



Team FE Metadata Tool

Architecture Presentation

Team members: Sanford Johnston, Jacob Espinoza, Lucas
Hermann, Isaias Gerena

Project Overview

- Purpose: Update common tags once for a set of products
- The Metadata Tool will operate in sessions.
 - New/open/save sessions
 - Import (append Table of Contents)
 - Export (XML files for all data products)
 - Main view:
 - Table of Contents tree
 - Jump to an element in the tree
 - Visual cues for progress
 - Empty elements needing input
 - Elements with unverified content
 - Elements verified to be correct
 - Question/Answer/Verify
 - Previous/next - modify element contents in order
 - Saves automatically

Key Architectural Drivers

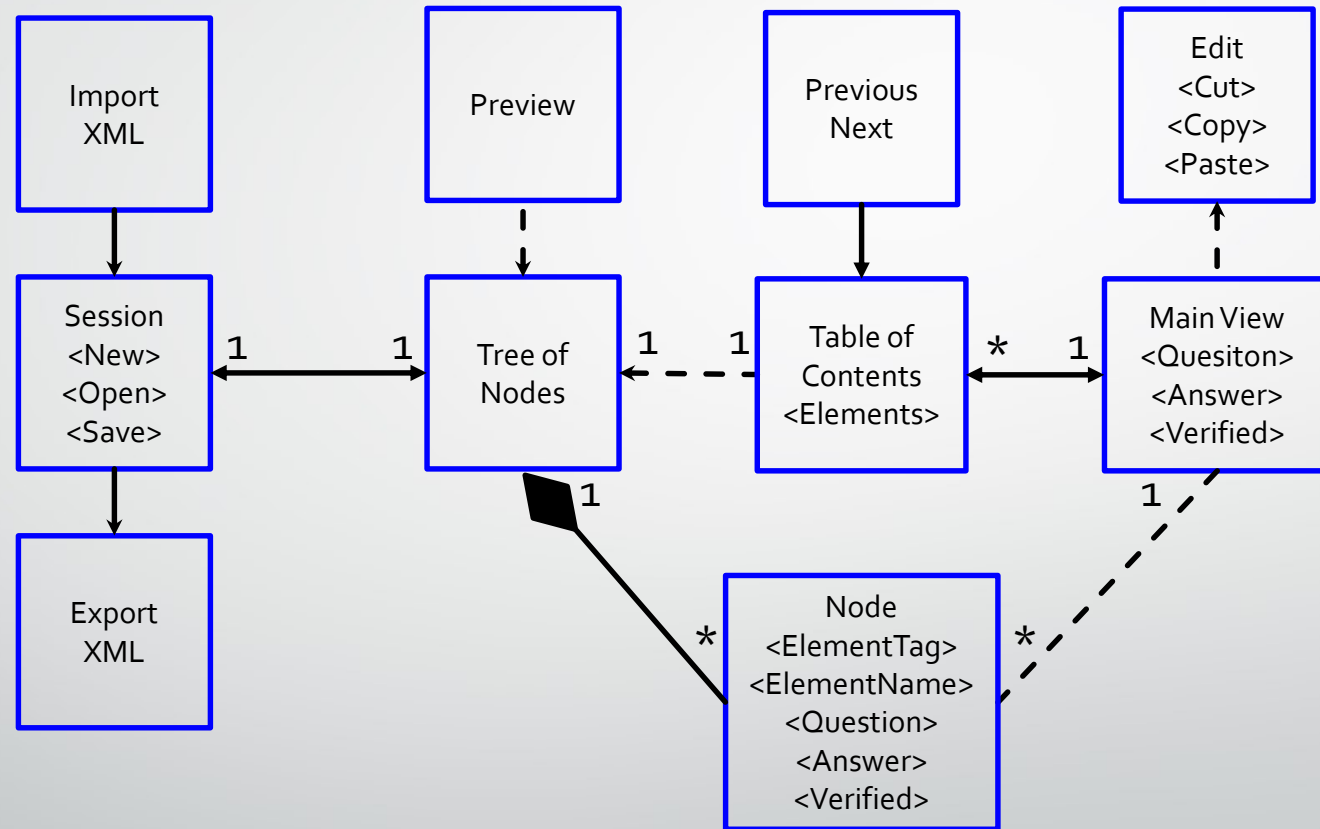
- Maintain the tree integrity of the metadata templates
 - Duplicate tags containing identical information for all data products need to not be duplicated in the Table of Contents
 - Duplicate tags that have unique information for a product need to appear in the Table of Contents with a unique identifier and treated carefully on export
- Program behavior and flow
 - Users need a visual cue to know the status of each element in the Table of Contents (empty, updated, verified)
 - Ask the user a question, record their answer (like a 'wizard')
 - Fewest commands possible

Architecture Style Choices

- Repository
 - Central data store of the metadata (Node Tree) and control of its elements
 - Blackboard - some elements are reactive to updates to the tree (TOC, Main View)
 - Display of knowledge sources in the form of visual cues
 - Methods perform actions on data accessed from server (TOC) and are then written back to the data store (Node Tree).
- Pipe and Filter
 - Passing input data through a sequence of data transforming components.
 - Ideally, program typically uses a linear flow through the TOC elements.
 - All actions on an element must preserve element data format

WINNER - Repository

Project Architecture Diagram



Conclusion

- Repository Architectural Style
- Issues:
 - Control of behavior and flow
 - Interaction of many GUI components.
 - Designing the node to capture requirements
 - Handling duplicates
 - Testing
 - Learning curve
- Risks
 - Time / Schedule
- Questions?