Developing XML-based User Assistance at IBM and Lotus

Don Day

IBM Corporation

John Hunt

Lotus Development Corporation

Developing XML-based User Assistance at IBM and Lotus

- Why an XML Markup Language for User Assistance?
- Topics, content, and structure
- Linking and navigation
- Multiple uses and presentations
- **Special Bonus:**
  XML-based UA at Tivoli Systems
  - Linda Meyer and Chris Rothemich

Why an XML Markup Language for User Assistance

- It is the emerging standard for deploying content, both across the internet and within software applications
- It is part of the W3C family of Web-oriented technologies/standards
- It provides ways to address cross-browser/cross-platform issues
- It lets us easily exchange content with third-party developers

Useful traits of XML for User Assistance

- Intent-based markup (element names can describe content meaningfully)
- Broad tools support
- Open resource management based on URIs (principle of XLink)
- Alternative to compound documents for access and reuse of parts
- Style sheet driven transforms and formatting

Using XML: Functional ideals

- Rich in intent (semantically significant element names)
- Rich in typography (with style sheets)
- Able to be used either compiled or not
- Able to be reused in multiple delivery methods
- Lots of options for things like ToC, Index, Search

Using XML: Practical ideals

- Topic oriented
- Web oriented
- Few vendor dependencies
- Semantically rich
- Can grow to meet new needs

We took these as requirements for a next-generation UA architecture... in XML!

Our design approach

- Identify topics by information type, using a 'top-down' analysis
- Identify ways in which topics may be "repurposed" (chapters for books, etc.)
- Follow W3C standards for processing
- Extension mechanisms should be tools-friendly
- Do not rule out cutting edge ideas

Result: the Darwin Information Typing Architecture

- **Darwin** - it uses principles of differentiation and inheritance
- **Information Typing** - it capitalizes on semantics of topics (concept, task, reference) and of content (messages, typed phrases, semantic tables)
- **Architecture** - it provides vertical headroom (new applications) and edgewise extension (derivation to new types) for information

New terminology

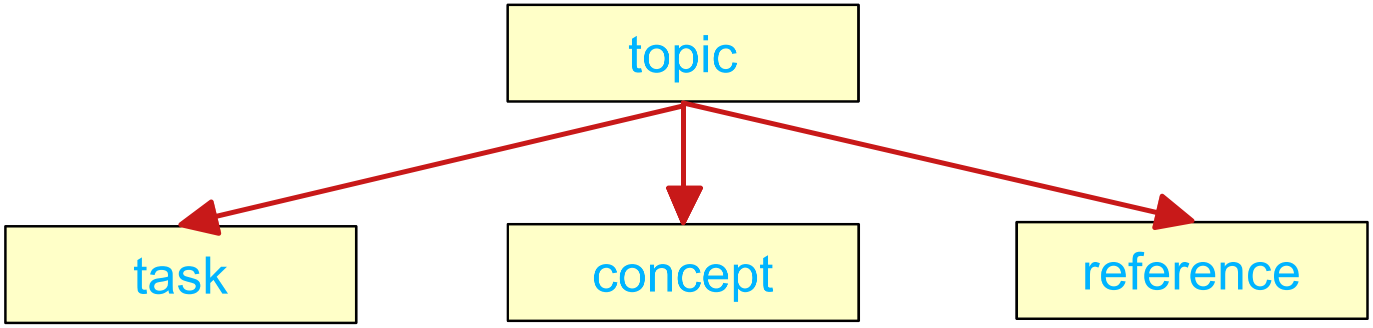
- DITAbase - the base DTD that contains the main topic definitions:
  - topic (generic, broadest)
  - concept
  - reference
  - task
- Specialization - defining a new authoring DTD using specific naming and constraints on prior content models

Topics and Information Types

- A **topic** is the core information unit in the Darwin Information Typing Architecture (DITA)
  - "A unit of information that is complete enough to describe a single task, concept, or reference item."
- An **information type** defines the role of a topic

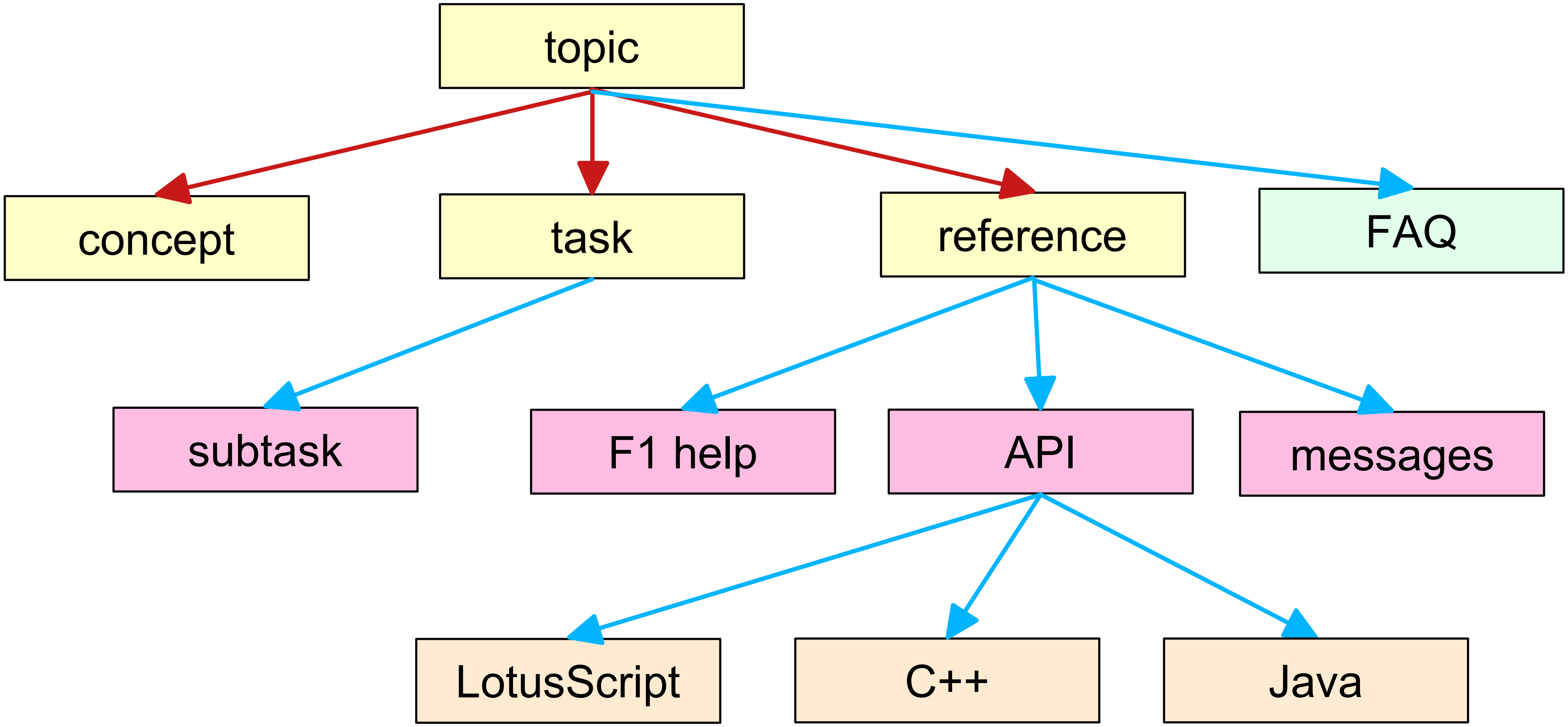
Four Basic Information Types

- Topic, Concept, Task, Reference



Extending the basic types

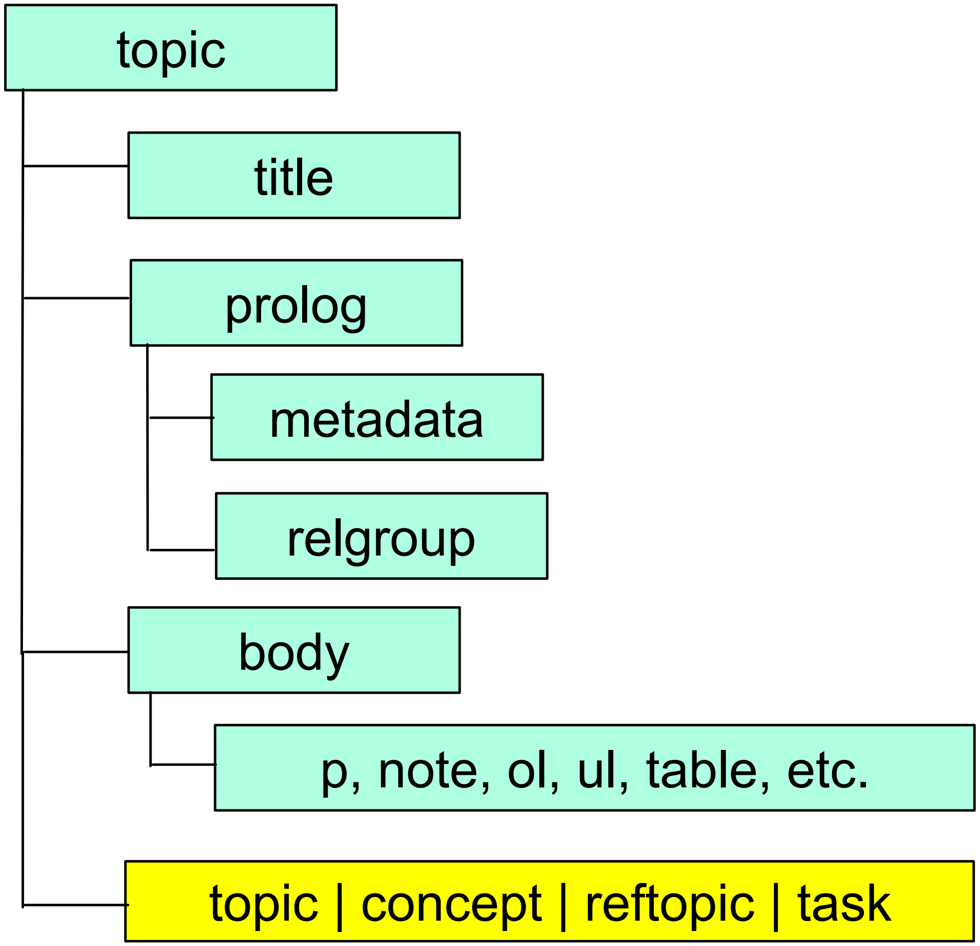
- Add new information types as specializations of one of the basic types



Topic types in Lotus Notes and Domino "End-user" Help

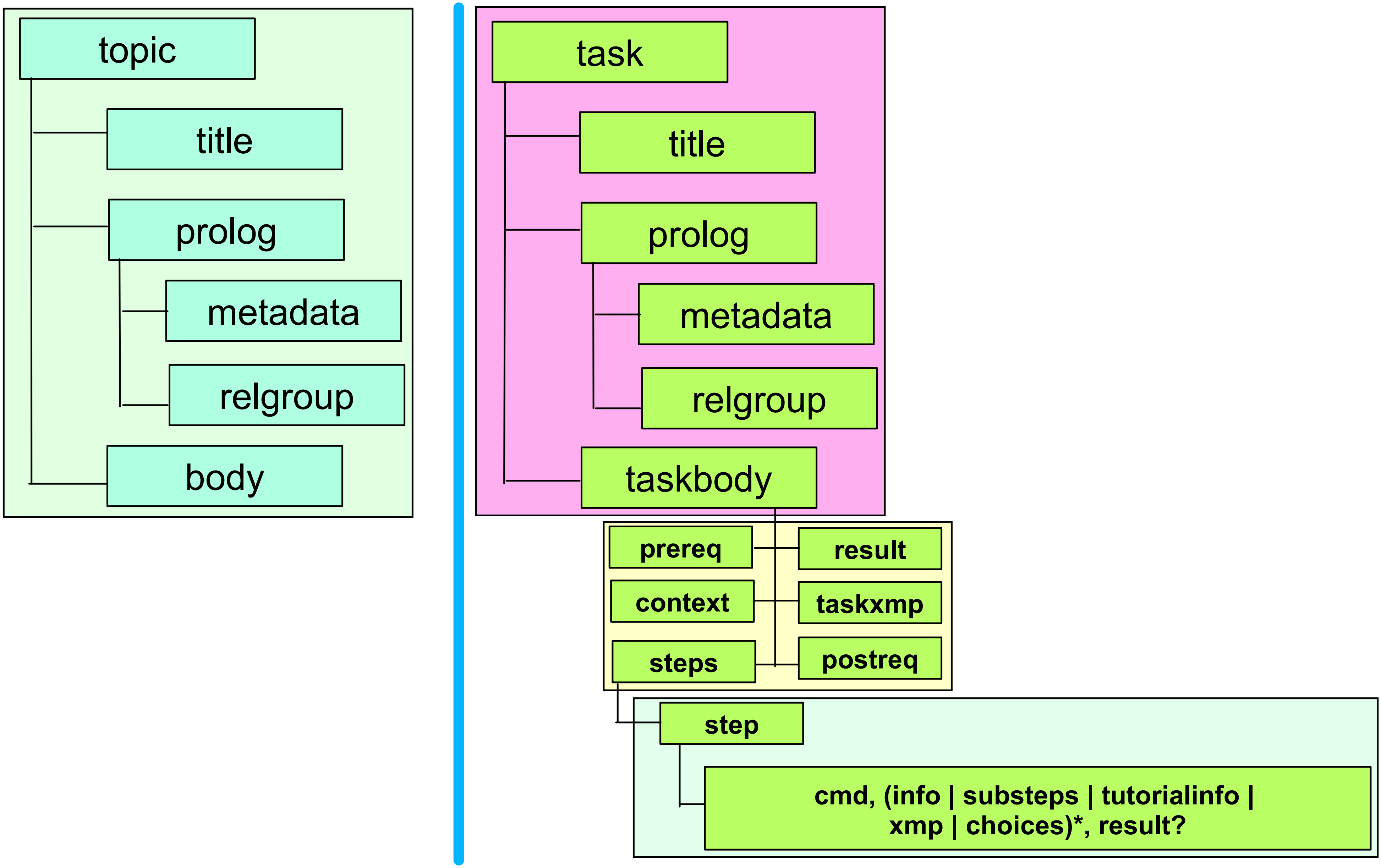
- **Assumption: All topics are either tasks or concepts**
- **Task topics**
  - Any topic with a Numbered list that's not in a table
- **Concept topics**
  - All other topics
- **Loose ends**
  - Troubleshooting, FAQs, Glossary

Process: First derive "generic" topics



- topic
  - title
  - prolog
    - metadata
    - relgroup
  - body
    - p, note, ol, ul, table, etc.
  - topic | concept | reftopic | task

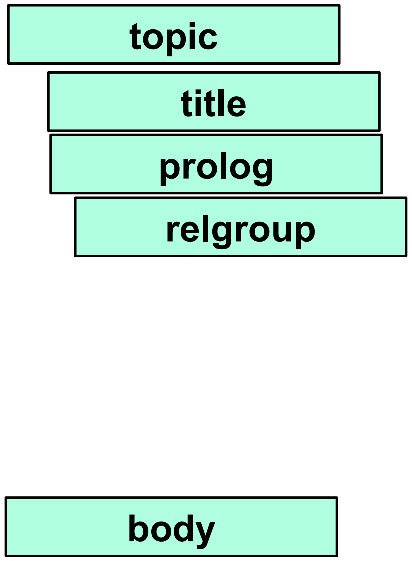
Then morph "topic" to "task"



- topic
  - title
  - prolog
    - metadata
    - relgroup
  - body

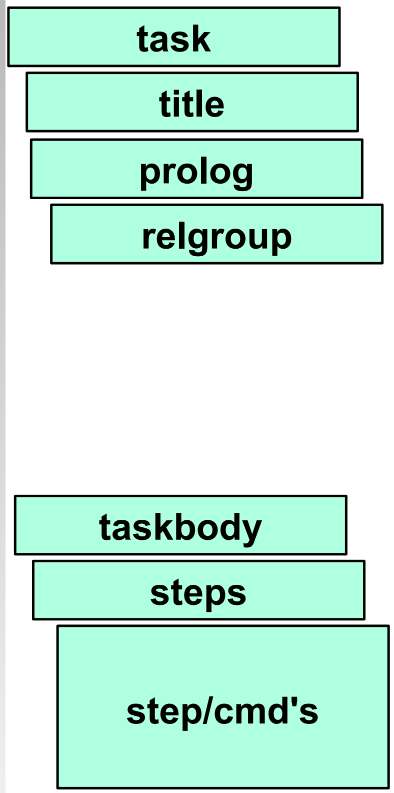
- task
  - title
  - prolog
    - metadata
    - relgroup
  - taskbody
    - prereq
    - context
    - steps
      - step
        - cmd, (info | substeps | tutorialinfo | xmp | choices)*, result?
    - result
    - taskxmp
    - postreq

This "Generic" topic...



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic SYSTEM "ditabase.dtd">
<topic>
    <title>To choose a location</title>
    <prolog>
        <relgroup role="friend" dupes="nodupes">
            <link url="H_CREATE.xml">Creating locations </link>
            <link url="H_SPECIFY.xml">To switch a User ID</link>
        </relgroup>
    </prolog>
<body>
    <ol>
        <li>Choose File - Mobile - Choose Current Location.</li>
        <li>Select a location.</li>
        <li>If necessary, click OK.</li>
    </ol>
</body>
</topic>
```

becomes this "task" topic



task

title

prolog

relgroup

taskbody

steps

step/cmd's

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task SYSTEM "ditabase.dtd">
<task>
    <title>To choose a location</title>
    <prolog>
        <relgroup role="friend" dupes="nodupes">
            <link url="H_CREATE.xml">Creating locations </link>
            <link url="H_SPECIFY.xml">To switch to a User ID</link>
        </relgroup>
    </prolog>
<taskbody>
    <steps>
        <step><cmd>Choose File - Mobile....</cmd></step>
        <step><cmd>Select a location.</cmd></step>
        <step><cmd>If necessary, click OK.</cmd></step>
    </steps>
</taskbody>
</task>
```

General patterns for more complex task topics

- Cast "indented" content as *info*
- All subheadings start a new sub-topic
  - Content preceding the task becomes a sub-task section of *taskbody/context*
  - Content following the task becomes a sub-task section of *taskbody/result*

What about tables?

- Content in a table is very interesting, XML-wise!
- Notes task topics contain lots of two-column tables
  - Option - Description
  - Click... - To...
  - Name - Mood Stamp
- Can represent most two-column tables in DITAbase as a dl "definition list"

Examples from Notes End-user Help

- "Basic" tasks with context, steps, result
  - *To change the size of SmartIcons*
- Tasks with sub-tasks
  - *Using database libraries to group databases*

Examples from Notes End-user Help *(continued)*

- "Big" task with sub-tasks
  - *Embedding data in a Notes document*
- Option - Description table in a task
  - *Creating appointments, anniversaries, reminders, and all-day events*
- A "concept" topic
  - *Navigation buttons: moving around*

Linking and navigation

- Ditabase provides several ways to specify a link
  - A *relgroup* in the topic *prolog* lists "see also's" or "related topics"
  - xref
  - link
  - ref
  - ldesc
- Link attributes provide additional detail
  - url
  - role (parent, child, friend, sibling)
  - type (topic-type; fig, table, list, heading)

TOC, Index, and Search

- Have experimented with:
  - Using JavaHelp XML ToC & Index files plus DHTML to produce ToC and Index navigators
- No widespread solution yet for XML search

Multiple uses and presentation using XML

- XML-based topics are well-suited for use as:
  - content for UA helpsets
  - content for Web-based info centers
  - section or chapter content for books
  - sections for POD booklets
  - articles for journals or technical newsletters
  - many others!

How to reuse a topic

- What use do you want to make of this topic? This is an "application context."
- Devise a DTD to represent how the new application uses the topics:
  - provides link access between topics
  - enables making indexes and ToCs
  - provides ways to attach new overviews or styling concerns

How to reuse a topic *(continued)*

- Write additional XSLT template rules
  - augment the basic set (HTML output) with rules for the new, superset elements
  - override existing template rules that need to be treated differently in the new context
- If the new application is greatly different, rewrite the processors accordingly (XSL Formatting Objects, other XML)

Mechanisms for reuse

- Definitional components that can be moved to a high-level location:
  - link descriptions (rels, links)
  - bibliographic descriptions
  - class definitions ("casts" for elements)
  - document metadata
- Referencing components:
  - local implementation of XInclude
  - citations, cross-referenced labels

How to extend a topic

- Relate the new "info type" to an existing type (concept, reftopic, task, or something new classed directly from a generic topic)
- Define new specialized element names
- Within content models of the same kind, you can further constrain allowed content, or redefine things of a kind (upwardly transformable)

How to extend a topic *(continued)*

- Create an authoring DTD
- Create a supporting XSL style sheet:
  - map new elements/attributes to existing "base" template rules
  - augment with new template rules where a processing rule does not exist
- Document the new specialization

Conclusions

- Is an XML for user assistance possible?
  - Yes.
- Has it worked?
  - Yes. We already moved over 7,000 topics and 5,000 pages of Notes/Domino content into DITAbase XML.
- Stay tuned.
  - The story goes on...

Demo of XML-based user assistance at Tivoli Systems

- Chris Rothemich
- Linda Meyer

Links

- Notes.net
  - Domino & Notes Doc Library
- www.alphaworks.ibm.com
- www.lotus.com/xml
- www.ibm.com/developer/xml

Contact Information

- **John Hunt**
  User Assistance Architect
  Lotus Development Corporation
  john_hunt@lotus.com

- **Don Day**
  Advisory Software Engineer
  IBM Corporation
  dond@us.ibm.com