Automating the Management of the Traceability Relation

Hyun Cho
robusta@uab.edu

Advisor: Dr. Jeff Gray

Software Composition and Modeling Laboratory
SoftCom
Department of Computer and Information Sciences
University of Alabama at Birmingham

Software changes are inevitable in the software development lifecycle. The scope and cost of such changes can be estimated through impact analysis. However, impact analysis is often performed only when it is absolutely necessary due to the level of effort and accuracy of the analysis when performed too early. This is due to the labor intensive and error prone nature of maintaining up-to-date traceability relations. This poster introduces an automated approach to traceability management that informs development activities and assists in tool integration.

Challenges in Traceability Relation Management

•Traceability Relation
- ✓The link between the artifacts of software development
- ✓A traceability relation follows the life of a requirement in a forward and backward direction

•Purpose of Traceability Relations
- ✓Assess the impact of a change in a requirement, design, implementation, and test case
- ✓Verify that all requirements of the system are designed, implemented, and tested

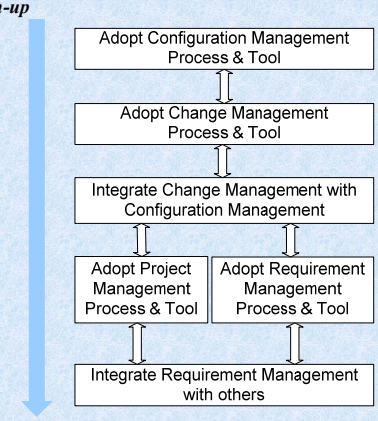
•Benefits of Traceability Relations
- ✓Improves software maintainability
  - Easily identify artifacts (e.g. requirement, design, and implementation) related to the changed feature
  - Locate the pieces of artifacts directly
- ✓Help in planning the schedule and estimating cost/effort

•Challenges in Traceability Relation Management
- ✓**Granularity**: Software artifacts are documented using different notations. Such heterogeneity makes it difficult to create an appropriate level of granularity between artifacts
- ✓**Evolution**: Traceability relations should evolve with their related artifacts. However, maintaining up-to-date traceability relations is difficult due to the labor intensive and error prone nature
- ✓**Process Integration**: Define how to use the traceability relations and when to use the information

Integration Approach

•Incremental process and tool integration

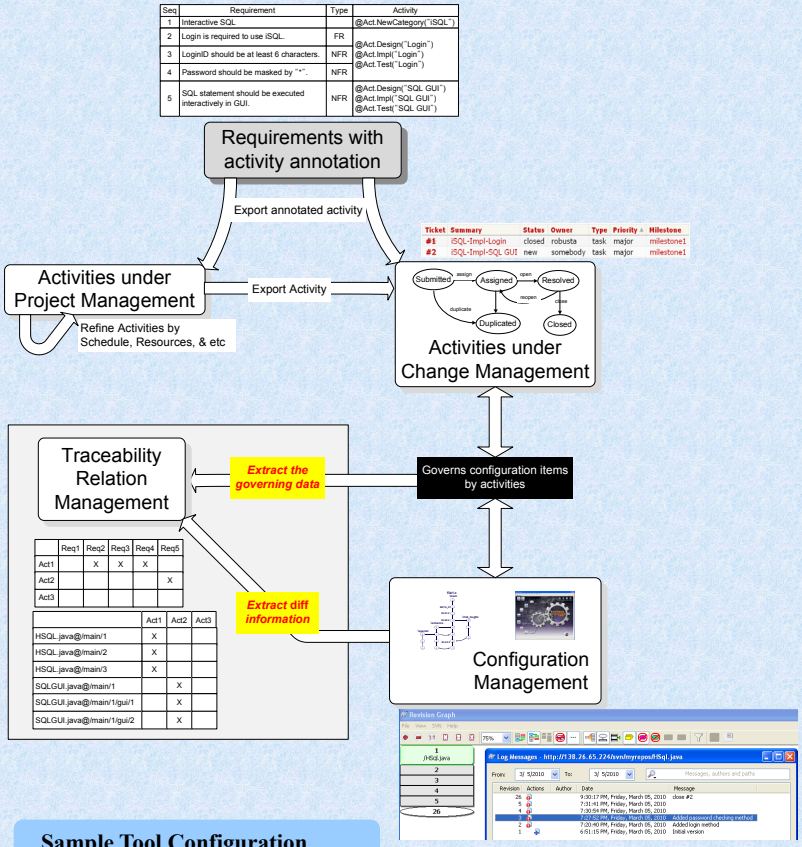
Bottom-up



Top-Down

Activity-Driven Traceability Relation Management

Activities govern the evolution of software artifacts



Sample Tool Configuration

No	Product Configuration	Integration				Remarks
		RM/PM	RM/CM	PM/CM	CM/SCM	
1	IBM Rational DOORS MS Project IBM Rational ClearQuest IBM Rational ClearCase	I	I	I	I	* Configured by commercial tools * Tool itself provides integration * Invest for tool purchase * Available many 3rd party integration tool
2	IBM Rational DOORS MS Project IBM Rational Change IBM Rational Synergy	I	I	I	I	
3	MS Excel MS Project Trac SVN	N/A	N/A	N/A	I	* Mixed configuration: commercial and open source * Need mass customization

Contribution

•Integration with existing development processes and tools
- ✓Commercial tools provide the integration as a feature of the product
- ✓Open source tools provide APIs or plug-ins for integration

•Can automate the management of traceability relations
- ✓Reduces traceability relation management errors
- ✓Minimizes the amount of manual work in maintaining relations

• Can control the granularity of traceability links
- ✓The granularity of traceability relations may be proportional to that of change activity

•Can help to improve the accuracy of impact analysis