




SoftQuad


Introduction



SoftQuad

Contents

- XML in Context
- Examples:
 - Application Linkage
 - Catalogs
 - Technical Manuals
- XML Labels
- Conversion
- Native XML



SoftQuad


Who am I?

- Director of Product Technology, SoftQuad Software Inc.
- Co-Designer XMetaL
- Project manager, conversion projects
- DTD designer
- DOM WG Chair



SoftQuad


XML in Context



SoftQuad

Different Uses

- Transfer syntax
- Everything else!



In Other Words...

- Potential automation
 - Humans seldom involved, except for debugging
- Content is king
 - Human intelligence is key



Example: Application Linkage



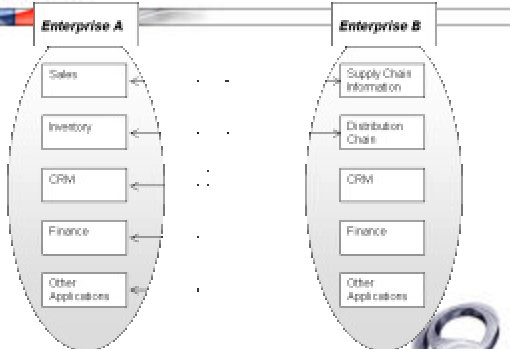
XML as Plumbing

- Transfer syntax between applications
- Serialize objects into XML
- Depends on applications
- Need to handshake



Example: Database

- Extract data as XML in one schema
- Transform using XSLT to some other schema
- Import to new database or application



B2B Integration

- Nobody sees the XML
- Set up schemas, rest happens automatically
- Some day, maybe automatic handshaking
- View XML for debugging



XML for Content



Why use XML?

- Content re-use
- Automation of processing after creation
- Ultimately viewed by human
- Non-proprietary data format



Some Final Formats

- HTML
- Printed
- CD-ROM
- XML, but another schema



Creation Methods

- Depend on
- What sort of re-use or processing anticipated
 - Current sources of data
 - Content management
 - Workflow



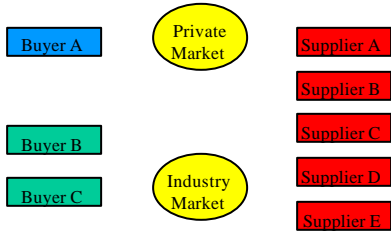
Example: Catalogs



Different Marketplaces

- Suppliers supply more than one marketplace
- Some are private (one buyer)
- Others are industry (many buyers)
- Some are public (many buyers)





Catalogs are Complicated

- Price depends on buyer
- Price depends on quantity
- Buyers will have more than one address
 - Shipping address
 - Invoice address
- Branding information changes with buyer



Marketplaces Aggregate

- Aggregate from many suppliers for one buyer
- Aggregate from many suppliers for many buyers
- Keep buyer information separate
- Keep supplier information separate
- Conditions apply
 - Some buyers won't buy from some suppliers
 - Some suppliers won't sell to some buyers



XML Helps

- Consistent format
- Labels all the information
- Can send just updates
- Merging relatively simple
- Even in merged catalog, can keep information separate



Example: Technical Manual



Cars Vary

- Basic design the same
- Variations for model
- Variations for year
- Variations for options



Documentation Must Match

- Repair steps depend on symptoms
- Repair steps depend on real car
- Link to parts systems



Complex Systems

- Design the way information fits together
- Design how information flows
- Include information for the future uses
- Include information for archival



XML as Labels



What is XML?

- (Almost) self-describing data
- Robust error-handling
- Internationalized
- Any schema allowed



Labels Have Meaning

Labels say something about the information

- What it is
- How to style it
- How to process it



Labels Need Intelligence

Why is this in italics?
(Quote, name of book, emphasis?)
Is this a part number, or date?
Date, or height of mountain?

Humans know, computers don't



Creating Useful XML



First Questions

What sort of XML?
How much intelligence is needed in the data?
Is the process automated?
Is the data tabular?
Is the data irregular?



Programmatic Creation



Generating XML

- Code creates XML
- Often created as input to another program
- Used as interface between programs
- Enables modular system architecture
- Not quite as easy as it sounds



Conversion



Definitions

- Legacy data
— What you already have
- Big-bang
— Convert it all at once
- Gradual
— Convert as you need it
- Built-in
— Continue to author in legacy system
— Convert after authoring



Conversion Methods

- Scanning
- Keyboarding
- Programmatic



Scanning

- From paper, microfilm etc
- Difficulties in distinguishing marks from characters
- Rule of thumb: 4 errors per page
- Spell-checkers used to catch errors
- OK for words
- No structure in legacy data
- Data goes to keyboarding or conversion



Keyboarding

- Usually starts with scanning
- People add structure, correct (some) errors
- Usually not subject experts
- Costs \$0.50 - \$1.50 per 1000 characters
- Cost depends on structure



When Keyboarding?

- Printed matter
- Formats vary
- Few documents
- Turnaround time not important



Programmatic Conversion

- Data converted from input to XML
- Quality of input vital
- Are there clues to the structure?
- Relative richness of source and target formats



Batch Demo

- Batch conversion from Word
 - Use Word macros to add metadata
 - Tag headlines etc with what they are
 - Generate HTML with lots of info
 - Clean up the HTML
 - Back-end converts to XML
 - Can stop on errors



When Batch?

- Users don't need to add much metadata
- Source formats are consistent
- Lots of source documents
- Documents updated and republished frequently
- Source relatively unstructured
- Target relatively unstructured



Interactive Demo

- Interactive conversion from Word
- Use Word API and XMetaL API
 - Get data from Word
 - Analyze
 - Put into XMetaL
 - Stop on errors



When Interactive?

- Target format is richer than source
- Source formats vary
- Teaches authors about formats
- Continue in XML or Word
- Documents not republished frequently



Conversion Points

- Use experienced consultant or well-known company
- Target easy intermediate format
 - Native XML or programmatic conversion to add the rest
 - Short element names saves money!
- Determine level of accuracy you need
- Decide on parameters up front
 - Validity
 - Schema
 - Other instructions



Native XML



When Native XML?

- Target format is richly structured
- Target format is variable
 - Different target formats for different documents
 - One author, many target formats
- Author wants to "touch up" content till last minute
- Need fast turnaround time
- Authors tend to spend time on formatting
- Add metadata when authoring information
- Future -proofing



XML Authoring Tools

- Browser form interface
 - Easy to use
- Forms-like view
 - Easy to use
- Word add-on
 - Sounds better than it is
- Tree-like view
 - Programmers like this
- Word-like view
 - Word users like this



Browser Form

- Simple XML only
- Rule of thumb: if it's OK in Excel, it's OK for a browser form
- Need a new form for each schema
- Need a new form for each schema change
- Doesn't work if author offline



Word add-on

- Difficult to write
- Authors tend to tweak documents
- Every tweak breaks the internal conversion
- Word only has two hierarchies
- Adding hierarchies makes it less like Word



General XML Authoring Tool

- Works for any schema
- Someone has to configure for each schema
- Someone needs to understand enough XML
- Appeals to programmers
- Appeals to XML-savvy authors
- Doesn't appeal to domain experts without XML knowledge
- Can be any style of view



Specialized XML Authoring Tool

- Works for one or two schemas
- Authors needn't know they're using XML
- Hard to do this with tree-like view
- Can be based on general tool



Demo: XML Catalog Creation

- MarketAgility Express
- Based on general XML authoring tool
 - Configured for one DTD
 - Complexity taken out
 - Author doesn't know it's XML
 - Forms view



Questions?

lauren@softquad.com
<http://www.softquad.com>
<http://www.laurenwood.org/talks>

