Document Analysis for DTD or Schema Development

Mulberry Technologies, Inc.
17 West Jefferson Street, Suite 207
Rockville, MD 20850
Phone: 301/315-9631
Fax: 301/315-8285
info@mulberrytech.com
http://www.mulberrytech.com

December 2001

Mulberry
Technologies, Inc.

Document Analysis for DTD or Schema Development

slide 1

Administrivia

- Class hours
- Break
- Introductions
  - Instructor
  - Class
- Questions are always welcome. We will
  - Answer most
  - Postpone a few until later in class
  - Take esoteric questions off-line

slide 2

Agenda

- Lecture
  - Introduction to XML Basics
  - Design for Retrieval
  - Document/Information Analysis
- Workshop
  - Requirements
  - Scope
  - Name/define information components
  - Identify relationships
  - Define Constraints
  - Enrich

Introduction

slide 3

A Few XML Basics

- Information is composed of components
  - Elements
  - Attributes
- Components can be related in predefined ways

```
message (a component) contains
          message header (another component)
          message body (yet another component)
```

slide 4

How XML Works

- <TAG>s inside the information:
  - Mark component start (<date>)
  - Mark component end (</date>)

```
<date>2001-03-25</date>
```

- Same information = same tag
- Different information = different tag

slide 5

A Component

- Is one continuous piece of information
- Has a beginning and an end
- For example:
  - Structural part of a document
  - Named piece of data content
  - Pointer to information

(There may also be additional information about a component)

slide 6

A Tagged Component

- Begins with a start tag
  - Names the element
  - May have attributes inside the start tag
- Ends with an end tag
- For example:

  `<tag attribute="value">`contents of the element, which may be very long`</tag>`

slide 7

XML-Tagged Data Stream

```
<bank-loan unique-id="D226-74-5619">
<account>3008070</account>
<interest-rate type="float">6.5</interest-rate>
<amount datatype="integer" curr="US">15000</amount>
<maturity date="08/22/99" time="1500" zone="EST">August 22, 1999
by the close of the United Bank business day</maturity>
<audit-comp>14862AC1286W5-110</audit-comp>
</bank-loan>
```

slide 8

XML-Tagged Document

<article>
<title>Introduction to XML***</title>***
<para><person emp.no = "BL432">Mr. Black***</person>*** has reviewed
the topic list you supplied and he believes the topics you suggested
may be covered in four seven-hour days. At your option these could be
either four contiguous days or two days in one week and two days the
following week. The basic we suggest is:
<outline><day>1</day>
<title>Introduction to the basic concepts of XML***</title>***
<para>This introduction will cover the basic
principles and syntax of XML; it will allow those students unfamiliar
with XML to learn the necessary concepts and vocabulary, and refresh
the subject for those students already familiar with XML.***</para>***
<para>Situations in which XML is and isn't appropriate
will be discussed,...***</para>*** ...</outline>...***</para></article>***

Why Define Tag Sets?

slide 9

What is Document/Information Analysis?

The process of identifying and defining
what tags to use to mark up a defined set of data
for specified purposes

slide 10

How XML *Might* Work

- Authors make up tags on the fly
- Each person makes up his own tags
- *Everything* is tagged
- XML is well-formed with start and end tags
- Everyone creates XML

slide 11

Creating XML That Works Versus Just Creating Tags

If the goal is information

- Retrieval
- Reuse
- Interchange

then you need to know

- What are the tags (in advance)
- What information is in which tag
- What's valid and what's an error

slide 12

The Usual Solution

- Groups agree on tag sets and tagging rules (models)
- Common groups include
  - A company
  - An industry
  - People with a common interest
  - Tool vendors

slide 13

An Information Model Can Say

- What the tags are
- What sort of information is inside each tag
- What are the relationships between the tags

slide 14

Models Express Rules

for example:

- **journal article**=*metadata*, followed by *article body*, followed by optional *back matter*
- **Purchase Order** =
  Order Header, followed by *List of Order Detail*, followed by optional *Order Summary*
- **paragraph** = data characters and may include any of the following: *Product Number*s, *Product Name*s, and/or *Trademark*s

slide 15

An Information Model and Accompanying Documentation Can

- Name the tag set and attributes
- Define how each element can/should be used
- Define containment and structures
- Document constraints (such as data types or generated text)
- Provide philosophy and naming conventions

slide 16

Why Make Rules?

- To support document creation (make only legal data)
  - Rules ensure documents conform to model
- To validate documents (that you or others make)
- To limit tag usage so that:
  - Style sheets can be created and exchanged
  - Behavior can be programmed
- As communication — to tell others what your tags and rules are
  - No need for same processor or application
- To automatically customize applications

slide 17

Rules Can Be Used in Sets

- Rules can be used in sets. Different but related rule sets can:
  - Provide conformance-testing at milestones in document lifecycle
  - Be more or less permissive

slide 18

Shared Models Allow (1)

- Precision searching across
  - Large document collections
  - Multiple document types
  - From diverse sources
- Repurposing
  - Author multiple similar documents
  - Cut and paste among documents

slide 19

Shared Models Allow (2)

- Leveraging other people's investment in software
  - Common browsers and other tools
  - Common WYSIWYG table processing
- Better information interchange
  - No need to translate between tag sets
  - Faster to set up new rules and systems

slide 20

How to Make Rules

- Machine-readable rules
  - DTD (Document Type Definition)
  - Schema
- Human-readable rules
  - Typesetting specifications
  - System functional specifications

slide 21

What Kinds of Rules

- Prescriptive versus descriptive
- Enforcing (validating) versus enabling (loose)
- Different models for different functions

slide 22

Prescriptive/Enforcing Models

- To create new information
- Use specific rules for content
  - Elements required
  - Element order enforced
- Modify older material to fit

slide 23

Descriptive/Enabling Models

- Describe existing material
- Use flexible rules for content
  - Many/most elements optional
  - Many different arrangements (order) permitted
- Enable documents that are variations on a theme

slide 24

Different Models for Different Purposes

- Enforcing model to build database or transaction
- Enforcing model to author book or article
- Enabling interchange models to exchange information
- Conversion model (to accommodate older documents)
- Output rules (information rearrangement or subset)
- Model-less browsing format

Design for Retrieval

slide 25

Search Effectiveness Problem

- When we search large bodies of information we:
  - Get too much stuff we don't want
  - Don't get all the stuff we do want (or don't know if we got all the stuff we wanted)
  - Have a hard time sorting the stuff we want from the rest

slide 26

This is Not a New Problem

- Measuring quality of information retrieval has been studied since the mid-1960s
- Measuring retrieval quality is very difficult
- The bigger and looser the collection, the harder to measure

slide 27

Definitions

- Swets defined Precision in 1963:
  - Relevance - the proportion of the retrieved stuff you want
  - Recall - the proportion of the relevant stuff is retrieved
  - Precision - the ratio of Relevance to Recall

slide 28

Precision

- Precision is not measurable in any real system
  - Cannot measure Relevance well
    (hard to measure how much of what is retrieved is of interest)
  - Cannot measure Recall at all
    (how do you know what you didn't find?)
- We can still aim to improve precision
- System users develop a "Feel" for Precision

slide 29

Use Markup to Improve Relevance and Recall

- Improve Relevance through Grammar
- Improve Recall through Vocabulary

slide 30

Grammar

- The way parts of the information are identified
- Roughly, the tags you use
- Design and control to improve Relevance

slide 31

Grammar Can Enable

- Content
  - Context Searching
  - Context Excluding
- Bibliographic Data and Navigational Tools

slide 32

Markup for Context Searching

- Markup should identify:
  - Key information (whatever size)
  - Major subdivisions of the information
  - Information-rich portions of information
- The stuff your users want to look for

slide 33

Examples of Markup that Supports Context Searching

Users in various domains might want to find a phrase if it occurs in one of these contexts:

- Conclusions Section
- Goals and Objectives
- Recommendations
- Long-term Implications
- Author
- Policies and Procedures
- Ingredients List
- Problem Description
- Prerequisites
- Typical Applications
- Question
- Purpose
- Answer
- Location

slide 34

Markup for Context Excluding

Markup should identify:

- Unimportant information
  (or information unimportant to some users)
- Secondary or supporting information
- Negated information
  (failed attempts, options not selected, what this info is NOT about)
- Large blocks that can be ignored

slide 35

Examples of Markup that Supports Context Excluding

- Background
- References
- Experimental Design
- Rejected Options
- Methodology
- Acknowledgments
- Introduction
- Veterinary Uses
- History
- Stage Directions
- Examples
- Geographic Names

slide 36

Markup to Support Navigation

- Metadata
- Linking
- Bibliographic information
- Digital Object Identifiers
- Webs, Nets, Maps, Indices ...
- URIs

slide 37

Relevance is Improved

- Find “black” in the description of a symptom instead of all occurrences of “black” in a medical database
- Find “shock” in that database except when it occurs in:
  - Names or People or Organizations
  - Documents about nervous or mental disorders
- Find books about Charles Darwin not references to Darwin Jones who lives on Charles Street

slide 38

Vocabulary

- The ways subjects are identified
- May be attributes, tags, or content
- Specialized to subject domain
- Designed and controlled to improve Recall

slide 39

Controlled Subject Vocabulary/Indexing

- Natural language is messy; the better the writer, the more variation in vocabulary
- Texts often fail to state the obvious, such as their subjects
- Subject indexing and cataloging were invented to provide subject access to large collections of material (like libraries)
- Structured thesauri help find more and less specific subject matter
- Indexes give access to medium-sized collections (like books)

slide 40

Control the Vocabulary

- Give a set of options instead of a fill-in
- Use the Grammar to encode limited choices
- Use a subject indexing vocabulary, apply it at finest grain affordable
  - Identify the thesaurus or indexing vocabulary you are using
  - If no public vocabulary available, make up key words

slide 41

Examples of Controlled Vocabulary

- Equipment type might have a value of
  - Telephone desk-sets
  - Cordless Telephones
  - Cellular Telephone
  - Telephone Switching Equipment
- Subject Codes may be from:
  - Dewey Decimal Classification system
  - Biological Abstracts Header List
  - General Accounting Office Thesaurus

slide 42

Subject Indexing: Expensive and Valuable

- Requires intellectual effort
- Automatic indexing better than none, but not as good as expert human indexing
- If you'd pay for a back-of-the-book index in print, you should pay for the same access to electronic information

slide 43

Recall is Improved

- Find shade-loving plants not shade-intolerant plants in a garden catalog
- Find references to "Miner's Cough" when looking for "Black Lung Disease"
- Find books by Dr. Seuss when looking for books by Theodor S. Geisel

What is Information Analysis?

What Functions Does Analysis Fill?

slide 44

Collects Data for Modeling

Use Analysis to Determine

- What's relevant in your data
- What's useful in your data
- What to identity in your data
- What's the framework/scaffolding supporting what you need

slide 45

What's Relevant in Your Data

- Key information
- Information-rich portions of information
- Major subdivisions of the information
- Content that has many purposes

slide 46

What's Useful in Your Data

- Use
  - Drive the handheld device
  - Deliver picking/packing information
- Reuse
  - Make a list of all the …
  - Collect all the …
- Retrieval/access
  - The users want to look for …
  - The DBA/librarian/data manager needs this information to organize the …

slide 47

Design a Framework/Scaffolding

- Basic infrastructure
  - Record structure for transaction-based data
  - Hierarchical structure for reference book
- Extra "containers" to support display
- Revision control/permissions/status
- Metadata *about* your data

slide 48

Establish Data Constraints To

- Make data usable
- Ensure received data will work in applications
- Remove interchange roadblocks

slide 49

Constraints to Ensure Useful Data

- Elements and attributes have “appropriate” values
  - Closed value set
  - Authority file or database lookup
  - Semantic constraints
- Express complex interdependencies
- Repeating structures are appropriately sized

slide 50

Constraints to Ensure Clean Data

- Improve the relevance of what you retrieve
- Context Searching
- Context Exclusion
- Bibliographic data and navigational tools

What You Do During Analysis

- Identify and name information components
  (for tagging)
- Define components
  (to identify them consistently)
- Describe component relationships
  (for use/verification)
- Describe component properties
- Describe constraints

The OLD Methods of Analysis

slide 52

Who Performs Information Analysis? (aka Document Analysis)

slide 53

Expert-based Analysis (boo hiss!)

- Client provides some documents or schema
- XML "experts" do the analysis
  - Interview users, authors, others
  - Name and define elements
  - Work in mysterious ways
  - Deliver a full-blown "DTD" (written in a strange, technical language)

slide 54

What's Wrong with This Picture?

- Samples aren't the universe
  (typical versus special, limited view)
- Only printed material or latest CD-ROM provided
- Only current material
  - Not historical and backlog
  - Don't consider future use and reuse
- XML experts don't know the data
  (logical but wrong, known bugs, and rare cases)

slide 55

Only Users Know

- Purpose and uses of information
  (who uses, why, how)
- What potential components are relevant
- How data created
- Component names/definitions
- The Real Rules (written and unwritten)
- Deep structure
- Current processing (database, formatting, etc.)

slide 56

Experts/Consultants Know

- XML modeling syntax (DTD/Schema/etc.)
- How XML is intended to work
- How similar problems were solved
- Common errors and pitfalls
- Implications of decisions
- Real world tools

Analysis the Modern Way

slide 57

User-Based Analysis

- Exploits user knowledge
- Uses group process to:
  - Extract information
  - Build consensus
  - Let users (NOT outsiders) resolve conflicts

Users construct solution, not just provide input

slide 58

Facilitated Analysis Workshop

- One-time, one-place decision making
  - Selected users
  - Trained, outside facilitator
  - Written information capture
- Analysis top-down and bottom-up
- Consensus decisions

slide 59

Who Participates in Analysis?

- Outsiders
  - Facilitator
  - Analyst/recorder
- Company and clients
  - Authors
  - Editors
  - Subject experts
  - Production staff
  - Systems staff
  - Marketers and designers
  - Observers

Document/Information Analysis Process

slide 60

Document/Information Analysis Process

- Start top down
- Determine element boundaries BEFORE looking at the contents
- Work element groups from the bottom up
- Identify information *about* each element
- Compare similar components for similar contents and metadata

slide 61

Analysis Step-by-Step

- Determine requirements
- Determine scope
- Find and define components
- Define relationships between components
- Identify constraints
- Enrich
  - Element collection
  - Attributes (about elements)

Step 1: Requirements

slide 62

Requirements and Scope are the Most Important

- Drive decisions for
  - Architecture
  - Analysis
  - Design
- Determine
  - Granularity
  - Relationships
  - Trade-offs between design/implementation conflicts

slide 63

Know WHERE you are going and WHY

slide 64

Define Requirements

- Application goals
- Application non-goals
- Organizational requirements
- Existing document standards
- Existing application standards

slide 65

Goals of the Application

- Why do you need/want XML?
- Who uses the information? How?
- What are the requirements for
  - Interchange?
  - Searching/access?
  - Restriction/security?
  - Data reuse?
- What do you want to do that you can't do now?

slide 66

Non-Goals of the Application

- What are the limits to what you are trying to do?
- What is NOT in this application, by design?
- State where you aren't going now
- State where you might go later

slide 67

What Output Products from This Information?

- WWW linked pages
- Enterprise-wide information base(s)
- Electronic books
- Hardcopy publications
- Search service databases
- CD-ROMs
- Alternative formats

slide 68

What do You Want to DO with the Information?

- Information/document interchange
- On-demand printing
- Database search and retrieval
- Electronic distribution and review
- Data reuse — “slice and dice” publications

slide 69

Organizational Requirements

- Who are the stakeholders?
- Who will benefit?
- Who holds the purse strings?
- Who stands to lose?
- Is there support from the top?
- Is there support from the bottom?

slide 70

Existing Production Standards

- Database schema(s)
- Corporate, agency, or project standards
- “House style” or writer's guidelines
- Rules for information production/appearance
- Typesetting/printing contracts
- Web design look and feel

slide 71

Existing XML/SGML Standards

- Business or process models
- Tag sets
- DTDs or schemas
- Industry standard
  - Your organization
  - Similar organization/vertical market
- Applicable public application standard

Step 2: Scope

slide 72

Scope

- Information universe
- Information types
- The limits of what you are describing
- The limits to what you are trying to do

slide 73

Information Universe and Types

- What is *produced currently* versus what *should be* produced
- Model-Ts versus marble sculptures
- How many types of
  - Databases/information bases
  - Documents
  - Transactions

  (and how many **must** you deal with)
- What data is NOT included

slide 74

Sidebar: The Fine Art of Gathering Samples

- Samples should include:
  - A wide range (common to weird)
  - Not just the easy-to-find
- Database schema or data dictionary
- Printed documents (past, present, and future)
- Electronic and other media
- Templates

Step 3: Name/Define Elements

Which Elements to Name

slide 75

Find and Name Elements

Begin with the outmost boundaries

- What are the "documents"? (usable or revisable units)
- What are the large structural/content pieces of the information?
- What are the smaller pieces inside the big parts?
- What floats loose in text with the data characters?

slide 76

Name What You Want to Use!

- Name in order to
  - Collect or identify
  - Reuse or discard
  - Search within (or exclude from a search)
- Name
  - What is important
  - What you want to use
  - Everything that looks different

slide 77

How Big is an Element?

As big as it needs to be:

- A multi-volume encyclopedia
- One article in that encyclopedia
- One word in an article
- Smaller than a word (tokens and phonemes)
- Non-print or non-display elements

slide 78

What You *Can* Tag
Versus
What You *Should* Tag

(True Versus Useful)

How Many Elements? (True)

```
Fred's Auto Parts
Turbo Building, Suite 207
131 Road Street
Elizabeth City, NC 33196-3541
```

1. Vendor Entry (the whole thing)
2. Vendor Name
3. Building
4. Suite Designator
5. Street Number
6. Street Name
7. City
8. State (abbrev)
9. Full State Name
10. Full Zip code
11. Zip: Five digit
12. Zip: Four digit

slide 80

How Many Elements? (Useful)

Requirements:

- Print mailing labels
- List supplier names
- Sort mail into zip code order
- Print address directory
- Record if supplier is partner

Five elements:

1. Vendor Entry (the whole thing) [Attribute: “partner”]
2. Supplier Name
3. Supplier Address (contains Address Lines)
4. Address Line
5. Zip Code

slide 81

Compromise on Utility

- There is no “ideal” markup
- If you optimize for X, you won't do Y
  (Subject retrieval for chemistry won't work for linguistic analysis)
- Markup costs money
- Markup that requires a subject matter expert costs more money

slide 82

How Big are *Your* Elements?

- How many useful components?
- How big are the components?
- Bottom Line: There is no right answer!

slide 83

Words of Advice

- DO NOT tag everything you can
- Tag only what you need
- Always determine edges before considering contents
- Work top-down (outside-in) first, then bottom up by groups

slide 84

Determine the Contents of the Element

What's Inside an Element?

- Other elements
- Data characters
- Data characters and other elements
- Nothing at all!

Types of Elements

slide 85

How to Find Elements

- Content — What kind of information is it?
- Structure — What is it?
- Description/Access — What do I know *about* this information
- Format — Does it look different?

slide 86

Structural Elements versus Content Elements

Structure (what is it?)

- Physical piece
- Logical piece

Content (what does it say?)

- Meaning
- Information class or content

slide 87

A Book: Structural View

```
Book
   Title
   Chapter      (may repeat)
      Title
      Section   (may repeat)
         Title
         Para   (may repeat)
         Section (may repeat)
   Appendix     (may repeat)
```

slide 88

Same Book: a Content View

```
Book
   Title
   Executive Summary
   Introduction
      Preface
      Background
   Methodology
   Results
   Conclusion
   Appendix     (may repeat)
```

slide 89

Elements that Describe

Metadata elements and Bibliographic elements

- Identifiers and database keys
- Who created, when
- Purpose, intent
- Description
- Ownership, copyright, publisher
- Usage: how-used, where-used
- Who uses: permissions, audience, security
- Source, history, author's notes

slide 90

Access and Finding Aid Elements

- Keywords, search terms
- Subject areas
- Index terms
- Information type or class
- Position in other hierarchies, ontologies

slide 91

Elements for Format/Display/Behavior

- Highlighting and Typographic emphasis
- Linking
- Calling programs to perform functions
- Print formatting information such as orientation, centering, column or page float

Help Stamp Out Tag Abuse

slide 92

Tag Abuse

- What it is
- Why people do it
- Problems it causes
- How to reduce it

slide 93

The What and Why of Tag Abuse

- Tag Abuse is
  - Identifying data as something other than what it is
  - Deliberate misuse of tags
- It's done to:
  - Make it "look" right
  - Save time or energy
  - Compensate for inadequate tag set

slide 94

Problems Caused by Tag Abuse

- Bad retrieval (false drops or lost data)
- Poor portability (looks right on browser but not in print or in a different browser)
- Misleading to user
- Reduced reusability

Design to Reduce Tag Abuse

- Provide ways to tag
  - Typographic emphasis (italic, color)
  - Common display types
- Expect unexpected content
  - Generic structures as well as named ones
    - Section
    - Title
    - Paragraph
    - List
- Escape hatches or loose places in the models
- Mechanism for tag set growth

Identify, Then Name, Then Define

slide 96

Name the Elements

- Elements commonly have two names
  - Tag name (short, used in data)
    (aka Element Type Name)
  - Expanded name (longer, used in help screens/documentation)
  - <PARA> versus Paragraph
- Some applications, users see tags only
- Some applications, users see long names only

Design assuming that end users will see tag names!

slide 97

Design Names for Human Use

- Familiar to users
- Short (therefore keyable) but readable
- Mnemonic (easy to remember)
- For example:
  - `<title>`, `<time-of-day>`, `<Table-of-Figures>`, `<death-time>`
- Not like this:
  - `<ti>`, `<tod>`, `<tofig>`, `<tod>`

slide 98

Context-dependent Names

- One element can be used in many contexts
  - Format depends on context
  - Specific meaning may also vary
- Context-specific Elements
  - Add to system complexity
  - Reduce reuse, cut-and-paste, editing ease
  - May make processing easier
  - May be necessary for unique processing

slide 99

Context-dependent Names (2)

- Title may appear inside
  - Book or Article
  - Section
  - Example
  - Figure
  - Table
- The alternative is to name tags:
  - Book-Title or Article-Title
  - Section-Title
  - Example-Title
  - Figure-Title
  - Table-Title

slide 100

Define the Elements

- Write a description (in English!)
- Give examples of the element in use
- Disambiguate similar elements
- Provide “remarks” on when or how to use

slide 101

What is a Good Definition?

- Describes only *one* element
- Is not so general it could apply to more than one
- Is expressed in your vocabulary (not XML-ese)
- Uses a standard definition, if available
- Points a novice to the correct component

slide 102

What is a Bad Definition?

- Is recursive
  (The blort number identifies the number of the blort)
- Just lists the contents
  (An alphabet contains the letters a, b, c,...)
- Explains only the exceptions, not the typical
- Fails to say “why”

Step 4: Define Information Relationships

slide 103

Component Relationships

Part of

Hierarchy, containment

Sequence

Elements follow one another

Occurrence

How many of each element

Similarities

Types — Components with similar properties or models

Groups — Collections of similar elements

Kinds of — A, B, and C are types of D

About

Additional info about an element

slide 104

Hierarchy Indicates Containment

- Elements are containers
- Large containers contain smaller ones, which in turn contain smaller ones
- A Transaction contains
  - A Transaction Number followed by
  - One or more Actions
- Each Action contains
  - A Part Number followed by
  - Quantity and Price...

slide 105

Sequence

- In order:
  This element is followed by that element
- Choice:
  This element or that element
- Choice including data:
  Data Characters, which may also include this element

slide 106

Sequence Examples

- In order

  error-number, then error-text, then error-interpretation

  city, then state, then zip code
- Choice

  consumer or producer or supplier

  red or blue or green or magenta

  data characters or platform or operating-system or command-name

slide 107

Occurrence

How many of each Element?

- Required
- Optional
- Repeatable
  - As many as you need
  - Minimum number (how many must there be)
  - Maximum number (how many might there be)

slide 108

Typing Elements

- Similarity of content mode
- Information classing (similar information)
- Similarity of function (similar in how used)
- Temporal or packaging similarity (used at same time)
- Kind-of similarity
- Element equivalency (element equivalent to another element)
- Similarity of usage (groups, see next slide)

Groups of Similar Elements (Usage Similarity)

- Things inside a Paragraph
  - Highlighting
  - Place names
  - People
  - Legal citations
  - Error messages
  - Unix commands
- Things at the same level as (instead of) a Paragraph
  - Figure
  - Table
  - Note
  - Sidebar
- Floating elements (anywhere in text)
  - Hyperlink
  - Footnote
  - Electronic review

slide 110

Define Constraints

Constraints on

- Context (where or how used)
- Content (what's inside)
- Occurrences (how many)

slide 111

Constraints on Context

- Associated with what
- In these contexts but not those
  (A paragraph can contain footnotes, but a paragraph inside a footnote can't.)

slide 112

Constraints on Content

- Data Type
  (integer, date, positive integer, decimal, duration, ...)
- Authority list
  - Open or closed list of values
  - Match authority database
- Default
  (simple default, dependent on context or other information)
- Semantic type
  (map, dealing with people, procedure, ...)

slide 113

Constraints on Occurrences

- Not more than (Maximum number of)
- Not fewer than (Minimum number of)
- Dependency occurrence
  - If there is a blivet, there must at least one a blort
  - Each blort must contain an even number of massins

slide 114

Determine Dependencies

Data may contain dependencies such as:

- If there is an element AA, there must be an element BB
- Elements XX and ZZ are mutually exclusive
- There can only be a UU if there has already, somewhere earlier, been a QQ
- If the value of the "type" attribute is "other" then you must give a value for the "new-type" attribute as well as create an Exception element

slide 115

Some Dependencies Can Be Modeled in DTDs

- The last element in a transaction must always be a Closure-Code
- Every Chapter must begin with a Title
- A Measurement has two parts: a Value and a Unit
- A Policy-Holder must have at least one valid Policy-Number
- Surname may come either before or after Given-Names

Some Additional Dependencies Can Be Modeled in Schemas

(for example, using W3C XML Schema)

- Exact number of occurrences (min, max)
- Data types
  - atomic (e.g., date, integer, `ID`)
  - list (e.g., `NMTOKENS`, `IDREFS`)
- Length (min, max)
- Extended types
  - same as something else, minus
  - same as something else, plus
- Uniqueness
  - must be unique
  - must match other unique in some way
- Must be exactly like another element except for some additional constraints

slide 117

Some Dependencies Can't Be Modeled *Currently* in W3C XML

- Attribute dependencies
  - Among two or more attributes
  - Between an attribute and an element
  - Requiring fixed order of attributes
- Many multi-way dependencies
- Complex dependencies (described using words like "except", "unless", and "only when all of the following are true")

slide 118

Record All Constraints

Even if not enforceable with a DTD or a Schema

- Can document for human enforcement
- May write special validation software
- May guide authoring tool customization

slide 119

Step 6: Enrich the Information Collection

- New elements
- Additional attributes
- Additional relationships
- Data types and validation
- Formatting or behavioral properties

slide 120

Elements/Attributes to Help Manage/Organize the Information

- Revision, change levels
- Security/privacy
- Approvals and authorizations
- Machine-readable values for dates
- Document routing or tracking
- Electronic review

slide 121

Connections/Finding Aid Elements

- Links within the document
- Links to outside sources
- Index terms
- Links to graphics or tables
- Glossary, abbreviation, acronym entries

slide 122

Attributes to Add Information About Each Occurrence of an Element

- Unique identifier
- Status
- Security level
- Make this element a link
- How an element should look (placement, style)
- Behavior expected from element

Elements have attributes; attributes do not have attributes

slide 123

Review Element Groups

- Look at where each group is used:
  - Is the group in that context too broad?
  - Is it too restricted?
  - What else might be part of the group?
- Review your elements that have only data content
  - Should they include groups?

slide 124

Formatting or Behavioral Properties

For each element

- Text generated on display
- Associated internal or external links
- Scope-controlling or inheritance to children
- Different format or behavior in different contexts or circumstances

Analysis Wrap-up

slide 125

How Can You Tell When You're Finished?

A place for everything, and everything in its place

- Identified all components to be:
  - Selected
  - Used (to make something happen)
  - Reused
  - Formatted
  - Searched for / Searched inside
- Satisfied project goals and requirements

slide 126

Real Ways to Tell You're Finished

(Testing the Model)

- Tag some samples
- Create simple sample behaviors
- Make a few web pages
- Print a few pages
- User acceptance testing
- Pilot project

slide 127

Potential Cost of No Analysis

- Worst case = complete failure
  (XML application does not work)
- Application
  - Is never finished
  - Limps — not all it should be
  - Does not meet unrealistic expectations and is perceived as a failure

slide 128

Can I Build an Application Without Information Analysis?

Of course you can, just like you can:

- Write a computer system without a functional specification
- Build a house without a blueprint

But how good will it be?

slide 129

Why User-participation Analysis is Better

- Faster
- Self-documenting
- Systems concerns addressed *during* development
- Multiple viewpoints considered
- Not as many surprises during implementation

slide 130

Better for Users, Too

- User distrust/hostility reduced
  - DTD is considered *"Our"* DTD not *"your"* DTD
  - Key users own the analysis (they did it)
  - Key users create structure
- Training time/cost reduced
  - Familiar vocabulary and names
  - Real examples
  - Key users know content
- Implementation and maintenance become easier

Now: An Analysis, Step-by-Step

- Determine requirements
- Determine scope
- Find and define components
- Define relationships between components
- Identify constraints
- Enrich
  - Element collection
  - Attributes (about elements)