Overview

• Introduction
  • Classes of models
  • When to model
  • Overview of analysis and design

• Basic Tools
  • Schemas
  • Models

• Steps
  • Planning
  • Analysis
  • Design and modeling
  • Implementation and Review
Context

• Schema are replacing DTDs (We use the terms schema, but the terms are generally interchangeable)
• XML is replacing SGML and there are both minor and major differences
• Under XML, schema are used for a number of different functions and the design methodologies will be different for the different forms.
  • This document focuses on document content modeling (DCM)
• Even DCM schema can serve different purposes – e.g. reference, authoring, etc.
When is modeling needed

• XML is increasingly used for data interchange.
  • The “document” interchanged may simply be an XML wrapped DBMS record or table.
  • Data type modeling might be required in such a case, but content modeling is likely not needed.

• Before developing a schema, make sure that one that meets needs doesn’t already exist

• Document content modeling is called for when a model for a class of documents is required.
  • A DCM schema should apply to a range of documents – e.g. all of the policy statement in an organization, all the entries in a catalog, all classes of patient medical records
Building a Document Content Model

- Building a DCM schema is a form of analysis and design and requires five basic steps:
  - Requirements gathering
  - Analysis of data
  - Modeling of application
  - Implementation
  - Validation

- The model can be implemented in a standard waterfall or an iterative design
DAGs

- Under XML, modeling a document is very simple at the conceptual level. Documents are directed acyclic graphs (DAGs), meaning:
  - They have a root element
  - The root element may have children elements which in turn may have children
  - Elements may be defined as:
    - Having a sequences
    - Optional or required
    - Repeatable
    - A choice among alternatives
  - Elements may be further defined by attribute value pairs associated with the element
Syntax

• SGML, and originally XML, used a special syntax for modeling a document
• XML has now turned to a method of modeling a document that defines a document content model through an XML document.
• A DCM defined in this form is called a schema
• The basic elements defined for schema include:
  • element
  • attribute
  • complextype
  • simpleType
  • sequence
  • choice
  • group
  • all
An Simple Example

• The following example says that the element USAddress is a sequence that includes name, street, etc.

```
<xsd:element name="USAddress">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street" type="xsd:string"/>
    <xsd:element name="city" type="xsd:string"/>
    <xsd:element name="state" type="xsd:string"/>
    <xsd:element name="zip" type="xsd:decimal"/>
  </xsd:sequence>
</xsd:element>
```
An Example with an Attribute

The following example says that the element USAddress is a sequence that includes name, street, etc. and has the attribute country

```xml
<xsd:element name="USAddress">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street" type="xsd:string"/>
    <xsd:element name="city" type="xsd:string"/>
    <xsd:element name="state" type="xsd:string"/>
    <xsd:element name="zip" type="xsd:decimal"/>
  </xsd:sequence>
  <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
</xsd:element>
```
Requirements

- Identify the stakeholders in the process
  - Users of the schema
  - Individuals to be involved in decision making
- Articulate the goals of the project, for example
  - Document validation,
  - Author productivity,
  - Multiple delivery formats
- Scope of the project
  - Documents included
  - Time frame for use
  - People and systems impacted
- Budget and timeline
Analysis

- Collect samples of the document to be modeled
- Identify candidate elements and separate elements into content, structure and presentation
  - Generalize content elements to structure
  - Translate or discard presentation elements
- Identify and define the basic elements
  - Classify the elements into logical groups
  - Identify attributes of the elements
- Validate the components and the classification
Design the DTD

- Select the components that should be modeled
- Build top level element and attribute models
- Build middle level element and attribute models
- Build low level element and attribute models
- Populate the model where choices will be made
- Establish the linkages with the outside world
Tree Diagrams

- Tree diagrams can be used to represent XML documents
- Nodes
  - Rectangles are used for elements
    - No symbol indicates required and non-repeatable
    - + sign on vertical link indicates required and repeatable
    - * means optional and repeatable
    - ? means optional and non-repeatable
  - Ovals indicate content e.g. PCDATA
- Edges
  - horizontal bracket means all elements at level in order
  - diagonal lines specify choice from set
  - A circle with nodes means all elements in any order
Element/Attribute Decisions

• If data is to be accessible, it should be defined as an element
  • XSLT provides access to attribute data, but it is not necessarily the preferred way to do it.
• Attributes should be reserved for meta information – information about information.
• In general, when it is not clear whether something should be an attribute or a subelement, choose subelement
  • The rendering of a subelement can make it invisible.
Validation

• Validate the model -- technically and semantically
• Review the relationships between the developed DTD and other existing DTDs
• Test the DTD to determine if it meets the Goals of the project
  • Does it do what we set out to do
• Implement the model providing appropriate end user training