:choice>
          <xs:sequence>
            <xs:element ref="title"/>
            <xs:group ref="HeadOpts.mix"/>
            <xs:sequence minOccurs="0">
              <xs:element ref="base"/>
              <xs:group ref="HeadOpts.mix"/>
            </xs:sequence>
          </xs:sequence>
          <xs:sequence>
            <xs:element ref="base"/>
            <xs:group ref="HeadOpts.mix"/>
            <xs:element ref="title"/>
            <xs:group ref="HeadOpts.mix"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
    </xs:group>
  </xs:redefine>

</xs:schema>
```

D.4.20. Name Identification

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Name Identifier module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
```

```
  </xs:annotation>

<!--
!!! All redefines
-->
  <xs:attributeGroup name="form.name.attlist">
    <xs:attribute name="name" type="Text"/>
  </xs:attributeGroup>

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

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

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

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

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

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

</xs:schema>
```

D.4.21. Legacy

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Legacy Markup module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>
```

```
  <xs:annotation>
    <xs:documentation>
      XHTML Legacy Markup

      this module includes modules for both frames and
      additional deprecated elements and attributes.
    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="xhtml-framedefs-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Framedefs module
        Elements defined here:

          * frameset, frame, noframes, att:target, iframe
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="misc/xhtml-misc-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Miscellaneous module
        Attributes defined here:

          * font, basefont, center, s, strike, u, dir, menu, isindex
            (plus additional datatypes and attributes)

      </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

D.4.22. Ruby

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the Ruby module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>
```

```
  <xs:annotation>
    <xs:documentation>
      "Ruby" are short runs of text alongside the base text, typically
      used in East Asian documents to indicate pronunciation or to
      provide a short annotation. The full specification for Ruby is here:

        http://www.w3.org/TR/ruby

      This module defines "Ruby " or "complex Ruby" as described
      in the specification:

        http://www.w3.org/TR/ruby/#complex

      Simple or Basic Ruby are defined in a separate module.
    </xs:documentation>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Ruby Elements

        * ruby, rbc, rtc, rb, rt, rp

      This module declares the elements and their attributes used to
      support complex ruby annotation markup.
    </xs:documentation>
  </xs:annotation>

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

  <xs:group name="Ruby.content.group">
    <xs:sequence>
      <xs:element ref="rbc"/>
      <xs:element ref="rtc" minOccurs="1" maxOccurs="2"/>
    </xs:sequence>
  </xs:group>

<!--
add to this group any common attributes for all Ruby elements
-->
  <xs:attributeGroup name="Ruby.common.attrib">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="Ruby.common.attlist">
```

```
    <xs:attributeGroup ref="Ruby.common.attrib"/>
  </xs:attributeGroup>

  <xs:group name="Ruby.content">
    <xs:choice>
      <xs:group ref="Ruby.content.simple"/>
      <xs:group ref="Ruby.content.group"/>
    </xs:choice>
  </xs:group>

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

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

<!--
rbc (ruby base component) element
-->
  <xs:attributeGroup name="Rbc.attrib">
    <xs:attributeGroup ref="Ruby.common.attlist"/>
  </xs:attributeGroup>

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

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

<!--
 rtc (ruby text component) element
-->
  <xs:attributeGroup name="Rtc.attrib">
    <xs:attributeGroup ref="Ruby.common.attlist"/>
  </xs:attributeGroup>

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

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

<!--
rb (ruby base) element
-->
  <xs:attributeGroup name="Rb.attrib">
    <xs:attributeGroup ref="Ruby.common.attlist"/>
  </xs:attributeGroup>

  <xs:complexType name="Rb.type" mixed="true">
```

```
    <xs:attributeGroup ref="Rb.attrib"/>
  </xs:complexType>

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

<!--
 rt (ruby text) element
-->
  <xs:attributeGroup name="Rt.attrib">
    <xs:attributeGroup ref="Ruby.common.attlist"/>
    <!-- rbspan attribute is used for complex ruby only .. -->
    <xs:attribute name="rbspan"  type="xs:positiveInteger"
      default="1" use="optional" />
  </xs:attributeGroup>

  <xs:complexType name="Rt.type" mixed="true">
    <xs:attributeGroup ref="Rt.attrib"/>
  </xs:complexType>

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

<!-- rp (ruby parenthesis) element  -->
  <xs:attributeGroup name="Rp.attrib">
    <xs:attributeGroup ref="Ruby.common.attlist"/>
  </xs:attributeGroup>

  <xs:complexType name="Rp.type" mixed="true">
    <xs:attributeGroup ref="Rp.attrib"/>
  </xs:complexType>

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

</xs: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"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           xmlns:xml="http://www.w3.org/XML/1998/namespace"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
```

```
    This is the XML Schema block phrasal element module for XHTML

    @author: Daniel Austin austin.d@ic.grainger.com
    $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
</xs:documentation>
<xs:documentation source="../../../xhtml-copyright-1.xsd"/>
</xs:annotation>

<xs:annotation>
  <xs:documentation>
    Block Phrasal elements module

      * address, blockquote, pre, h1, h2, h3, h4, h5, h6

    This module declares the elements and their attributes used to
    support block-level phrasal markup.
  </xs:documentation>
</xs:annotation>

<!-- address -->
<xs:attributeGroup name="address.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

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

<xs:element name="address" type="address.type"/>

<!-- blockquote -->
<xs:attributeGroup name="blockquote.attlist">
  <xs:attributeGroup ref="Common.attrib"/>
  <xs:attribute name="cite" type="URI"/>
</xs:attributeGroup>

<xs:complexType name="blockquote.type">
  <xs:group ref="Block.mix"/>
  <xs:attributeGroup ref="blockquote.attlist"/>
</xs:complexType>

<xs:element name="blockquote" type="blockquote.type"/>

<!-- pre -->
<xs:group name="pre.content">
  <xs:choice>
    <xs:group ref="InlStruct.class"/>
    <xs:group ref="InlPhras.class"/>
    <xs:element ref="tt"/>
    <xs:element ref="i"/>
    <xs:element ref="b"/>
    <xs:group ref="I18n.class"/>
    <xs:group ref="Anchor.class"/>
    <xs:element ref="script"/>
    <xs:element ref="map"/>
    <xs:group ref="Inline.extra"/>
```

```
    </xs:choice>
  </xs:group>

  <xs:attributeGroup name="pre.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute ref="xml:space" default="preserve"/>
  </xs:attributeGroup>

  <xs:complexType name="pre.type" mixed="true">
    <xs:group ref="pre.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="pre.attlist"/>
  </xs:complexType>

  <xs:element name="pre" type="pre.type"/>

  <!-- Heading Elements  -->
  <xs:attributeGroup name="heading.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

<!--  !!! define a sub group for these -->
  <xs:element name="h1" type="heading.type"/>
  <xs:element name="h2" type="heading.type"/>
  <xs:element name="h3" type="heading.type"/>
  <xs:element name="h4" type="heading.type"/>
  <xs:element name="h5" type="heading.type"/>
  <xs:element name="h6" type="heading.type"/>

</xs:schema>
```

D.5.2. Block Presentational

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML SchemaBlock presentation element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
  </xs:annotation>

  <xs:annotation>
```

```
    <xs:documentation>
      Block Presentational Elements

        * hr

      This module declares the elements and their attributes used to
      support block-level presentational markup.
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:attributeGroup name="hr.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="hr.type">
    <xs:attributeGroup ref="hr.attlist"/>
  </xs:complexType>

  <xs:element name="hr" type="hr.type"/>

</xs:schema>
```

D.5.3. Block Structural

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Block Structural module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Block Structural

        * div, p

      This module declares the elements and their attributes used to
      support block-level structural markup.
     </xs:documentation>
  </xs:annotation>

  <!-- div -->
```

```
  <xs:attributeGroup name="div.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="div" type="div.type"/>

  <!-- p -->
  <xs:attributeGroup name="p.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="p" type="p.type"/>

</xs:schema>
```

D.5.4. Inline Phrasal

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Phrasal module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
  </xs:documentation>
  </xs:annotation>
```

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

  <xs:attributeGroup name="InlPhras.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="abbr" type="InlPhras.type"/>

  <xs:element name="acronym" type="InlPhras.type"/>

  <xs:element name="cite" type="InlPhras.type"/>

  <xs:element name="code" type="InlPhras.type"/>

  <xs:element name="dfn" type="InlPhras.type"/>

  <xs:element name="em" type="InlPhras.type"/>

  <xs:element name="kbd" type="InlPhras.type"/>

  <xs:element name="samp" type="InlPhras.type"/>

  <xs:element name="strong" type="InlPhras.type"/>

  <xs:element name="var" type="InlPhras.type"/>

  <xs:attributeGroup name="q.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="cite" type="URI"/>
  </xs:attributeGroup>

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

  <xs:element name="q" type="q.type"/>

</xs:schema>
```

D.5.5. Inline Presentational

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Presentation element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="InlPres.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="b" type="InlPres.type"/>

  <xs:element name="big" type="InlPres.type"/>

  <xs:element name="i" type="InlPres.type"/>

  <xs:element name="small" type="InlPres.type"/>

  <xs:element name="sub" type="InlPres.type"/>

  <xs:element name="sup" type="InlPres.type"/>

  <xs:element name="tt" type="InlPres.type"/>

</xs:schema>
```

D.5.6. Inline Structural

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Structural element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Inline Structural

        * br, span

      This module declares the elements and their attributes
      used to support inline-level structural markup.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="br.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="br.type">
    <xs:attributeGroup ref="br.attlist"/>
  </xs:complexType>

  <xs:element name="br" type="br.type"/>

  <xs:attributeGroup name="span.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="span" type="span.type"/>

</xs:schema>
```

D.5.7. Param

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Param Element module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
      </xs:documentation>
      <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="param.attlist">
    <xs:attributeGroup ref="id"/>
    <xs:attribute name="name" type="Text"/>
    <xs:attribute name="value" type="Text"/>
    <!-- xs:attribute name="valuetype" use="required" default="data" 07302002 DBA -->
    <xs:attribute name="valuetype" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="data"/>
          <xs:enumeration value="ref"/>
          <xs:enumeration value="object"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="type" type="ContentType"/>
  </xs:attributeGroup>

  <xs:complexType name="param.type" mixed="true">
    <xs:attributeGroup ref="param.attlist"/>
  </xs:complexType>

  <xs:element name="param" type="param.type"/>

</xs:schema>
```

D.5.8. Miscellaneous Legacy

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Miscellaneous Legacy
      Markup module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      XHTML 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 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.
    </xs:documentation>
  </xs:annotation>

  <xs:attributeGroup name="font.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attributeGroup ref="I18n.attrib"/>
    <xs:attribute name="size" type="Text"/>
    <xs:attribute name="color" type="Color"/>
    <xs:attribute name="face" type="Text"/>
  </xs:attributeGroup>

  <xs:group name="font.content">
    <xs:choice>
      <xs:group ref="Inline.mix"/>
    </xs:choice>
  </xs:group>

  <xs:complexType name="font.type" mixed="true">
```

```
    <xs:group ref="font.content" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attributeGroup ref="font.attlist"/>
  </xs:complexType>

  <xs:element name="font" type="font.type"/>

  <xs:attributeGroup name="basefont.attlist">
    <xs:attributeGroup ref="id"/>
    <xs:attribute name="size" type="Text"/>
    <xs:attribute name="color" type="Color"/>
    <xs:attribute name="face" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="basefont.type">
    <xs:attributeGroup ref="basefont.attlist"/>
  </xs:complexType>

  <xs:element name="basefont" type="basefont.type"/>

  <xs:attributeGroup name="center.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="center" type="center.type"/>

  <xs:attributeGroup name="s.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="s" type="s.type"/>

  <xs:attributeGroup name="strike.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

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

  <xs:element name="strike" type="strike.type"/>

  <xs:attributeGroup name="u.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
  </xs:attributeGroup>

  <xs:complexType name="u.type" mixed="true">
```

```
    <xs:group ref="Inline.mix"/>
    <xs:attributeGroup ref="u.attlist"/>
  </xs:complexType>

  <xs:element name="u" type="u.type"/>

  <xs:attributeGroup name="dir.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="compact" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="dir.type" mixed="true">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="li"/>
    </xs:sequence>
    <xs:attributeGroup ref="dir.attlist"/>
  </xs:complexType>

  <xs:element name="dir" type="dir.type"/>

  <xs:attributeGroup name="menu.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="compact" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="menu.type" mixed="true">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="li"/>
    </xs:sequence>
    <xs:attributeGroup ref="menu.attlist"/>
  </xs:complexType>

  <xs:element name="menu" type="menu.type"/>

  <xs:attributeGroup name="isindex.attlist">
    <xs:attributeGroup ref="Core.attrib"/>
    <xs:attributeGroup ref="I18n.attrib"/>
    <xs:attribute name="prompt" type="Text"/>
  </xs:attributeGroup>

  <xs:complexType name="isindex.type">
    <xs:attributeGroup ref="isindex.attlist"/>
  </xs:complexType>

  <xs:element name="isindex" type="isindex.type"/>

  <xs:annotation>
    <xs:documentation>
      Attribute redefinitions
    </xs:documentation>
  </xs:annotation>

  <!-- !!! Redefines -->
  <xs:attributeGroup name="align.legacy.attlist">
    <xs:attribute name="align">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
```

```
          <xs:enumeration value="left"/>
          <xs:enumeration value="center"/>
          <xs:enumeration value="right"/>
          <xs:enumeration value="justify"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

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

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

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

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

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

  <xs:attributeGroup name="h1.legacy.attlist">
```

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

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

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

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

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

  <xs:attributeGroup name="hr.legacy.attlist">
    <xs:attribute name="align">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="left"/>
          <xs:enumeration value="center"/>
          <xs:enumeration value="right"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="noshade" use="optional" default="none"/>
    <xs:attribute name="size" type="Pixels"/>
    <xs:attribute name="width" type="Length"/>
  </xs:attributeGroup>

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

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

  <xs:attributeGroup name="legend.legacy.attlist">
    <xs:attribute name="align">
```

```
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="top"/>
        <xs:enumeration value="bottom"/>
        <xs:enumeration value="left"/>
        <xs:enumeration value="right"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

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

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

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

<xs:attributeGroup name="p.legacy.attlist">
  <xs:attributeGroup ref="align.legacy.attlist"/>
</xs:attributeGroup>

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

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

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

<xs:attributeGroup name="tr.legacy.attlist">
```

```
    <xs:attribute name="bgcolor" type="Color"/>
  </xs:attributeGroup>

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

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

  <xs:attributeGroup name="ul.legacy.attlist">
    <xs:attribute name="type" type="Text"/>
    <xs:attribute name="compact" use="optional" />
  </xs:attributeGroup>

</xs:schema>
```

D.5.9. Legacy Frames

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
     <xs:documentation>
       This is the XML Schema Frames Definition module for XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Frames Definitions

        * frameset, frame, noframes, att:target, iframe

      This module includes 3 additional modules, for frames,
      the target attribute, and iframe.
    </xs:documentation>
  </xs:annotation>
```

```
  <xs:include schemaLocation="frames/xhtml-frames-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Frames module

        Elements defined here:

          * frameset, frame, noframes
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="frames/xhtml-target-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Target module

        Attributes defined here:

          * target
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="frames/xhtml-iframe-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Iframe module

        Elements defined here:

          * iframe
      </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

D.5.10. Optional Module Hub

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XHTML optional elements module for XHTML.
      Please use this namespace for XHTML elements:

        http://www.w3.org/1999/xhtml
```

```
      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Optional Element modules

      Comment out those you don't need

          +  Edit
          +  Bdo
          +  Presentational
          +  Link
          +  Meta
          +  Base
          +  Scripting
          +  Style
          +  Image
          +  Client side image maps
          +  Server side image maps
          +  Param
          +  Applet
          +  Object
          +  Tables
          +  Forms
          +  Nameident
          +  Legacy
          +  Ruby
          +  Basic forms
          +  Basic tables
          +  Basic Ruby

    </xs:documentation>
  </xs:annotation>

  <xs:include schemaLocation="opt/xhtml-edit-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Edit module

        Elements defined here:
          * ins, del
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-bdo-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Bidirectional element module

        Elements defined here:
          * bdo
      </xs:documentation>
```

```
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-pres-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Presentational module

         Elements defined here:
           * hr, b, big, i, small,sub, sup, tt
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-link-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Link module

        Elements defined here:
          * link
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-meta-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Meta module

        Elements defined here:
        * meta
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-base-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Base module

        Elements defined here:
          * base
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-script-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Scripting module

        Elements defined here:
          * script, noscript
      </xs:documentation>
    </xs:annotation>
  </xs:include>
```

```
<xs:include schemaLocation="opt/xhtml-style-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Style module

      Elements defined here:
        * style
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="opt/xhtml-inlstyle-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Style attribute module

      Attribute defined here:
        * style
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="opt/xhtml-image-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Image module

      Elements defined here:
        * img
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="opt/xhtml-csismap-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Client-side mage maps module

      Elements defined here:
        * area, map
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="opt/xhtml-ssismap-1.xsd">
  <xs:annotation>
    <xs:documentation>
     Server-side image maps module

      Attributes defined here:
        * ismap on img
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="opt/xhtml-param-1.xsd">
```

```
    <xs:annotation>
      <xs:documentation>
        Param module

        Elements defined here:
          * param
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-applet-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Applet module

        Elements defined here:
          * applet
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-object-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Object module

        Elements defined here:
          * object
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-table-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Tables module

        Elements defined here:
          * table, caption, thead, tfoot, tbody, colgroup, col, tr, th, td
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="opt/xhtml-form-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Forms module

        Elements defined here:
          * form, label, input, select, optgroup, option,
          * textarea, fieldset, legend, button
      </xs:documentation>
    </xs:annotation>
  </xs:include>

 <!--
  <xs:include schemaLocation="opt/xhtml-nameident-1.xsd">
```

```
      <xs:annotation>
        <xs:documentation>
          Nameident module

          Attributes defined here:
            * name (for retro browsers)
        </xs:documentation>
      </xs:annotation>
    </xs:include>
  -->

  <!--
    <xs:include schemaLocation="opt/xhtml-legacy-1.xsd">
      <xs:annotation>
        <xs:documentation>
          Legacy module

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

    <xs:include schemaLocation="opt/xhtml-ruby-1.xsd">
      <xs:annotation>
        <xs:documentation>
          Ruby module

          Elements defined here:
            * ruby, rbc, rtc, rb, rt, rp

          Note that either Ruby or Basic Ruby should be used but not both
        </xs:documentation>
      </xs:annotation>
    </xs:include>

  <!--
    <xs:include schemaLocation="opt/xhtml-ruby-basic-1.xsd">
      <xs:annotation>
        <xs:documentation>
          Basic Ruby module

          Elements defined here:
          * ruby, rb, rt, rp


          Ruby Basic omits the rbc and rtc elements, the rbspan attribute,
          and modifies the content model of the ruby element.
        </xs:documentation>
      </xs:annotation>
    </xs:include>
  -->

    <xs:annotation>
```

```
      <xs:documentation>
        Basic modules for forms and tables

        Note that to use these modules you must comment out
        the XHTML Forms and Tables modules
      </xs:documentation>
    </xs:annotation>

  <!--
    <xs:include  schemaLocation="opt/basic/xhtml-basic-form-1.xsd">
      <xs:annotation>
        <xs:docume ntation>
          Basic Forms module

          Note that this module is not used in XHTML 1.1. It is designed
          for use with XHTML Basic

          Elements defined here:
            * form, label, input, select, option, textarea
        </xs:documentation>
      </xs:annotation>
    </xs:include>
  -->

  <!--
    <xs:include schemaLocation="opt/basic/xhtml-basic-table-1.xsd">
      <xs:annotation>
        <xs:documentation>
          Basic Tables module

          Note that this module is not used in XHTML It is designed
          for use with XHTML Basic

          Elements defined here:
            * table, caption, tr, th, td
        </xs:documentation>
      </xs:annotation>
    </xs:include>
  -->

    <!--xs:include schemaLocation="opt/xhtml-events-1.xsd"/-->
  </xs:schema>
```

D.5.11. Core Hub Module

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
```

```
    This is the XML Schema Core Elements module for XHTML

    @author: Daniel Austin austin.d@ic.grainger.com
    $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
  </xs:documentation>
  <xs:documentation source="../xhtml-copyright-1xsd"/>
</xs:annotation>

<xs:annotation>
  <xs:documentation>

    Required core modules for elements

        +  text
        +  hypertext
        +  lists
        +  structure

  This module includes only other modules and is required.
</xs:documentation>
</xs:annotation>

<xs:include schemaLocation="core/xhtml-text-1.xsd">
  <xs:annotation>
    <xs: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
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="core/xhtml-hypertext-1.xsd">
  <xs:annotation>
    <xs:documentation>
     Hypertext module

     Elements defined here:
       * a
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="core/xhtml-list-1.xsd">
  <xs:annotation>
```

```
      <xs:documentation>
        Lists module

        Elements defined here:
          * dt, dd, dl, ol, ul, li
      </xs:documentation>
    </xs:annotation>
  </xs:include>

  <xs:include schemaLocation="core/xhtml-struct-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Structural module

        Elements defined here:
          * title, head, body, html
      </xs:documentation>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

D.5.12. XHTML 1.1 Content Model

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           blockDefault="#all"
           finalDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema module common content models in XHTML

      @author: Daniel Austin austin.d@ic.grainger.com
      $Id: xhtml-m12n-schema.ps,v 1.1 2002/12/09 15:08:27 henri Exp $
    </xs:documentation>
    <xs:documentation source="../../xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs: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).
  </xs:documentation>
  </xs:annotation>

  <xs:group name="HeadOpts.mix">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="script"/>
      <xs:element ref="style"/>
      <xs:element ref="meta"/>
      <xs:element ref="link"/>
      <xs:element ref="object"/>
    </xs:choice>
  </xs:group>

<!--
ins and del are used to denote editing changes
-->
  <xs:group name="Edit.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="ins"/>
      <xs:element ref="del"/>
    </xs:choice>
  </xs:group>

<!--
script and noscript are used to contain scripts
and alternative content
-->
  <xs:group name="Script.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="script"/>
      <xs:element ref="noscript"/>
    </xs:choice>
  </xs:group>

  <xs:group name="Misc.extra">
    <xs:choice/>
  </xs:group>

<!--
These elements are neither block nor inline, and can
essentially be used anywhere in the document body.
-->
  <xs:group name="Misc.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="Edit.class"/>
      <xs:group ref="Script.class"/>
      <xs:group ref="Misc.extra"/>
    </xs:choice>
  </xs:group>

  <!-- Inline Elements -->
  <xs:group name="InlStruct.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br"/>
```

```
      <xs:element ref="span"/>
    </xs:choice>
  </xs:group>

  <xs:group name="InlPhras.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="em"/>
      <xs:element ref="strong"/>
      <xs:element ref="dfn"/>
      <xs:element ref="code"/>
      <xs:element ref="samp"/>
      <xs:element ref="kbd"/>
      <xs:element ref="var"/>
      <xs:element ref="cite"/>
      <xs:element ref="abbr"/>
      <xs:element ref="acronym"/>
      <xs:element ref="q"/>
    </xs:choice>
  </xs:group>

  <xs:group name="InlPres.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="tt"/>
      <xs:element ref="i"/>
      <xs:element ref="b"/>
      <xs:element ref="big"/>
      <xs:element ref="small"/>
      <xs:element ref="sub"/>
      <xs:element ref="sup"/>
    </xs:choice>
  </xs:group>

  <xs:group name="I18n.class">
    <xs:sequence>
      <xs:element ref="bdo"/>
    </xs:sequence>
  </xs:group>

  <xs:group name="Anchor.class">
    <xs:sequence>
      <xs:element ref="a"/>
    </xs:sequence>
  </xs:group>

  <xs:group name="InlSpecial.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="img"/>
      <xs:element ref="map"/>
      <xs:element ref="applet"/>
      <xs:element ref="object"/>
    </xs:choice>
  </xs:group>

  <xs:group name="InlForm.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="input"/>
      <xs:element ref="select"/>
```

```
      <xs:element ref="textarea"/>
      <xs:element ref="label"/>
      <xs:element ref="button"/>
    </xs:choice>
  </xs:group>

  <xs:group name="Inline.extra">
    <xs:choice/>
  </xs:group>

  <xs:group name="Ruby.class">
    <xs:sequence>
      <xs:element ref="ruby"/>
    </xs:sequence>
  </xs:group>

<!--
Inline.class includes all inline elements,
used as a component in mixes
-->
  <xs:group name="Inline.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="InlStruct.class"/>
      <xs:group ref="InlPhras.class"/>
      <xs:group ref="InlPres.class"/>
      <xs:group ref="I18n.class"/>
      <xs:group ref="Anchor.class"/>
      <xs:group ref="InlSpecial.class"/>
      <xs:group ref="InlForm.class"/>
      <xs:group ref="Ruby.class"/>
      <xs:group ref="Inline.extra"/>
    </xs:choice>
  </xs:group>

<!--
InlNoRuby.class includes all inline elements
except ruby, used as a component in mixes
-->
  <xs:group name="InlNoRuby.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="InlPhras.class"/>
      <xs:group ref="InlPres.class"/>
      <xs:group ref="I18n.class"/>
      <xs:group ref="Anchor.class"/>
      <xs:group ref="InlSpecial.class"/>
      <xs:group ref="InlForm.class"/>
      <xs:group ref="Inline.extra"/>
    </xs:choice>
  </xs:group>

<!--
NoRuby.content includes all inlines except ruby
-->
  <xs:complexType name="NoRuby.content" mixed="true">
    <xs:sequence>
      <xs:group ref="InlNoRuby.class"/>
      <xs:group ref="Misc.class"/>
```

```
    </xs:sequence>
  </xs:complexType>

<!--
InlNoAnchor.class includes all non-anchor inlines,
used as a component in mixes
-->
  <xs:group name="InlNoAnchor.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="InlPhras.class"/>
      <xs:group ref="InlPres.class"/>
      <xs:group ref="I18n.class"/>
      <xs:group ref="InlSpecial.class"/>
      <xs:group ref="InlForm.class"/>
      <xs:group ref="Ruby.class"/>
      <xs:group ref="Inline.extra"/>
    </xs:choice>
  </xs:group>

<!--
InlNoAnchor.mix includes all non-anchor inlines
-->
  <xs:group name="InlNoAnchor.mix">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="InlNoAnchor.class"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>

<!--
Inline.mix includes all inline elements, including Misc.class
-->
  <xs:group name="Inline.mix">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="Inline.class"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>

<!--
In the HTML 4 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.
-->
  <xs:group name="Heading.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="h1"/>
      <xs:element ref="h2"/>
      <xs:element ref="h3"/>
      <xs:element ref="h4"/>
      <xs:element ref="h5"/>
      <xs:element ref="h6"/>
    </xs:choice>
  </xs:group>

  <xs:group name="List.class">
```

```
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="ul"/>
      <xs:element ref="ol"/>
      <xs:element ref="dl"/>
    </xs:choice>
  </xs:group>

  <xs:group name="BlkStruct.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="p"/>
      <xs:element ref="div"/>
    </xs:choice>
  </xs:group>

  <xs:group name="BlkPhras.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="pre"/>
      <xs:element ref="blockquote"/>
      <xs:element ref="address"/>
    </xs:choice>
  </xs:group>

  <xs:group name="BlkPres.class">
    <xs:sequence>
      <xs:element ref="hr" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:group name="BlkSpecial.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="table"/>
      <xs:element ref="form"/>
      <xs:element ref="fieldset"/>
    </xs:choice>
  </xs:group>

  <xs:group name="Block.extra">
    <xs:choice minOccurs="0" maxOccurs="unbounded"/>
  </xs:group>

<!--
Block.class includes all block elements,
used as an component in mixes
-->
  <xs:group name="Block.class">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="BlkStruct.class"/>
      <xs:group ref="BlkPhras.class"/>
      <xs:group ref="BlkPres.class"/>
      <xs:group ref="BlkSpecial.class"/>
      <xs:group ref="Block.extra"/>
    </xs:choice>
  </xs:group>

<!--
Block.mix includes all block elements plus %Misc.class;
-->
```

```
  <xs:group name="Block.mix">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="Heading.class"/>
      <xs:group ref="List.class"/>
      <xs:group ref="Block.class"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs: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 structure module.
-->
  <xs:group name="Flow.mix">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="Heading.class"/>
      <xs:group ref="List.class"/>
      <xs:group ref="Block.class"/>
      <xs:group ref="Inline.class"/>
      <xs:group ref="Misc.class"/>
    </xs:choice>
  </xs:group>

</xs:schema>
```