Handling Relationships as Links in Native XML Databases — Concepts, Applications, Implementation —

Anne Brüggemann-Klein
Lorenz Singer
Technische Universität München

Context: Engineering Web document systems

- Goals
  - use XML technology
  - use off-the-shelf XML processing components
  - strive for proven techniques: principles, patterns, procedures
    - to reduce complexity
    - to ensure sustainability (change)
- Case studies (lab courses on XML technology)
  - XBlog: XML-based blog system
  - XTunes: Management of metadata and recordings of classical music [focus on integration and linking of live data]

Context: Engineering Web document systems

- Extreme 2007: From domain model (UML class diagram) to XML Schema — Schema development process
  [demonstrated with blog system XBlog]
- Balisage 2008: Handling general relationships between domain entities that are managed as XML documents in an XML database system ➢ MTh Lorenz Singer
- Work in progress: Automatic translation of UML domain model into XML Schema via XMI ➢ MTh Dennis Pagano
  [automate schema development process and extend it to relationships]

Data: decomposed into related pieces

Deal with data in the classical sense and with documents

- natural decomposition of data that represent interrelated domain entities
  - results from data modelling (ER diagrams, UML class diagrams)
- further decomposition into interrelated pieces
  - results from normalization to avoid redundancy of data
    [also for XML documents (Arenas & Libkin 2002)]
- kinds of relationships
  - types: inheritance (is-a), composition (part-of), other
  - arity: binary, multinary
  - cardinality: one-to-many, many-to-many
  - typed, annotated with data

Methods of relating data: Databases

- Data in relational databases
  - organized into tables as records, identified by keys
  - implicitly related through separate relationship tables via foreign keys
- Applications
  - system: support of data consistency
  - queries: mostly indirect use (in joins)
- Evaluation
  - requires organization to interpret foreign keys: database schema
    - problems in data integration

Methods of relating data: Databases

- Alternative: data in object-oriented databases
  - organized into objects that have IDs
  - explicitly related through references (pointers)
- Application
  - direct use of references in query language
- Evaluation
  - database scope of references
  - data integration problem not solved

Methods of relating data: The Web

- HTML documents on the Web
  - related via simple hyperlinks
    - universal address scheme for targets
  - "anything may link to anything"
    rigorous repudiation of GOD (Grand Organizing Directorate) structures — the factor to make the Web superior to information systems that use categories for organization (Newsgroups, Gopher)
- Application
  - direct, user-driven use of hyperlinks in navigation
- Evaluation
  - global scope of hyperlinks
  - one building block for data integration

Methods of relating data: The Web

- Alternative: XML documents on the Web
  - related via links á la XLink
- What's in a link? ➢ Some XLink terminology
  - link: an explicit relationship between (portions of) ressources
    - supports multinary relationships
    - uses universal address scheme for linked ressources
    - carries metadata (including type)
    - can be defined separately from linked ressources (linkbase)
  - hyperlink: a link that is intended primarily for presentation to a human user, for navigation
  - traversal: using or following a link for any purpose [always involves a pair of ressources]
  - arc: information about how to traverse a pair of ressources

System of related XML data: Hyperdata system

- Evaluation: Create hyperdata system by putting interrelated XML documents into a database
  - native XML database
  - populated with interlinked XML documents
  - ➢ links gainfully used for storing, querying and processing the XML data

System of related XML data: Hyperdata system

- Applications
  - direct use of links to traverse documents in XML database with query language (XQuery)
  - use links when normalizing XML documents
  - maintain referential integrity by adding actions ("on delete restrict", "on update cascade") to links as metadata (roles)
  - define views as links, also views on top of views
  - define fine-graned access control (data access via link views that require minimal security level when traversed)
  - "link-driven applications": system behaviour when traversing a link depends on metadata (type) of link

Choice of link technologies for hyperdata systems

- Requirements
  - linking of ressources across system boundaries
  - linking vocabulary independent of applications
  - use of XML namespaces
- Candidates: XLink and RDF
  - XLink
    - explicit description of links
    - vocabulary of attributes, easy to integrate into data
    - complete vocabulary in the XLink namespace
    - mapping to RDF
    - no conflicts with other uses anticipated
  - RDF
    - interpretation of link as one option
    - vocabulary of elements, more difficult to integrate into data
    - predicates not in the RDF namespace
    - mapping to XLink ?
    - conflicts with other uses anticipated

Addressing scheme for hyperdata systems

- Organization of data in native XML databases
  - hierarchy of named collections
  - end-points: named XML documents
- Addressing via database URLs
  - «db prot»://«server»:«port»/«db name»/«collection path»/ «doc name»#«xpointer-expression»
    xindice://host.domain.com/db/coll1/subcoll2/doc3.xml#xpointer(«»)

HyQuery

- Integrating XLink processing into XQuery:
  XLink processor HyQuery as an XQuery function library
  - turns XML database such as eXist into a hyperdata system
- Functions for simple links
  - follow ($links)
  - followByRole ($context, $role), followByArcrole ($context, $arcrole)
  - follow-embed ($links), follow-embed ($links, $context)
  - follow-replace ($links), follow-replace ($links, $context)
- Functions for extended links
  - ext-getArcroles ($link), ext-followByArcRole ($link, $arcrole)
  - ext-followAllArcs ($link), ext-followAllArcs ($link, $linktype)
  - ext-followArcs ($link, $arcs), ext-followArcs ($link, $arcs, $linktype)
  - ext-getArcType ($link, $arc), mergeLinkbase ($base, $nodes)

XTunes as a hyperdata system

- XTunes (lab course project)
  - management of metadata and recordings of classical music
  - focus on integration and linking of live data
    [showcase turntable character of XML]
- Relationships in the domain model
- Data integration
  - internal representative for any entity that is known to XTunes
    whether actually imported or not (internal IDs, based on UUIDs)
  - mapping table maps internal representative to external sources (URI)
  - import of external sources, including internal interlinking
  - merging of internal representatives

XTunes as a hyperdata system

- XTunes schema (XML Schema)
  - relationships as typed simple links (XLink)
- Implementation platform
  - Apache XML publishing framework Cocoon
  - XQuery-enables native XML database eXist as Cocoon block
  - HyQuery as XQuery module in eXist

Conclusion

- Relationships between XML documents in XML database
  - concepts
  - applications
  - implementation
- A perspective for (currently under-used) XLink

Future work

- Consolidate this work as teaching materials and reference implementation ➢ student projekt Tamer Demirel
- Automatic translation of UML domain model into XML Schema via XMI ➢ MTh Dennis Pagano
  [automate schema development process and extend it to relationships]
- Solve last year's metadata / inheritance problem with linking
  [incorporate into schema development process]
- Validate principles, patterns, procedures
  with other case studies (suggestions?)
- Embed into development process for document systems
  - ➢ PhD work Thomas Schöpf: meta-model approach