DTD Reengineering: A Case Study

Kurt W. Conrad

President, The Sagebrush Group

conrad@SagebrushGroup.com

XML 2002

Baltimore

December 10

What is DTD Reengineering?

- No layoffs
- Not routine maintenance or modification
- Fresh application of engineering principles
  - Even if never “engineered” in the first place

Why Reengineer DTDs?

- Like organizations, structures ca:
  - Become bloated and too complex
  - Loose relevance
  - Reflect a myriad of individual decisions
  - Rational has been lost or forgotten
  - No longer make sense
  - Lack cohesiveness
  - Don’t reflect an integrating philosophy
  - Inefficient

The Client

- I can’t tell you anything about them

The Data

- I can’t tell you anything about it
- Other than it had paragraphs

Business Context

- Been using and enhancing DTD for a number of years
- Passed among numerous developers
- Will be converting a large amount of content in near future
  - “Now’s a good time”
- Very knowledgeable, but limited bandwidth

Timeline

- 08.31: Initial SOW
  - Phase 1: Don’t get my new shoes dirty
- 09.09: Analysis Report #1
  - Phase 2: In the muck
- 09.20: Analysis Report #2
- 10.03: Start of active coding
- 10.30: Turnover

08.31: Initial SOW

- Level-of-effort: 8 days (64 hours)
  - 1/3 analysis
  - 1/3 reporting
  - 1/3 changes to DTDs
- Work scope focused on known problem areas

Goals

- Want simplified DTD
  - Whatever makes sense
  - Focusing on specific structures (e.g., para)
- Rationalize metadata (Dublin Core)
- Research a number of standard DTDs for possible inclusion/alignment
  - Topic maps
  - AAP
  - Parts of TEI
  - A couple of others
  - Every element subject to review
- Minimize impact on customer resources

Inputs

- 2 base DTDs
- ~25 supporting modules
- MS Word document
- Various hard-copy samples
- 3 tagged SGML samples

Initial Analysis Approach

- Created .pdf from Word doc
  - Inserted notes to record observations and questions
- Intent was to use .pdf as a communication vehicle
- Wasn't realized
  - Customer couldn't differentiate questions from recommendations
  - Too much stuff to go through (249 pps)
  - Document was incomplete
  - Less a transformation than a clean slate project
- Did produce a set of issues that seeded the dialogue

09.09: Analysis Report #1

- Hours to Date: 15
- Accomplishments
- Next Steps
- Findings / Potential Issues
- List of element-specific issues
- List of specific questions

Design Goals (Optimizations)

- Primary tension
  - Desire for 3-5 years of stability
  - Expectation of new archival and electronic transfer in 6-9 months
- Disambiguate names
  - Especially parameter entity names
- Focus on design of structure, not just elements
- DTD not used for authoring
  - Conversion by conversion houses
  - Population of system
  - If content upgraded, then source (mostly Word, some SGML) sent to conversion house

Initial Strategy

- Isolate the specific patterns and design approaches which drove complexity
- Come up with alternative set of design principles or a set of solutions, each targeted at a class of problem

Tools and Methods

- Intranets.com
- DTD Chart
- Live DTD
- Batch / XSLT / AWK
  - Develop a general-purpose, line-oriented state machine that supported a range of pretty-printers and DTD reporting tools

Intranets.com

- Used a bit
- No real advantages over email
- Kept getting email messages from an obsolete schedule

DTD Chart

- Draws structure diagrams from DTDs
- Found at www.intsysr.com/dtdchart.htm
- Not nearly as useful as Live DTD
  - Can’t print or save image in shareware version
  - Didn’t really work right
  - Didn’t try real hard to get it to work

Live DTD

- Perl scripts that creates a set of HTML pages from a DTD
- www.sagehill.net/livedtd/
- Excellent tool
- Also served as the basis for various reports and quick references
- Had a couple of problems with Perl Unicode

How Live DTD Was Used

- Rapid navigation through declarations and references
- Reformatted output as quick references
  - Element Usage Table
  - Entity Usage Table
- Served as the basis for expanded reports, as necessary

Element Reference Report

bibref element seen in:

```
   com
  (#PCDATA|loc|bibref)*
   %ref.class;
      bibref
      specref
      termref
      titleref
      xspecref
      xtermref
      %local.ref.class;
         ""
```

Analysis

good model

a bit messy

Expanded Content Model

```
code
    %tech.pcd.mix
        #PCDATA
        %loc.class;
            loc
            %local.loc.class;
                ""
        %ednote.class;
            ednote
            %local.ednote.class;
                ""
```

Usage Reporting Tools

- FullUse.bat
  - Launches UseRpt.bat for individual XML documents and merges results
- UseRpt.bat
  - Generates report from an individual XML file using Usage.xsl and Usage.awk
- Usage.xsl
  - Outputs name of element, parent, each sibling, each child (elements and text nodes)
- Usage.awk
  - Processes XML file produced by Usage.xsl to calculate totals

Element Occurrence Report

Document: I can’t tell you

para Elements	2328

Parent Elements	
aaaaa	64
bbbbbb	256
ccccccccccc	1
ddddddd	3
eeeeee	247
fffffff	50
ggggg	347
hhhh	147
iiiiiii	1213
jjjjjjjjj	4

Child Elements	
aaaaa	19
kkkkkkkk	10
llllll	1661
mmmm	636
nnnnnnnnn	1
oooo	1
ppp	820
qqq	1
rrrrr.rrr	2
ssss	518
	3669

Text Nodes	4273

09.20: Analysis Report #2

- Hours to date: 41.25
- Accomplishments
- Issues
- Status
- Decision Points
- Findings
- Recommendations

The Bottom Line

- “I have not uncovered a pattern which would facilitate an algorithmic approach to simplification” (i.e., no silver bullet)
- Reduction of complexity became the dominant design target
- Set new baselines for schedule, level of effort, and priorities
  - Regular status and review meetings
  - Client work items
- Armed with specific issues, the client expanded the budget

10.02: Elements Categorized

- Field
  - Purpose (Semantic, Format, Metadata)
  - Coding
    - Empty
    - Text-only
    - .model
    - .mix (Formatting, Linking, Other)
- Section
  - Purpose (Low-level, Mid-level, High-level, Metadata, Semantic)
- Structure
  - Purpose (Low-level, Mid-level, Format, Metadata, Semantic)

10.04: Document DTD

- Based on Mulberry DTD commenting approach
- Expanded to document
  - Current Declarations
  - Changes and Rationale
  - Original Declarations
- Found it necessary to add the commentary earlier than expected to support analysis

DTD Commenting Example

```
<!-- ============================================================= -->
<!--                  ROOT ELEMENT - XXXXXXX STRUCTURES            -->
<!-- ============================================================= -->

<!--                  XXXXXXX                                      -->
<!--                  This is the top-level document element
                                                                   -->
<!--                * Current Declarations                         -->
<!ELEMENT xxxxxxx       (title.group, xxxxxxx.FM, xxxxxxx.body, xxxxxxx.RM?)  >
<!ATTLIST xxxxxxx
      %required.id.attribute;
      yyyyyyy.number    CDATA                        #REQUIRED
      zzzzzzz           CDATA                        #REQUIRED
      wwww              CDATA                        #REQUIRED     >

<!--                * Changes & Rationale                          -->
<!--                  - Shift to normalized FM, Body, RM structure
                      - Shift from %titles; to title.group to create
                        a single structure for all titles
                                                                   -->
<!--                * Original Declarations                        -->
<!--
*  <!ELEMENT xxxxxxx       (%titles;, xxxxxxx.FM*,
*                          (part+ | %local.body.unit;)*, xxxxxxx.RM*)     >
*  <!ATTLIST xxxxxxx
*        %common.attrib.reqd;
*        WWWW              CDATA                        #REQUIRED
*        xxxxx             CDATA                        #REQUIRED
*        zzzzzzz           CDATA                        #REQUIRED     >
****                                                               -->
```

DTD Normalization

- Normaliz.bat (control script)
- DTDPP.awk
  - Simple pretty-printer
  - Really only lined up MDCs (“-->”)
- TOC.awk
  - Generated two-level TOC mapped to line numbers
  - Resulting TOC pasted in manually
- Splitter.awk
  - Removed extra revision tracking commentary, leaving declarations and headers

10.04 -10.27: Regular Telecons and Draft DTDs

- Emphasized
  - Restructuring
  - Capturing
    - Design decisions
    - Supporting rationale
- Issues referenced specific line numbers
- First validation on 10.17

10.25 - 10.27: Analysis and Normalization of Attributes

- Attlist.bat (control script)
- Attlist.awk
  - Output each attribute declaration and parent element
- Addln.awk
  - Added line number to end of each line to preserve original order
  - Resulting list was sorted by attribute name to see
- Analyzed for
  - Distribution and variations in names
  - Naming conventions

Attribute Normalization Method

- Added new attribute names to beginning of each line
- Used awk script to update more common portions of attribute names and entity references
- Made remaining changes by hand
- Re-ran reports, as necessary, to identify other attribute naming anomalies

10.29: Code Freeze & Turnover

- Working copies of the core files
  - DTD
  - DTD modules
  - XML stub file (declarations only)
- DTD files w/ version number in filename
- “Condensed Reports”
  - DTD files with only active declarations
- Referenced character entity files

11.27: Closeout

- Complete turnover package
- .zip with working directories
- Some files and directories renamed for clarity
- Final hours
  - Initial analysis (09.04 - 10.02): 57.25 hours
  - Design and coding (10.03 - 10.30): 201.75 hrs
  - Final turnover prep (11.01 - 11.27): 8.5 hrs

Results

- Little attention paid to external standards
  - Sounds Good, Maybe Later
- Initial goals of minimizing changes scrapped
  - Used parameter entities to document design decisions
  - Liberal inline documentation sufficient for turnover

Conclusions

- Figure out whether more a transformation of current DTD or Clean slate
  - How much do you really need to understand the current structure?
- Figure out how to start the systematic discussion early
  - Independent analysis doesn't provide enough traction
- Don't be afraid to ask the same question twice (or three times, or four times)
  - If answer doesn't stick (become internalized) there's probably an issue
  - Thinking changes over time

DTD Reengineering: A Case Study

Kurt W. Conrad

President, The Sagebrush Group

conrad@SagebrushGroup.com

XML 2002

Baltimore

December 10

This presentation can be found at:

sagebrushgroup.com/new/archives/DTDReengineering.pdf