Modularization of XHTML™ in XML Schema

Modularization of XHTML™ in XML Schema

W3C Working Draft 3 October 2003

This version:
    http://www.w3.org/TR/2003/WD-xhtml-m12n-schema-20031003
Latest version:
    http://www.w3.org/TR/xhtml-m12n-schema
Previous version:
    http://www.w3.org/TR/2002/WD-xhtml-m12n-schema-20021209
Diff-marked version:
    xhtml-m12n-schema-diff.html
Editors:
    Daniel Austin, W. W. Grainger, Inc.
    Subramanian Peruvemba, Oracle Corporation
    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 modularization of XHTML using XML Schema. The document also provides a XML Schema implementation of the abstract modules defined in *Modularization of XHTML* [XHTMLMOD [p.104]].

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

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. It is being re-submitted for last call because of substantial changes in the way the Schemas are implemented to ease their use in non-XHTML contexts. After going through Last Call again, this document will be integrated into a new edition of *Modularization of XHTML* [XHTMLMOD [p.104]]. The Last Call review period ends 14 November 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.

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.104]].

The *Modularization of XHTML* [XHTMLMOD [p.104]] specifies well-defined sets of abstract modules that can be combined and extended allowing the creation of a new XHTML-family document types that are portable among XHTML-family user agents. This document provides XML Schema implementation of the abstract modules defined in *Modularization of XHTML* [XHTMLMOD [p.104]]. In addition to the schema modules themselves, the framework presented here describes a means of further extending and modifying XHTML.

To a large extent, the modularization framework presented here attempts to duplicate the modularization concepts used in XHTMLMOD. Here and throughout this document, the term "XHTMLMOD" is used to refer to [XHTMLMOD [p.104]]).

2. Terms and Definitions

This section is *normative*.

This document is the XML Schema implemantaion of abstract modules defined in *XHTML Modularization* [XHTMLMOD [p.104]] and terms used in the document are defined in *Terms and Definitions* section of *XHTML Modularization* [XHTMLMOD [p.104]].

3. Conformance Definition

This section is *normative*.

This document is the XML Schema implemantaion of abstract modules defined in *XHTML Modularization* [XHTMLMOD [p.104]] and defines conformance requirements as defined in *Conformance Definition* section of *XHTML Modularization* [XHTMLMOD [p.104]]

A. Building Schema Modules

This chapter is *normative*.

This chapter contains instructions on how to define XML Schema modules that are compatible with the XHTML Modularization implementation via XML Schema [XMLSCHEMA [p.104]]. This is in an appendix so that it can be a drop in replacement for the corresponding appendix in XHTML Modularization [XHTMLMOD [p.104]].

XHTML Schema modules are implemented as fragments. When these fragments are assembled in a specific manner (described in Developing Schemas with defined and extended modules [p.15]), the resulting Schema is a representation of a complete document type. This representation can then be used for validation of instances of the document type.

The key to combining these fragments into a meaningful, cohesive Schema is the rules used to define the fragments. This section defines those rules. When these rules are followed, markup language authors can be confident that their modules will interface cleanly with other XHTML-compatible modules.

Modules conforming to these rules also need to satisfy the conformance requirements defined in *XHTML Family Module Conformance* of XHTML Modularization [XHTMLMOD [p.104]] in order to be called *XHTML Family Modules*.

A.1. Named Content Models

This specification classifies named content model into categories and names them consistently using the following suffixes

.content
: model group definitions use the suffix `.content` when they are used to represent the content model of an element type.

.class
: model group definitions use the suffix `.class` when they are used to represent elements of the same class.

.mix
: model group definitions use the suffix `.mix` when they are used to represent a collection of element types from different classes.

.extra
: model group definitions use the suffix `.extra` when they are used to extend other groups above.

.type
: named complex type definitions use the suffix `.type` when they are used to represent type of an element.

.attlist
: attribute groups use the suffix `.attlist` when they are used to represent the attributes for a specific element.

.attrib
: attribute groups use the suffix `.attrib` when they are used to represent a group of tokens representing one or more complete attribute specifications within an .attlist declaration.

For example, in HTML 4, the `%block;` parameter entity is defined to represent the heterogeneous collection of element types that are block-level elements. In this specification, the corollary named content model is `Block.mix`.

When defining named content model in the classes defined here, modules should scope the names of the model group definitions and attribute groups by using unique prefixes. For example, the content model for the element `myelement` in the module mymodule could be named `myelement.content`. Other schemes are possible. Regardless of the scheme used, module authors should strive to ensure that named content model they define are named uniquely so that they do not collide with other named content model and so that the interface methods for the module are obvious to its users.

A.2. Defining the Namespace of a Module

XHTML requires that the elements and attributes declared in a module be within a defined XML namespace [XMLNAMES] [p.104] . The identification of this namespace is an arbitrary URI. XHTML requires that when a module is implemented using an XML Schema, the module declare its target namespace.

A.2.1. Global and Local Element Declarations

While XML Schema allows an the definition of global and local element declarations, to be compatible with DTD definitions of XHTML Modularization module implementations must not declare local elements.

A.2.2. Global and Local Attribute Declarations

While the approach defined here permits the definition of global as well as attributes declarations, schema authors should consider the consequences of such definitions on an document instance. Global attributes must always be explicitly prefixed in a instance document by declaring a namespace prefix `xmlns:prefix`, while local attributes depending on the schema implementation may be be explicitly prefixed.

A.3. Content model Redefinitions

Quite often modules change the content model of elements defined by other modules. For example, XHTML Events module add event attributes to element defined by forms module. It is also possible multiple modules may change the content model of a single element defined by a third module, for example both XHTML Events Module and XHTML Image Map module add attributes to elements in form module.

XML Schemas allows for changes to a declared content model using the `redefine` element. While XML Schema supports `redefine` element that redefines the named content model and type definition, XML Schema does not directly support redefintion of an element or attribute declaration.

To support element content model redefinitions,

- Schema module implementations must define the content model of an element using named complex types schema component `.type`. Further the named schema types must be defined in terms of named content model `.content` and `.attlist`
- `redefine` in XML Schema by default includes of the referenced schema, hence module implementations must not directly redefine content model of other modules in its implementation. The redefining module must declare the named content models (model group defintions and/or attribute groups) that may be used for redefinitions by the schema that defines the document model
- Finally, the schema defining document model (of the new document type) may redefine the content model of modules. The redefinitions define the new content model of an included element that comprise the named content models defined in both the redefined module and redefining module.

B. Developing Schema with defined and extended modules

This chapter is *informative*.

The primary purpose of defining XHTML modules and a general modularization methodology is to ease the development of document types that are based upon XHTML. These document types may extend XHTML by integrating additional capabilities (e.g., [SMIL] [p.105]), or they may define a subset of XHTML for use in a specialized device. This section describes the techniques that document type designers must use in order to take advantage of the XML Schema implementation of this modularization architecture. It does this by applying the XHTML Modularization techniques in progressively more complex ways, culminating in the creation of a complete document type from disparate modules.

Note that in no case do these examples require the modification of the XHTML-provided module *file entities* themselves. The XHTML module file entities are completely parameterized, so that it is possible through separate module definitions and *driver files* to customize the definition and the content model of each element and each element's hierarchy.

Finally, remember that most users of XHTML are *not* expected to be XML Schema authors. XML Schema authors are generally people who are defining specialized markup that will improve the readability, simplify the rendering of a document, or ease machine-processing of documents, or they are client designers that need to define the specialized markup language for their specific client. Consider these cases:

- An organization is providing subscriber's information via a Web interface. The organization stores its subscriber information in an XML-based database. One way to report that information out from the database to the Web is to embed the XML records from the database directly in the XHTML document. While it is possible to merely embed the records, the organization could define a module that describes the records, attach that module to an XHTML implementation, and thereby create a complete markup language for the pages. The organization can then access the data within the new elements via the Document Object Model [DOM] [p.103] , validate the documents, provide style definitions for the elements that cascade using Cascading Style Sheets [CSS2] [p.103] , etc. By taking the time to define the structure of their data and create a markup language using the processes defined in this section, the organization can realize the full benefits of XML.

- An Internet client developer is designing a specialized device. That device will only support a subset of XHTML, and the devices will always access the Internet via a proxy server that validates content before passing it on to the client (to minimize error handling on the client). In order to ensure that the content is valid, the developer creates a markup language that is a subset of XHTML using the processes defined in this section. They then use the new language definition in their proxy server and in their devices, and also make the language definition available to content developers so that developers can validate their content before making it available. By performing a few simple steps, the client developer can use the architecture defined in this document to greatly ease their language development cost *and* ensure that they are fully supporting the subset of XHTML that they choose to include.

B.1. Defining additional attributes

In some cases, an extension to XHTML can be as simple as additional attributes. Schema authors should an provide the attribute definitions for each attribute, for example:

```
<xs:attributeGroup name="myattrs.attrib">
    <xs:attribute name="myattribute" type="xs:string"/>
</xs:attributeGroup>
```

would declare an attribute "myattr" and attribute group "myattrs.attrib" in the target namespace of the schema ("xs" is the prefix for XML Schema Namespace). Authors should note that the attribute is created as local attribute (as part attribute group) Alternatively, declaring an attribute by placing the attribute declaration as direct child of `schema` element creates an Global attribute (and would require the document instances to use a qualified attribute names such as xlink:show). *For a discussion of qualified names and Namespace prefixes, see Defining the Namespace of a Module [p.12] .*

To add this attribute to the content model of an element, the attribute group (that makes the content model of the element) would need to be redefined to include the new attribute. for example:

```
<xs:redefine schemaLocation="xhtml-hypertext-1.xsd">
    <xs:attributeGroup name="xhtml:a.attlist">
        <xs:attributeGroup ref="xhtml:a.attlist"/>
        <xs:attributeGroup ref="myml:myattrs.attrib"/>
    </xs:attributeGroup>
</xs:redefine>
```

where "xhtml" is the prefix for XHTML Namespace. The schema modularization requires such redefinitions be defined as part of the markup language definition and the strategies for those are addressed in the later sections. Naturally, adding an attribute to a schema does not mean that any new behavior is defined for arbitrary clients. However, a content developer could use an extra attribute to store information that is accessed by associated scripts via the Document Object Model (for example).

B.2. Defining additional elements

Defining additional elements is similar to attributes, but a typical XHTML module would define the element as a global element (as a direct child of `schema` element) Schema authors should first write the element declaration for each element:

```
<!-- In the myml-module-1.xsd -->
<xs:group name="myelement.content">
    <xs:choice>
        <xs:element ref="myotherelement"/>
    </xs:choice>
</xs:group>
```

```
<xs:attributeGroup name="myelement.attlist">
    <xs:attribute name="myattribute" type="xs:string"/>
</xs:attributeGroup>

<xs:complexType name="myelement.type" mixed="true">
    <xs:choice>
        <xs:group ref="myelement.content" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
    <xs:attributeGroup ref="myelement.attlist"/>
</xs:complexType>

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

The element "myelement" is declared in the target namespace of the schema document and "xs" is the prefix for XML Schema Namespace. After the elements are defined, they need to be integrated into the content model. Strategies for integrating new elements or sets of elements into the content model are addressed in the next section.

B.3. Defining the content model for a collection of modules

Since the content model of XHTML modules is fully parameterized using named content models, Schema authors may modify the content model for every element in every module. The details of the schema module interface are defined in Building Schema Modules [p.11] . Basically there are two ways to approach this modification:

1. Re-define the named content model, `.content`, for each element.
2. Define one or more of the global named content model entities to include the element in those named model definitions (normally via the named content model, `.extras`).

The strategy taken will depend upon the nature of the modules being combined and the nature of the elements being integrated. The remainder of this section describes techniques for integrating two different classes of modules.

B.3.1. Integrating a stand-alone module into XHTML

When a module (and remember, a module can be a collection of other modules) contains elements that only reference each other in their content model, it is said to be "internally complete". As such, the module can be used on its own; (for example, you could define a schema that was just that module, and use one of its elements as the root element). Integrating such a module into XHTML is a three step process:

1. Decide what element(s) can be thought of as the root(s) of the new module.
2. Decide where these elements need to attach in the XHTML content tree.
3. Then, for each attachment point in the content tree, add the root element(s) to the content definition for the XHTML elements.

Consider attaching the elements defined above [p.16] . In that example, the element `myelement` is the root. To attach this element under the `img` element, and only the `img` element, of XHTML, the following redefinition would work:

```
<xs:redefine schemaLocation="xhtml-image-1.xsd">
    <xs:group name="img.content">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="myml:myelement"/>
        </xs:choice>
    </xs:group>
</xs:redefine>
```

Such redefinition must not included in the module definitions, but instead provided as part of the markup model definition. A schema defined with this content model would allow a document like the following fragment:

```
<img src="...">
  <myml:myelement >This is content of a locally defined element</myml:myelement>
</img>
```

It is important to note that normally the `img` element has a content model of `EMPTY`. By adding myelement to that content model, we are really just replacing `EMPTY` with `myelement`. In the case of other elements that already have content models defined, the addition of an element would require the restating of the existing content model in addition to `myelement`.

B.3.2. Mixing a new module throughout the modules in XHTML

Extending the example above, to attach this module everywhere that the `%Flow.mix` content model group is permitted, would require something like the following in the schema that defines the document model of the document type:

```
<xs:redefine schemaLocation="xhtml11-model.xsd">
    <xs:group name="Misc.extra">
        <xs:choice>
            <xs:group ref="Misc.extra"/>
            <xs:element ref="xhtml:myelement"/>
        </xs:choice>
    </xs:group>
</xs:redefine>
```

Since the %Misc.extra content model class is used in the content model the named model Misc.class, and that named model is used throughout the XHTML modules, the new module would become available throughout an extended XHTML document type.

B.4. Creating a new Document Type

So far the examples in this section have described the methods of extending XHTML and XHTML's content model. Once this is done, the next step is to collect the modules that comprise the Document Type into a schema driver and schema file that provides the content model redefinitions of included modules, incorporating the new definitions so that they override and augment the basic XHTML definitions as appropriate.

B.4.1. Creating a simple Docoment Type

Using the trivial example above, it is possible to define a new schema that uses and extends the XHTML modules pretty easily. First, define the new elements and their content model in a module:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.example.com/xmlns/simpleml1"
           xmlns="http://www.example.com/xmlns/simpleml1">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema module for module SimpleML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
  </xs:annotation>

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

  <xs:group name="otherelement.content">
     <xs:sequence/>
  </xs:group>

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

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

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

  <xs:group name="element.content">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="otherelement"/>
    </xs:choice>
  </xs:group>

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

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

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

```
  <!-- Note: Global attribute -->
  <xs:attribute name="myattr" type="xs:string"/>

</xs:schema>
```

Now, define the schema module that redefines the content model of modules included in XHTML Basic 1.0

```
<?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:simpleml="http://www.example.com/xmlns/simpleml1">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema that redefines the
      individual XHTML modules (image module) and
      redefines a named content model of XHTML modularization
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:redefine schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-basic10-model-1.xsd">
    <xs:group name="Misc.class">
      <xs:choice>
        <xs:group ref="Misc.class"/>
        <xs:element ref="simpleml:element"/>
      </xs:choice>
    </xs:group>
  </xs:redefine>

  <xs:redefine schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-image-1.xsd">
    <xs:attributeGroup name="img.attlist">
       <xs:attributeGroup ref="img.attlist"/>
       <xs:attribute ref="simpleml:myattr"/>
    </xs:attributeGroup>
  </xs:redefine>

</xs:schema>
```

Now, define the schema driver for the new language:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.example.com/xmlns/simpleml1"
           xmlns:xhtml="http://www.w3.org/1999/xhtml"
           xmlns="http://www.example.com/xmlns/simpleml1"
           blockDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Driver for new
      Document Type XHTML Basic 1.0 + SimpleML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
  </xs:annotation>
```

```
  <xs:include schemaLocation="simpleml-module-1.xsd"/>

  <xs:import namespace="http://www.w3.org/1999/xhtml"
             schemaLocation="simpleml-module-redefines-1.xsd"/>

  <xs:import namespace="http://www.w3.org/1999/xhtml"
             schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-basic10.xsd"/>

</xs:schema>
```

B.4.2. Creating a Language by extending XHTML

Next, there is the situation where a complete, additional, and complex module is added to XHTML (or to a subset of XHTML). In essence, this is the same as in the example above, the only difference being that the module being added is incorporated in the schema by reference rather than explicitly including the new definitions in the Schema.

One such complex module is the Schema for [MATHML] [p.105] . In order to combine MathML and XHTML into a single Schema, an author would just decide where MathML content should be legal in the document, and add the MathML root element to the content model at that point. First, define a new document model that instantiates the MathML Schema and connects it to the content XHTML content model by redefining the XHTML content model. Providing a redefintion of the XHTML content model by implication includes the XHTML content model in the new document content model :

```
<!-- File: math-model.xsd -->
<xs:import namespace="http://www.w3.org/1998/Math/MathML"
            schemaLocation="http://www.w3.org/Math/XMLSchema/mathml2/mathml2.xsd"/>

<xs:redefine schemaLocation="xhtml11-model.xsd">
    <xs:group name="InlSpecial.class">
        <xs:choice>
            <xs:group ref="InlSpecial.class"/>
            <xs:element ref="math:math"/>
        </xs:choice>
    </xs:group>
</xs:redefine>
```

Next, define a Schema driver that includes our new dcoument content model and hands off processing to the XHTML 1.1 driver (for example):

```
<!-- File: xhtml+mathml.xsd -->
<xs:include schemaLocation="mathml-model.xsd"/>

<xs:include schemaLocation="xhtml11.xsd"/>
```

B.4.3. Creating a Language by removing and replacing XHTML modules

Another way in which Schema authors may use XHTML modules is to define a Schema that is a subset of an XHTML family document type (because, for example, they are building devices or software that only supports a subset of XHTML). To do this simple create a Schema driver that does not include the relevant modules. Schema author should note that `redefine` in schema by default includes all the content modelof the referenced schema, authors should also not include any redefinitions of modules that they do not wish to include. The basic steps to follow are:

1. Take an XHTML family Schema as the basis of the new document type (e.g. XHML 1.1).
2. Select the modules to remove from that Schema.
3. Physically, remove `include` elements that include the non relevant modules from the Schema driver file.
4. Physically, remove `redefine` elements that redefine any non relevant modules from module redefinitions file.
5. Introduce some new modules

B.4.4. Creating a the new Document Type

Finally, some Schema authors may wish to start from scratch, using the XHTML Modularization framework as a toolkit for building a new markup language. This language must be made up of the minimal, required modules from XHTML. It may also contain other XHTML-defined modules or any other module that the author wishes to employ. In this example, we will take the XHTML required modules, add some XHTML-defined modules, and also add in the module we defined above.

The first step is to define a module that defines the elements and attributes using the XHTML provided template.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.example.com/xmlns/myml"
           xmlns="http://www.example.com/xmlns/myml">

  <xs:annotation>
    <xs:documentation>
      This is XML Schema template for MLML module
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      My Elements Module
        + myelement
        + myotherelement
```

```
        This module has no purpose other than to provide structure for some
        PCDATA content.
      </xs:documentation>
    </xs:annotation>

    <xs:attributeGroup name="myotherelement.attlist"/>

    <xs:group name="myotherelement.content">
      <xs:sequence/>
    </xs:group>

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

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

    <xs:group name="myelement.content">
      <xs:choice>
         <xs:element ref="myotherelement"/>
      </xs:choice>
    </xs:group>

    <xs:attributeGroup name="myelement.attlist">
         <xs:attribute name="myattribute" type="xs:string"/>
    </xs:attributeGroup>

    <xs:complexType name="myelement.type" mixed="true">
       <xs:choice>
          <xs:group ref="myelement.content" minOccurs="0" maxOccurs="1"/>
       </xs:choice>
       <xs:attributeGroup ref="myelement.attlist"/>
    </xs:complexType>

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

    <!-- Note: Global attribute -->
    <xs:attribute name="myattr" type="xs:string"/>

  </xs:schema>
```

Now, build a content model description that hooks the new elements and attributes into the other XHTML elements. The following example is patterned after the XHTML Basic content model, but is a complete, free-standing content model 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"
             xmlns:myml="http://www.example.com/xmlns/myml">

    <xs:annotation>
      <xs:documentation>
        This is the XML Schema module of common content models for MyML
        $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
      </xs:documentation>
      <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
```

```
</xs:annotation>

<xs:annotation>
  <xs:documentation>
    XHTML Document Model

    This module describes the groupings of elements/attributes that make up
    common content models for XHTML elements.

    XHTML has following basic content models:

        Inline.mix;           character-level elements
        Block.mix;            block-like elements, eg., paragraphs and lists
        Flow.mix;             any block or inline elements
        HeadOpts.mix;         Head Elements
        InlinePre.mix;      Special class for pre content model
        InlineNoAnchor.mix;  Content model for Anchor

    Any groups declared in this module may be used
    to create element content models, but the above are
    considered 'global' (insofar as that term applies here).

    XHTML has the following Attribute Groups
         Core.extra.attrib
         I18n.extra.attrib
         Common.extra

    The above attribute Groups are considered Global

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

 <xs:attributeGroup name="I18n.extra.attrib">
    <xs:annotation>
      <xs:documentation>
        Extended I18n attribute
      </xs:documentation>
    </xs:annotation>
 </xs:attributeGroup>

 <xs:attributeGroup name="Core.extra.attrib">
    <xs:annotation>
      <xs:documentation>
         Extended Core Attributes
      </xs:documentation>
    </xs:annotation>
 </xs:attributeGroup>

 <xs:attributeGroup name="Common.extra">
    <xs:annotation>
      <xs:documentation>
         Extended Common Attributes
      </xs:documentation>
    </xs:annotation>
 </xs:attributeGroup>

<xs:group name="HeadOpts.mix">
  <xs:choice>
```

```
      <xs:element ref="meta"/>
      <xs:element ref="link"/>
      <xs:element ref="object"/>
    </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>
      <xs:element ref="myml:myelement"/>
    </xs:choice>
  </xs:group>

  <!-- Inline Elements -->
  <xs:group name="InlStruct.class">
    <xs:choice>
      <xs:element ref="br"/>
      <xs:element ref="span"/>
    </xs:choice>
  </xs:group>

  <xs:group name="InlPhras.class">
    <xs:choice>
      <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/>
  </xs:group>

  <xs:group name="I18n.class">
    <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>
      <xs:element ref="img"/>
      <xs:element ref="object"/>
```

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

<xs:group name="InlForm.class">
  <xs:choice>
    <xs:element ref="input"/>
    <xs:element ref="select"/>
    <xs:element ref="textarea"/>
    <xs:element ref="label"/>
  </xs:choice>
</xs:group>

<xs:group name="Inline.extra">
  <xs:choice/>
</xs:group>

<!--xs:group name="Ruby.class">
  <xs:sequence/>
</xs:group-->

<!--
 Inline.class includes all inline elements,
 used as a component in mixes
-->
<xs:group name="Inline.class">
  <xs:choice>
    <xs:group ref="InlStruct.class"/>
    <xs:group ref="InlPhras.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>

<!--
  InlinePre.class
  Used as a component in pre model
-->
<xs:group name="InlinePre.mix">
  <xs:choice>
    <xs:group ref="InlStruct.class"/>
    <xs:group ref="InlPhras.class"/>
    <xs:group ref="Anchor.class"/>
    <xs:group ref="Inline.extra"/>
  </xs:choice>
</xs:group>


<!--
  InlNoAnchor.class includes all non-anchor inlines,
  used as a component in mixes
-->
<xs:group name="InlNoAnchor.class">
  <xs:choice>
    <xs:group ref="InlStruct.class"/>
    <xs:group ref="InlPhras.class"/>
    <xs:group ref="InlSpecial.class"/>
```

```
    <xs:group ref="InlForm.class"/>
    <xs:group ref="Inline.extra"/>
  </xs:choice>
</xs:group>

<!--
 InlNoAnchor.mix includes all non-anchor inlines
-->
<xs:group name="InlNoAnchor.mix">
  <xs:choice>
    <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>
    <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>
    <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>
    <xs:element ref="ul"/>
    <xs:element ref="ol"/>
    <xs:element ref="dl"/>
  </xs:choice>
</xs:group>

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

<xs:group name="Form.class">
  <xs:choice>
```

```
    <xs:element ref="form"/>
  </xs:choice>
</xs:group>

<xs:group name="BlkStruct.class">
  <xs:choice>
    <xs:element ref="p"/>
    <xs:element ref="div"/>
  </xs:choice>
</xs:group>

<xs:group name="BlkPhras.class">
  <xs:choice>
    <xs:element ref="pre"/>
    <xs:element ref="blockquote"/>
    <xs:element ref="address"/>
  </xs:choice>
</xs:group>

<xs:group name="BlkPres.class">
   <xs:choice/>
</xs:group>

<xs:group name="BlkSpecial.class">
  <xs:choice>
    <xs:group ref="Table.class"/>
    <xs:group ref="Form.class"/>
  </xs:choice>
</xs:group>

<xs:group name="Block.extra">
   <xs:choice/>
</xs:group>

<!--
 Block.class includes all block elements,
  used as an component in mixes
-->
<xs:group name="Block.class">
  <xs:choice>
    <xs:group ref="BlkStruct.class"/>
    <xs:group ref="BlkPhras.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>
    <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>
      <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:group name="FlowNoTable.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="Form.class"/>
      <xs:group ref="Inline.class"/>
      <xs:group ref="Misc.class"/>
    </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="Misc.class"/>
    </xs:choice>
  </xs:group>

</xs:schema>
```

Now, build a list of redefinitions that change the content model of various included modules. The following example is patterned after the redefinitions in XHTML Basic document type and modified as appropriate for our new markup language:

```
<?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:myml="http://www.example.com/xmlns/myml">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema changes to the content models
      of modules included in XHTML Basic 1.0 + MyML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/MarkUp/SCHEMA/xhtml-copyright-1.xsd"/>
  </xs:annotation>

  <xs:annotation>
    <xs:documentation>
      Module Content Model Redefinitions

      This schema describes the changes (Redefinitions) to
      content model of individual modules as they are instantiated as part of
      XHTML Basic 1.0 + MyML document
    </xs:documentation>
  </xs:annotation>

  <!-- Changes to Structural Module -->
  <xs:redefine schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-struct-1.xsd">
     <xs:group name="head.content">
       <xs:annotation>
         <xs:documentation>
           Redefinition by Base module
         </xs:documentation>
       </xs:annotation>
       <xs:sequence>
           <xs:group ref="HeadOpts.mix" minOccurs="0" maxOccurs="unbounded"/>
           <xs:choice>
             <xs:sequence>
                <xs:element ref="title"/>
                <xs:group ref="HeadOpts.mix" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="base" minOccurs="0"/>
             </xs:sequence>
             <xs:sequence>
               <xs:element ref="base"/>
               <xs:group ref="HeadOpts.mix" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="title"/>
             </xs:sequence>
           </xs:choice>
           <xs:group ref="HeadOpts.mix" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:group>

     <xs:attributeGroup name="version.attrib">
       <xs:annotation>
         <xs:documentation>
            Redefinition by the XHTML11 Markup (for value of version attr)
         </xs:documentation>
       </xs:annotation>
       <xs:attribute name="version" type="FPI" fixed="-//W3C//DTD XHTML Basic 1.0 + MyML//EN"/>
     </xs:attributeGroup>
  </xs:redefine>

  <xs:redefine schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-image-1.xsd">
    <xs:attributeGroup name="img.attlist">
      <xs:attributeGroup ref="img.attlist"/>
      <xs:attribute ref="myml:myattr"/>
```

```
    </xs:attributeGroup>
  </xs:redefine>

</xs:schema>
```

Finally, build a driver for a new SCHEMA. The following example is patterned after the XHTML Basic content model and modified as appropriate for our new markup language:

```
<?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"
           xmlns:myml="http://www.example.com/xmlns/myml"
           blockDefault="#all">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema driver for XHTML Basic 1.0.
      Please use this namespace for XHTML elements:
         "http://www.w3.org/1999/xhtml"

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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;1998-2003 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 is the Schema Driver file for
      XHTML Basic 1.0  + MyML Document Type

      This schema includes
        + modules for XHTML Basic 1.0 Document Type.

        + module MyML (MyML Namespace)

        + schema that defines all the named model for
          the XHTML Basic 1.0 + MyML Document Type

        + schema that redefines the content model of
          individual elements defined in the Module
          implementations.

      This Document Type includes the following Modules

        XHTML Core modules (Required for XHTML Family Conformance)
            +  text
            +  hypertext
            +  lists
            +  structure

        Other XHTML modules
          +  Link
          +  Meta
          +  Base
          +  Image
          +  Object
          +  Param
```

```
        +  Basic forms
        +  Basic tables

      Other Modules
        + MyML Module

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

<xs:include schemaLocation="myml-module-redefines-1.xsd">
  <xs:annotation>
    <xs:documentation>
       This schema redefines the content model
       defined by the individual modules for
       XHTML Basic 1.0Document Type
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="myml-model-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Document Model module for the XHTML Basic 1.0 Document Type
      This schema file defines all named models used by XHTML
      Modularization Framework for XHTML Basic 1.0 Document Type
    </xs:documentation>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-framework-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Schema Framework Component Modules:
          +  notations
          +  datatypes
          +  common attributes
          +  character entities
    </xs:documentation>
    <xs:documentation
       source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_commonatts"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/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:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_textmodule"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-hypertext-1.xsd">
  <xs:annotation>
    <xs:documentation>
       Hypertext module

       Elements defined here:
        * a
    </xs:documentation>
    <xs:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_hypertextmodule"/>
  </xs:annotation>
</xs:include>
```

```
<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-list-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Lists module

      Elements defined here:
        * dt, dd, dl, ol, ul, li
    </xs:documentation>
    <xs:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_listmodule"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-struct-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Structural module

      Elements defined here:
        * title, head, body, html
    </xs:documentation>
    <xs:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_structuremodule"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-link-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Link module

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

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-meta-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Meta module

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

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-base-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Base module

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

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-image-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Image module

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

<xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-object-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Object module

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

```
  <xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-param-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Param module

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

  <xs:include  schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml-basic-form-1.xsd">
    <xs:annotation>
      <xs:documentation>
        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="http://www.w3.org/MarkUp/SCHEMA/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:import namespace="http://www.example.com/xmlns/myml"
             schemaLocation="myml-module-1.xsd">
    <xs:annotation>
      <xs:documentation>
        MyML Module

        Elements defined here:
          * myelement, myotherelement
      </xs:documentation>
    </xs:annotation>
  </xs:import>

</xs:schema>
```

Once a new SCHEMA has been developed, it can be used in any document. Using the Schema is as simple as just referencing it in the schemaLocation attribute of a document root element:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:myml="http://www.example.com/xmlns/myml"
      xmlns:schemaLocation="http://www.w3.org/1999/xhtml myml-1_0.xsd">
<head>
<title>An example using defaults</title>
</head>
<body>
<p>This is content in the XHTML namespace</p>
<myml:myelement>
  This is content in the myml namespace.
  <myml:myotherelement />
```

```
</myml:myelement>
<p><img src="missing" alt="Missing image" myml:myattr="value" /></p>
</body>
</html>
```

C. XHTML Schema Module Implementations

This appendix is *normative*.

This appendix will contain implementations of the modules defined in XHTML Abstract Modules via XML Schemas. These module implementations can be used by XHTML Family Document Types.

C.1. Character Entities

XML Schema uses DTDs to support character entities. The DTD implementation of XHTML Modularization [XHTMLMOD] defines the required character entities for XHTML. The schema implementation includes the set of character entities for XHTML, including the Latin 1, Symbol and Special character collections defined as part the DTD implementation of XHTML Modularization [XHTMLMOD].

Even though the named character entities are defined using DTDs and included in the Schema Modularization Framework, schema processors typically do not expand the named character entities in an XML document instance. The html DOCTYPE declaration must still be present if one wishes to use named entities defined by this Framework module

C.2. XHTML Schema Modular Framework

The Framework modules instantiates a set of support model that define the common Datatypes, Notations, Common attribute definitions and character entities. These are 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">

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

      This required module instantiates the necessary modules
      needed to support the XHTML modularization framework.

      The Schema modules instantiated are:
        +  notations
        +  datatypes
        +  common attributes
        +  character entities

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:annotation>
      <xs:documentation>
        Common Attributes
        This import brings in the attributes
        xml:lang and xml:space in the XML namespace.
      </xs:documentation>
```

```
      </xs:annotation>
    </xs:import>

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

    <xs:include schemaLocation="xhtml-datatypes-1.xsd">
      <xs:annotation>
        <xs:documentation>
          This module defines XHTML Attribute DataTypes
        </xs:documentation>
        <xs:documentation
            source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstraction.html#s_common_attrtypes"/>
      </xs:annotation>
    </xs:include>

    <xs:include schemaLocation="xhtml-attribs-1.xsd">
      <xs:annotation>
        <xs:documentation>
          This module defines Common attributes for XHTML
        </xs:documentation>
        <xs:documentation
            source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_commonatts"/>
      </xs:annotation>
    </xs:include>

    <xs:include schemaLocation="xhtml-charent-1.xsd">
      <xs:annotation>
        <xs:documentation>
          Character entities module
          Note: Entities are not supported in XML Schema
          The Schema Module uses DTDs to define Entities

          This module defines
            + XHTML Latin 1 Character Entities
            + XHTML Special Characters
            + XHTML Mathematical, Greek, and Symbolic Characters
      </xs:documentation>
      </xs:annotation>
    </xs:include>

  </xs:schema>
```

The Modular Schema framework also relies upon the following component modules:

C.2.1. 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">

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

      Defines the following notations, many of these imported from
      other specifications and standards. When an existing FPI is
      known, it is incorporated here.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
```

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

C.2.2. 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">

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

          Defines containers for the XHTML datatypes, many of
          these imported from other specifications and standards.

          $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
        </xs:documentation>
        <xs:documentation source="xhtml-copyright-1.xsd"/>
        <xs:documentation
            source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstraction.html#s_common_attrtypes"/>
    </xs:annotation>

    <!-- nn for pixels or nn% for percentage length -->
    <xs:simpleType name="Length">
        <xs:union memberTypes="xs:nonNegativeInteger">
           <xs:simpleType>
             <xs:restriction base="xs:token">
               <xs:pattern value="\d+[%]|\d*\.\d+[%]"/>
              </xs:restriction>
           </xs:simpleType>
        </xs:union>
    </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:union memberTypes="Length">
       <xs:simpleType>
         <xs:restriction base="xs:token">
           <xs:pattern value="\d*\*"/>
         </xs:restriction>
       </xs:simpleType>
    </xs:union>
</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>

<!-- sixteen color names or RGB color expression-->
<xs:simpleType name="Color">
    <xs:union memberTypes="xs:NMTOKEN">
       <xs:simpleType>
          <xs:restriction base="xs:token">
             <xs:pattern value="#[0-9a-fA-F]{6}"/>
          </xs:restriction>
       </xs:simpleType>
    </xs:union>
</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:length value="1" fixed="true"/>
    </xs:restriction>
</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="Charset"/>
</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:dateTime"/>
</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:language"/>
    </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:anyURI"/>
    </xs:simpleType>

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

    <!-- character Data -->
    <xs:simpleType name="CDATA">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>

</xs:schema>
```

C.2.3. 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:xml="http://www.w3.org/XML/1998/namespace"
           xmlns="http://www.w3.org/1999/xhtml">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema common attributes module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </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.attrib">
    <xs:attributeGroup ref="id"/>
    <xs:attributeGroup ref="class"/>
    <xs:attributeGroup ref="title"/>
    <xs:attributeGroup ref="Core.extra.attrib"/>
  </xs:attributeGroup>
```

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

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

</xs:schema>
```

C.2.4. 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"
   "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-lat1.ent">
%HTMLlat1;
<!-- These are the entity sets for special characters for the XHTML -->
<!ENTITY % HTMLsymbol PUBLIC
   "-//W3C//ENTITIES Symbols for XHTML//EN"
   "http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-symbol.ent">
%HTMLsymbol;
<!-- These are the entity sets for symbol characters for the XHTML -->
<!ENTITY % HTMLspecial PUBLIC
   "-//W3C//ENTITIES Special for XHTML//EN"
   "http://www.w3.org/TR/xhtml-modularization/DTD/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">

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

      This module declares the set of character entities for XHTML,
      including the Latin 1, Symbol and Special character collections.
      XML Schema does not support Entities, hence Entities are enable
      through an Internal DTD Subset.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>

</xs:schema>
```

C.3. XHTML Module Implementations

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

C.3.1. XHTML Core Modules

C.3.1.1. Structure

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

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

          * title, head, body, html

        The Structure Module defines the major structural elements and
        their attributes.

        $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
      </xs:documentation>
      <xs:documentation source="xhtml-copyright-1.xsd"/>
      <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_structuremodule"/>
    </xs:annotation>

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

    <xs:group name="title.content">
      <xs:sequence/>
    </xs:group>

    <xs:complexType name="title.type" mixed="true">
      <xs:group ref="title.content"/>
      <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" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="title"/>
        <xs:group ref="HeadOpts.mix" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:group>

    <xs:attributeGroup name="profile.attrib">
      <xs:attribute name="profile" type="URI"/>
    </xs:attributeGroup>

    <xs:attributeGroup name="head.attlist">
      <xs:attributeGroup ref="profile.attrib"/>
      <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:group name="body.content">
      <xs:sequence>
        <xs:group ref="Block.mix" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:group>

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

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

    <xs:attributeGroup name="version.attrib">
      <xs:attribute name="version" type="FPI"/>
    </xs:attributeGroup>

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

    <xs:group name="html.content">
      <xs:sequence>
         <xs:element ref="head"/>
         <xs:element ref="body"/>
      </xs:sequence>
    </xs:group>

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

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

</xs:schema>
```

C.3.1.2. 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">

  <xs:annotation>
    <xs:documentation>
      Textual Content
      This is the XML Schema Text module for XHTML

      The Text module includes declarations for all core
      text container elements and their attributes.

        +  block phrasal
        +  block structural
        +  inline phrasal
        +  inline structural

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_textmodule"/>
  </xs:annotation>

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

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

</xs:schema>
```

C.3.1.3. 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">

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

        * 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.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_hypertextmodule"/>
  </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:group name="a.content">
     <xs:sequence>
```

```
        <xs:group ref="InlNoAnchor.mix" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.3.1.4. 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">

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

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

      This module declares the list-oriented element types
      and their attributes.
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_listmodule"/>
  </xs:annotation>

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

  <xs:group name="dt.content">
    <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="dt.type" mixed="true">
    <xs:group ref="dt.content"/>
    <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:group name="dd.content">
    <xs:sequence>
      <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="dd.type" mixed="true">
    <xs:group ref="dd.content"/>
    <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:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="dt"/>
        <xs:element ref="dd"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="dl.type">
    <xs:group ref="dl.content"/>
    <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:group name="li.content">
    <xs:sequence>
      <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="li.type" mixed="true">
    <xs:group ref="li.content"/>
    <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:group name="ol.content">
    <xs:sequence>
      <xs:element ref="li" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="ol.type">
    <xs:group ref="ol.content"/>
    <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:group name="ul.content">
    <xs:sequence>
      <xs:element ref="li" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.3.2. 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">

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

        * applet (param)

      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 also instantiates of the
      Param Element Module.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_appletmodule"/>
  </xs:annotation>

  <xs:include schemaLocation="xhtml-param-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Param module
        Include Param Module
      </xs:documentation>
    </xs:annotation>
  </xs:include>

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

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

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

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

</xs:schema>
```

C.3.3. Text Modules

C.3.3.1. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Presentation module for XHTML
      This is a REQUIRED module.
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_presentationmodule"/>
  </xs:annotation>

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

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

  <xs:include schemaLocation="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>
```

C.3.3.2. 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">
    <xs:annotation>
      <xs:documentation>
          Editing Elements
          This is the XML Schema Editing Markup module for XHTML

           * ins, del

         This module declares element types and attributes used to indicate
         inserted and deleted content while editing a document.

          $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
        </xs:documentation>
      <xs:documentation source="xhtml-copyright-1.xsd"/>
```

```
        <xs:documentation
           source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_editmodule"/>
    </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:sequence>
            <xs:group ref="Flow.mix"  minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:group>

    <xs:complexType name="edit.type" mixed="true">
        <xs:group ref="edit.content"/>
        <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>
```

C.3.3.3. Bi-directional Text

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
            xmlns="http://www.w3.org/1999/xhtml"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xml="http://www.w3.org/XML/1998/namespace">
    <xs:annotation>
       <xs:documentation>
          Bidirectional Override (bdo) Element
          This is the XML Schema BDO Element module for XHTML

          This modules declares the element 'bdo' and 'dir' attributes,
          Used to override the  Unicode bidirectional algorithm for selected
          fragments of text.
          Bidirectional text support includes both the bdo element and
          the 'dir' attribute.

          $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
      </xs:documentation>
      <xs:documentation source="xhtml-copyright-1.xsd"/>
      <xs:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_bdomodule"/>
    </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:group name="bdo.content">
       <xs:sequence>
          <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:group>

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

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

    <xs:attributeGroup name="dir.attrib">
      <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:schema>
```

C.3.4. Forms

C.3.4.1. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Basic Forms module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_sformsmodule"/>
  </xs:annotation>

  <xs:attributeGroup name="form.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="action" type="URI" use="required"/>
    <xs:attribute name="method" use="optional" default="get">
      <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" default="application/x-www-form-urlencoded"/>
  </xs:attributeGroup>

  <xs:group name="form.content">
    <xs:sequence>
      <xs:group ref="BlkNoForm.mix" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

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

```
  </xs:complexType>

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

  <xs:group name="label.content">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="input"/>
        <xs:element ref="select"/>
        <xs:element ref="textarea"/>
        <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="Misc.class"/>
      </xs:choice>
    </xs:sequence>
  </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"/>
    <xs:attributeGroup ref="label.attlist"/>
  </xs:complexType>

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

<!--
Basic Forms removes button, 'image' and 'file' input types.
-->
  <xs:simpleType name="InputType.class">
      <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:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="input.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="type" type="InputType.class" default="text"/>
    <xs:attribute name="name" type="CDATA"/>
    <xs:attribute name="value" type="CDATA"/>
    <xs:attribute name="checked">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <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="tabindex" type="Number"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:group name="input.content">
     <xs:sequence/>
  </xs:group>

  <xs:complexType name="input.type">
    <xs:group ref="input.content"/>
    <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="CDATA"/>
    <xs:attribute name="size" type="Number"/>
    <xs:attribute name="multiple">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="multiple"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="tabindex" type="Number"/>
  </xs:attributeGroup>

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

  <xs:complexType name="select.type">
    <xs:group ref="select.content"/>
    <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">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="selected"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="value" type="CDATA"/>
    <xs:attribute name="tabindex" type="Number"/>
  </xs:attributeGroup>

  <xs:group name="option.content">
     <xs:sequence/>
  </xs:group>

  <xs:complexType name="option.type" mixed="true">
    <xs:group ref="option.content"/>
    <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="CDATA"/>
    <xs:attribute name="rows" type="Number" use="required"/>
    <xs:attribute name="cols" type="Number" use="required"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:group name="textarea.content">
     <xs:sequence/>
  </xs:group>

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

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

</xs:schema>
```

C.3.4.2. 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">

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

        * 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.


      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_extformsmodule"/>
  </xs:annotation>

  <!-- form: Form Element -->
  <xs:attributeGroup name="form.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="action" type="URI" use="required"/>
    <xs:attribute name="method" default="get">
      <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" default="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:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:group ref="BlkNoForm.mix"/>
        <xs:element ref="fieldset"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>

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

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

  <!-- input: Form Control -->
  <xs:simpleType name="InputType.class">
      <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>

<!--
    attribute 'name' required for all but submit & reset
-->
  <xs:attributeGroup name="input.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="type" type="InputType.class" default="text"/>
    <xs:attribute name="name" type="CDATA"/>
    <xs:attribute name="value" type="CDATA"/>
    <xs:attribute name="checked">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="checked"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="disabled">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="disabled"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="readonly">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <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:group name="input.content">
     <xs:sequence/>
  </xs:group>

  <xs:complexType name="input.type">
    <xs:group ref="input.content"/>
    <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="CDATA"/>
  <xs:attribute name="size" type="Number"/>
  <xs:attribute name="multiple">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="multiple"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="tabindex" type="Number"/>
</xs:attributeGroup>

<xs:group name="select.content">
  <xs:sequence>
    <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element ref="optgroup"/>
      <xs:element ref="option"/>
    </xs:choice>
  </xs:sequence>
</xs:group>

<xs:complexType name="select.type">
   <xs:group ref="select.content"/>
   <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:NMTOKEN">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="label" type="Text" use="required"/>
</xs:attributeGroup>

<xs:group name="optgroup.content">
  <xs:sequence>
     <xs:element ref="option" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

<xs:complexType name="optgroup.type">
  <xs:group ref="optgroup.content"/>
  <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:NMTOKEN">
        <xs:enumeration value="selected"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
```

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

<xs:group name="option.content">
  <xs:sequence/>
</xs:group>

<xs:complexType name="option.type" mixed="true">
  <xs:group ref="option.content"/>
  <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="CDATA"/>
  <xs:attribute name="rows" type="Number" use="required"/>
  <xs:attribute name="cols" type="Number" use="required"/>
  <xs:attribute name="disabled">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="disabled"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="readonly">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <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:group name="textarea.content">
  <xs:sequence/>
</xs:group>

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

<xs:complexType name="fieldset.type" mixed="true">
  <xs:group ref="fieldset.content"/>
  <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:group name="legend.content">
    <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="legend.type" mixed="true">
    <xs:group ref="legend.content"/>
    <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="CDATA"/>
    <xs:attribute name="value" type="CDATA"/>
    <xs:attribute name="type" 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:NMTOKEN">
          <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:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <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:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.3.5. Tables

C.3.5.1. 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">

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

```
    $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_simpletablemodule"/>
</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="valign">
    <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="scope">
    <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" 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:attributeGroup>

<xs:group name="caption.content">
   <xs:sequence>
```

```
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>


<xs:complexType name="caption.type" mixed="true" >
  <xs:group ref="caption.content"/>
  <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:sequence>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="th"/>
      <xs:element ref="td"/>
    </xs:choice>
  </xs:sequence>
</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="CDATA"/>
  <xs:attribute name="headers" type="xs:IDREFS"/>
  <xs:attributeGroup ref="scope.attrib"/>
  <xs:attribute name="rowspan" type="Number" default="1"/>
  <xs:attribute name="colspan" type="Number" default="1"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
  <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="th.content">
   <xs:sequence>
      <xs:group ref="FlowNoTable.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="th.type" mixed="true">
  <xs:group ref="th.content"/>
  <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="CDATA"/>
  <xs:attribute name="headers" type="xs:IDREFS"/>
  <xs:attributeGroup ref="scope.attrib"/>
  <xs:attribute name="rowspan" type="Number" default="1"/>
  <xs:attribute name="colspan" type="Number" default="1"/>
  <xs:attributeGroup ref="CellHAlign.attrib"/>
  <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="td.content">
   <xs:sequence>
      <xs:group ref="FlowNoTable.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

</xs:schema>
```

C.3.5.2. Tables

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns="http://www.w3.org/1999/xhtml"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:annotation>
        <xs:documentation>
          This is the XML Schema Tables module for XHTML
          $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
           source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_tablemodule"/>
    </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="CDATA"/>
    <xs:attribute name="headers" type="xs:IDREFS"/>
    <xs:attributeGroup ref="scope.attrib"/>
    <xs:attribute name="rowspan" type="Number" default="1"/>
    <xs:attribute name="colspan" type="Number" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="td.content">
   <xs:sequence>
      <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="td.type" mixed="true">
    <xs:group ref="td.content"/>
    <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="CDATA"/>
    <xs:attribute name="headers" type="xs:IDREFS"/>
    <xs:attributeGroup ref="scope.attrib"/>
    <xs:attribute name="rowspan" type="Number" default="1"/>
    <xs:attribute name="colspan" type="Number" default="1"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="th.content">
   <xs:sequence>
      <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="th.type" mixed="true">
    <xs:group ref="th.content"/>
    <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:sequence>
    <xs:choice maxOccurs="unbounded">
        <xs:element ref="th"/>
        <xs:element ref="td"/>
    </xs:choice>
  </xs:sequence>
</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" default="1"/>
    <xs:attribute name="width" type="MultiLength"/>
    <xs:attributeGroup ref="CellHAlign.attrib"/>
    <xs:attributeGroup ref="CellVAlign.attrib"/>
</xs:attributeGroup>

<xs:group name="col.content">
   <xs:sequence/>
</xs:group>

<xs:complexType name="col.type">
    <xs:group ref="col.content"/>
    <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" 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>
        <xs:element ref="tr" maxOccurs="unbounded"/>
    </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>
        <xs:element ref="tr" maxOccurs="unbounded"/>
    </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>
        <xs:element ref="tr" maxOccurs="unbounded"/>
    </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>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

</xs:schema>
```

C.3.6. 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">

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

        * img

      This module provides markup to support basic image embedding.

      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.


      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_imagemodule"/>
  </xs:annotation>


  <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:group name="img.content">
     <xs:sequence/>
  </xs:group>

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

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

</xs:schema>
```

C.3.7. 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">


  <xs:annotation>
```

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

        * area, map

      This module declares elements and attributes to support client-side
      image maps.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_imapmodule"/>
  </xs:annotation>

  <xs:simpleType name="Shape.Datatype">
    <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:simpleType name="Coords.Datatype">
     <xs:restriction base="Text"/>
  </xs:simpleType>

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

  <xs:attributeGroup name="area.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="href" type="URI"/>
    <xs:attribute name="shape" type="Shape.Datatype" default="rect"/>
    <xs:attribute name="coords" type="Coords.Datatype"/>
    <xs:attribute name="nohref">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="nohref"/>
          </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="alt" type="Text" use="required"/>
    <xs:attribute name="tabindex" type="Number"/>
    <xs:attribute name="accesskey" type="Character"/>
  </xs:attributeGroup>

  <xs:group name="area.content">
     <xs:sequence/>
  </xs:group>

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

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

  <!-- map -->
  <xs:attributeGroup name="map.attlist">
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attributeGroup ref="class"/>
    <xs:attributeGroup ref="title"/>
    <xs:attributeGroup ref="Core.extra.attrib"/>
    <xs:attributeGroup ref="I18n.attrib"/>
  </xs:attributeGroup>

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

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

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

</xs:schema>
```

C.3.8. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Server-side Image Maps module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>


  <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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_servermapmodule"/>
  </xs:annotation>

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

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

</xs:schema>
```

C.3.9. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Embedded Object module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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 (param)
    </xs:documentation>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_objectmodule"/>
  </xs:annotation>


  <xs:include schemaLocation="xhtml-param-1.xsd">
    <xs:annotation>
      <xs:documentation>
        Param module

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

  <xs:attributeGroup name="object.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
    <xs:attribute name="declare">
      <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
           <xs:enumeration value="declare"/>
         </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <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="CDATA"/>
    <xs:attribute name="tabindex" type="Number"/>
  </xs:attributeGroup>

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

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

```
    <xs:group ref="object.content"/>
    <xs:attributeGroup ref="object.attlist"/>
  </xs:complexType>

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

</xs:schema>
```

C.3.10. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Frames module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_framesmodule"/>
  </xs:annotation>

  <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" 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">
```

```
        <xs:simpleType>
           <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="noresize"/>
           </xs:restriction>
        </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="scrolling" 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:group name="frame.content">
      <xs:sequence/>
   </xs:group>

   <xs:complexType name="frame.type">
     <xs:group ref="frame.content"/>
     <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"/>
     </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>
```

C.3.11. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Target module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_targetmodule"/>
  </xs:annotation>
```

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

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

</xs:schema>
```

C.3.12. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Frame Element module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_iframemodule"/>
  </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" 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" 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:group name="iframe.content">
     <xs:sequence>
        <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.3.13. Intrinsic Events

```
<?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">

  <xs:annotation>
    <xs:documentation>
      Intrinsic Event Attributes
      This is the XML Schema Intrinsic Events module for XHTML

      These are the event attributes defined in HTML 4,
      Section 18.2.3 "Intrinsic Events".

      "Note: Authors of HTML documents are advised that changes
       are likely to occur in the realm of intrinsic events
       (e.g., how scripts are bound to events). Research in
       this realm is carried on by members of the W3C Document
       Object Model Working Group (see the W3C Web site at
       http://www.w3.org/ for more information)."

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_intrinsiceventsmodule"/>
  </xs:annotation>

  <xs:attributeGroup name="Events.attrib">
    <xs:attribute name="onclick" type="Script"/>
    <xs:attribute name="ondblclick" type="Script"/>
    <xs:attribute name="onmousedown" type="Script"/>
    <xs:attribute name="onmouseup" type="Script"/>
    <xs:attribute name="onmouseover" type="Script"/>
    <xs:attribute name="onmousemove" type="Script"/>
    <xs:attribute name="onmouseout" type="Script"/>
    <xs:attribute name="onkeypress" type="Script"/>
    <xs:attribute name="onkeydown" type="Script"/>
    <xs:attribute name="onkeyup" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on anchor element
-->
  <xs:attributeGroup name="a.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on form element
-->
  <xs:attributeGroup name="form.events.attlist">
    <xs:attribute name="onsubmit" type="Script"/>
    <xs:attribute name="onreset" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on label element
-->
```

```
  <xs:attributeGroup name="label.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on input element
-->
  <xs:attributeGroup name="input.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
    <xs:attribute name="onselect" type="Script"/>
    <xs:attribute name="onchange" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on select element
-->
  <xs:attributeGroup name="select.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
    <xs:attribute name="onchange" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on textarea element
-->
  <xs:attributeGroup name="textarea.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
    <xs:attribute name="onselect" type="Script"/>
    <xs:attribute name="onchange" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on button element
-->
  <xs:attributeGroup name="button.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on body element
-->
  <xs:attributeGroup name="body.events.attlist">
    <xs:attribute name="onload" type="Script"/>
    <xs:attribute name="onunload" type="Script"/>
  </xs:attributeGroup>

<!--
    additional attributes on area element
-->
  <xs:attributeGroup name="area.events.attlist">
    <xs:attribute name="onfocus" type="Script"/>
    <xs:attribute name="onblur" type="Script"/>
  </xs:attributeGroup>

</xs:schema>
```

C.3.14. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Metainformation module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
      source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_metamodule"/>
</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="CDATA" use="required"/>
  <xs:attribute name="scheme" type="CDATA"/>
</xs:attributeGroup>

<xs:group name="meta.content">
   <xs:sequence/>
</xs:group>

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

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

</xs:schema>
```

C.3.15. Scripting

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
           xmlns:xml="http://www.w3.org/XML/1998/namespace"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.w3.org/1999/xhtml">
    <xs:annotation>
        <xs:documentation>
      This is the XML Schema Scripting module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_scriptmodule"/>
    </xs:annotation>

    <xs:attributeGroup name="script.attlist">
        <xs:attribute name="charset" type="Charset"/>
        <xs:attribute name="type" type="ContentType" use="required"/>
        <xs:attribute name="src" type="URI"/>
        <xs:attribute name="defer">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="defer"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute ref="xml:space" fixed="preserve"/>
    </xs:attributeGroup>

    <xs:group name="script.content">
       <xs:sequence/>
    </xs:group>

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

```
        <xs:group ref="script.content"/>
        <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:group name="noscript.content">
       <xs:sequence>
          <xs:group ref="Block.mix" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:group>

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

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

C.3.16. Style Sheet

```
<?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:xml="http://www.w3.org/XML/1998/namespace"
           xmlns="http://www.w3.org/1999/xhtml">
    <xs:annotation>
        <xs:documentation>
      This is the XML Schema Stylesheets module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_stylemodule"/>
    </xs:annotation>

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

    <xs:group name="style.content">
       <xs:sequence/>
    </xs:group>

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

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

</xs:schema>
```

C.3.17. 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">

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

         * styloe attribute

      This module declares the 'style' attribute, used to support inline
      style markup.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_styleattributemodule"/>
  </xs:annotation>

  <xs:attributeGroup name="style.attrib">
    <xs:attribute name="style" type="CDATA"/>
  </xs:attributeGroup>

</xs:schema>
```

C.3.18. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Link Element module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_linkmodule"/>
  </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:group name="link.content">
     <xs:sequence/>
  </xs:group>

  <xs:complexType name="link.type">
```

```
    <xs:group ref="link.content"/>
    <xs:attributeGroup ref="link.attlist"/>
  </xs:complexType>

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

</xs:schema>
```

C.3.19. 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">
    <xs:annotation>
      <xs:documentation>
        Base element
        This is the XML Schema Base Element module for XHTML

          * 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.

        $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
      </xs:documentation>
      <xs:documentation source="xhtml-copyright-1.xsd"/>
      <xs:documentation
          source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_basemodule"/>
    </xs:annotation>

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

    <xs:group name="base.content">
       <xs:sequence/>
    </xs:group>

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

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

</xs:schema>
```

C.3.20. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Name Identifier module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_nameidentmodule"/>
  </xs:annotation>


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

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

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

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

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

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

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

</xs:schema>
```

C.3.21. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Module for HTML Legacy Markup

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

       This optional module declares additional markup for simple
       presentation-related markup based on features found in the
       HTML 4 Transitional and Frameset DTDs.

       The legacy module also include frames module, iframe module
       and target module. (Note: This module expects find the schema files
       of that declare these module)

       Elements/Attributes defined in frame, iframe and targer modules are

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

       $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_legacymodule"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_framesmodule"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_targetmodule"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_iframemodule"/>
  </xs:annotation>
```

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

        Elements defined here:

          * frameset, frame, noframes
      </xs:documentation>
      <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_framesmodule"/>
    </xs:annotation>
  </xs:include>

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

        Attributes defined here:

          * target
      </xs:documentation>
      <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_targetmodule"/>
    </xs:annotation>
  </xs:include>

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

        Elements defined here:

          * iframe
      </xs:documentation>
      <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_iframemodule"/>
    </xs:annotation>
  </xs:include>

</xs:schema>
```

C.3.22. 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">

  <xs:annotation>
    <xs:documentation>
      This is the Ruby module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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.

    This module expects the document model to define the
    following content models
      + InlNoRuby.mix

  </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.complex">
  <xs:sequence>
    <xs:element ref="rbc"/>
    <xs:element ref="rtc" maxOccurs="2"/>
  </xs:sequence>
</xs:group>

<!--
 add to this group any common attributes for all Ruby elements
-->
```

```
<xs:attributeGroup name="ruby.common.attrib"/>

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

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

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

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

<xs:group name="rbc.content">
  <xs:sequence>
    <xs:element ref="rb"/>
  </xs:sequence>
</xs:group>

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

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

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

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

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

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

<!--
```

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

<xs:group name="rb.content">
  <xs:sequence>
     <xs:group ref="InlNoRuby.mix" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

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

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

<!--
 rt (ruby text) element
-->
<xs:attributeGroup name="rt.attlist">
  <xs:attributeGroup ref="ruby.common.attrib"/>
  <xs:attribute name="rbspan" type="Number" default="1"/>
</xs:attributeGroup>

<xs:group name="rt.content">
  <xs:sequence>
     <xs:group ref="InlNoRuby.mix" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>

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

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

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

<xs:group name="rp.content">
  <xs:sequence/>
</xs:group>


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

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

</xs:schema>
```

C.4. XHTML Schema Support Modules

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

C.4.1. Block Phrasal

```
<?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">

    <xs:annotation>
        <xs:documentation>
        Block Phrasal elements module
        This is the XML Schema block phrasal element module for XHTML

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

        This module declares the elements and their attributes used to
        support block-level phrasal markup.

        $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
      </xs:documentation>
        <xs:documentation source="xhtml-copyright-1.xsd"/>
    </xs:annotation>

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

    <xs:group name="address.content">
       <xs:sequence>
          <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:group>

    <xs:complexType name="address.type" mixed="true">
        <xs:group ref="address.content"/>
        <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:group name="blockquote.content">
   <xs:sequence>
      <xs:group ref="Block.mix" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="pre.content">
   <xs:sequence>
      <xs:group ref="InlinePre.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

<xs:attributeGroup name="h1.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

<xs:group name="h1.content">
   <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="h1.type" mixed="true">
    <xs:group ref="h1.content"/>
```

```
    <xs:attributeGroup ref="h1.attlist"/>
</xs:complexType>

<xs:element name="h1" type="h1.type"/>

<xs:attributeGroup name="h2.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

<xs:group name="h2.content">
   <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="h2.type" mixed="true">
    <xs:group ref="h2.content"/>
    <xs:attributeGroup ref="h2.attlist"/>
</xs:complexType>

<xs:element name="h2" type="h2.type"/>

<xs:attributeGroup name="h3.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

<xs:group name="h3.content">
   <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="h3.type" mixed="true">
    <xs:group ref="h3.content"/>
    <xs:attributeGroup ref="h3.attlist"/>
</xs:complexType>

<xs:element name="h3" type="h3.type"/>

<xs:attributeGroup name="h4.attlist">
    <xs:attributeGroup ref="Common.attrib"/>
</xs:attributeGroup>

<xs:group name="h4.content">
   <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="h4.type" mixed="true">
    <xs:group ref="h4.content"/>
    <xs:attributeGroup ref="h4.attlist"/>
</xs:complexType>

<xs:element name="h4" type="h4.type"/>

<xs:attributeGroup name="h5.attlist">
```

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

    <xs:group name="h5.content">
       <xs:sequence>
          <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:group>

    <xs:complexType name="h5.type" mixed="true">
        <xs:group ref="h5.content"/>
        <xs:attributeGroup ref="h5.attlist"/>
    </xs:complexType>

    <xs:element name="h5" type="h5.type"/>

    <xs:attributeGroup name="h6.attlist">
        <xs:attributeGroup ref="Common.attrib"/>
    </xs:attributeGroup>

    <xs:group name="h6.content">
       <xs:sequence>
          <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:group>

    <xs:complexType name="h6.type" mixed="true">
        <xs:group ref="h6.content"/>
        <xs:attributeGroup ref="h6.attlist"/>
    </xs:complexType>

    <xs:element name="h6" type="h6.type"/>

</xs:schema>
```

C.4.2. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML SchemaBlock presentation element module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
         source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_presentationmodule"/>
  </xs:annotation>

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

```
  </xs:attributeGroup>

  <xs:group name="hr.content">
    <xs:sequence/>
  </xs:group>

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

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

</xs:schema>
```

C.4.3. 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">

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

        * div, p

      This module declares the elements and their attributes used to
      support block-level structural markup.

      This is the XML Schema Block Structural module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>

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

  <xs:group name="div.content">
     <xs:sequence>
        <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
  </xs:group>

  <xs:complexType name="div.type" mixed="true">
    <xs:group ref="div.content"/>
    <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:group name="p.content">
```

```
    <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.4.4. 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">

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

        * 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.
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>


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

  <xs:group name="abbr.content">
    <xs:sequence>
      <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

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

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

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

```
<xs:group name="acronym.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="cite.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="code.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="dfn.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>
```

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

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

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

<xs:group name="em.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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


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

<xs:group name="kbd.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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


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

<xs:group name="samp.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

```
</xs:complexType>

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


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

<xs:group name="strong.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="var.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

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

<xs:group name="q.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

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

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

</xs:schema>
```

C.4.5. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Inline Presentation element module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_presentationmodule"/>
  </xs:annotation>

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

  <xs:group name="InlPres.content">
    <xs:sequence>
       <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="InlPres.type" mixed="true">
    <xs:group ref="InlPres.content"/>
    <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>
```

C.4.6. 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">

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

```
      Inline Structural
      This is the XML Schema Inline Structural element module for XHTML

        * br, span

      This module declares the elements and their attributes
      used to support inline-level structural markup.

      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane Exp $
    </xs:documentation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>

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

  <xs:group name="br.content">
     <xs:sequence/>
  </xs:group>

  <xs:complexType name="br.type">
    <xs:group ref="br.content"/>
    <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:group name="span.content">
    <xs:sequence>
       <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

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

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

</xs:schema>
```

C.4.7. 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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Param Element module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_objectmodule"/>
    <xs:documentation
        source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_appletmodule"/>
  </xs:annotation>

  <xs:attributeGroup name="param.attlist">
    <xs:attributeGroup ref="id"/>
    <xs:attribute name="name" type="CDATA" use="required"/>
    <xs:attribute name="value" type="CDATA"/>
    <xs:attribute name="valuetype" default="data">
      <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:group name="param.content">
     <xs:sequence/>
  </xs:group>

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

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

</xs:schema>
```

C.4.8. Legacy - Miscellaneous

```
<?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">

  <xs:annotation>
    <xs:documentation>
      This is the XML Schema Miscellaneous Legacy
      Markup module for XHTML
      $Id: xhtml-m12n-schema.ps,v 1.3 2003/10/03 16:33:26 shane 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

      This is to allow XHTML documents to be transformed for
      display on HTML browsers where CSS support is inconsistent
      or unavailable.

      The module also declares legacy attributes for elements
      in other module. Note: This module only declares the
      attribute list, and it upto the document type to
```

```
    redefine the model of affected modules.
  </xs:documentation>
  <xs:documentation
       source="http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/abstract_modules.html#s_legacymodule"/>
</xs:annotation>

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

<xs:group name="font.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="font.type" mixed="true">
   <xs:group ref="font.content"/>
   <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="CDATA" use="required"/>
  <xs:attribute name="color" type="Color"/>
  <xs:attribute name="face" type="CDATA"/>
</xs:attributeGroup>

<xs:group name="basefont.content">
   <xs:sequence/>
</xs:group>

<xs:complexType name="basefont.type">
  <xs:group ref="basefont.content"/>
  <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:group name="center.content">
   <xs:sequence>
     <xs:group ref="Flow.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="center.type" mixed="true">
  <xs:group ref="center.content"/>
  <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:group name="s.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="s.type" mixed="true">
  <xs:group ref="s.content"/>
  <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:group name="strike.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="strike.type" mixed="true">
  <xs:group ref="strike.content"/>
  <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:group name="u.content">
   <xs:sequence>
     <xs:group ref="Inline.mix" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>

<xs:complexType name="u.type" mixed="true">
  <xs:group ref="u.content"/>
  <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">
     <xs:simpleType>
       <xs:restriction base="xs:NMTOKEN">
         <xs:enumeration value="compact"/>
       </xs:restriction>
     </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

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

<xs:complexType name="dir.type" mixed="true">
  <xs:group ref="dir.content"/>
  <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">
     <xs:simpleType>
       <xs:restriction base="xs:NMTOKEN">
         <xs:enumeration value="compact"/>
       </xs:restriction>
     </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

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

<xs:complexType name="menu.type" mixed="true">
  <xs:group ref="menu.content"/>
  <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:group name="isindex.content">
  <xs:sequence/>
</xs:group>

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

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

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

<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" 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="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">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="noshade"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <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="Pixels"/>
    <xs:attribute name="vspace" type="Pixels"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="input.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: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="CDATA"/>
    <xs:attribute name="value" type="Number"/>
  </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="Pixels"/>
  <xs:attribute name="vspace" type="Pixels"/>
</xs:attributeGroup>

<xs:attributeGroup name="ol.legacy.attlist">
  <xs:attribute name="type" type="Text"/>
  <xs:attribute name="compact">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="compact"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="start" type="Number"/>
</xs:attributeGroup>

<xs:attributeGroup name="dl.legacy.attlist">
    <xs:attribute name="compact">
      <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="compact"/>
         </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
</xs:attributeGroup>


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

<xs:attributeGroup name="heading.legacy.attlist">
  <xs:attributeGroup ref="align.legacy.attlist"/>
</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: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="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">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
```

```
          <xs:enumeration value="nowrap"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <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">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="nowrap"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <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="CDATA"/>
    <xs:attribute name="compact">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="compact"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:attributeGroup>

</xs:schema>
```

D. References

This appendix is *normative*.

D.1. Normative References

[CSS2]
"*Cascading Style Sheets, level 2 (CSS2) Specification*", W3C Recommendation, B. Bos, H. W. Lie, C. Lilley, I. Jacobs, *eds.*, 12 May 1998.
Available at: http://www.w3.org/TR/1998/REC-CSS2-19980512

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

[HTML4]
"*HTML 4.01 Specification: W3C Recommendation*", W3C Recommendation, D. Raggett, A. Le Hors, I. Jacobs, *eds.*, 24 December 1999.
Available at: http://www.w3.org/TR/1999/REC-html401-19991224

[ISO10646]
"*Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane*", ISO/IEC 10646-1:2000. This reference refers to a set of codepoints that may evolve as new characters are assigned to them. Also, this reference assumes that the character sets defined by ISO 10646 and [UNICODE] [p.104] remain character-by-character equivalent. This reference also includes future publications of other parts of 10646 (i.e., other than Part 1) that define characters in planes 1-16.

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

[RFC1808]
"*Relative Uniform Resource Locators*", RFC 1808, R. Fielding, June 1995.
Available at: http://www.ietf.org/rfc/rfc1808.txt

[RFC2045]
"*Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*", RFC 2045, N. Freed, N. Borenstein, November 1996.
Available at: http://www.ietf.org/rfc/rfc2045.txt

[RFC2119]
"*Key words for use in RFCs to indicate requirement levels*", RFC 2119, S. Bradner, March 1997.
Available at: http://www.ietf.org/rfc/rfc2119.txt

[RFC3066]
"*Tags for the Identification of Languages*", RFC 3066, H. Alvestrand, January 2001.
Available at: http://www.ietf.org/rfc/rfc3066.txt

[SRGB]
"*A Standard Default Color Space for the Internet*", version 1.10, M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, 5 November 1996. Available at:
http://www.w3.org/Graphics/Color/sRGB

[UNICODE]
"*The Unicode Standard*", The Unicode Consortium. Version 3.1 of the Unicode Standard is composed of a book documenting version 3.0 and an online Unicode Standard Annex documenting the changes and additions that make up version 3.1.
"*The Unicode Standard -- Version 3.0*", The Unicode Consortium, Reading, Mass.: Addison-Wesley Developers Press, 2000. ISBN 0-201-61633-5 (see http://www.unicode.org/unicode/uni2book/u2.html for the online edition of the book).
"*Unicode Standard Annex #27 -- Unicode 3.1*", Mark Davis, Michael Everson, Asmus Freytag, John H. Jenkins et al. (see http://www.unicode.org/unicode/reports/tr27/).
For more information about Unicode, see http://www.unicode.org.

[URI]
"*Uniform Resource Identifiers (URI): Generic Syntax*", RFC 2396, T. Berners-Lee, R. Fielding, L. Masinter, August 1998.
Available at: http://www.ietf.org/rfc/rfc2396.txt. This RFC updates RFC 1738 [URL] [p.104] and [RFC1808] [p.103] .

[URL]
"*Uniform Resource Locators (URL)*", RFC 1738, T. Berners-Lee, L. Masinter, M. McCahill, December 1994.
Available at: http://www.ietf.org/rfc/rfc1738.txt

[XHTML1]
"*XHTML 1.0: The Extensible HyperText Markup Language*", W3C Recommendation, S. Pemberton *et al.*, 26 January 2000.
Available at: http://www.w3.org/TR/2000/REC-xhtml1-20000126

[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 Proposed Recommendation, H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, *eds.*, 30 March 2001.
Available at: http://www.w3.org/TR/2001/PR-xmlschema-1-20010330
See also "*XML Schema Part 2: Datatypes*", available at: http://www.w3.org/TR/2001/PR-xmlschema-2-20010330
See also "*XML XML Schema Part 0: Primer*", available at: http://www.w3.org/TR/2001/PR-xmlschema-0-20010330

D.2. Informative 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

[MathML]
Mathematical Markup Language (MathML) Version 2.0, D. Carlisle *et al.*, 21 February 2001.
Available at: http://www.w3.org/TR/2001/REC-MathML2-20010221

[SMIL]
SMIL 2.0: Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation, Jeff Ayars, *et al.*, 07 August 2001.
Available at: http://www.w3.org/TR/2001/REC-smil20-20010807

E. Acknowledgements

This appendix is *informative*.

The following people provided support and assistance:

- Henry Thompson
- C. M. Sperberg-McQueen
- The XML Schema Working Group
- Peter Stark
- Steven Pemberton
- Rick Jelliffe
- Josef Dietl
- Sebastian Schnitzenbaumer
- Malte Wedel
- Jack Herer