

When and How to Visualize Traceability Links?

Andrian Marcus,
Xinrong (Travis) Xie,
Denys Poshyvanyk

Department of Computer Science
Wayne State University

Why We Visualize Links

- Traditional representation
 - Matrix, graph, etc
- Need for visualization
 - Recovery
 - Maintenance
 - Browse

When to Visualize

- Different Software Engineering tasks
- Many types of software artifacts
- Multiple attributes and categories of links

What to Visualize - Elements of Traceability Links

- Elements
 - Source
 - Target
- Properties of elements
 - Artifact name
 - Artifact type
 - Location
 - Creation time, Update time, Version, etc

What to Visualize - Properties of Traceability Links

- Discovery method
- Creation time, update time, version
- Usage history (like CVS)
- Status
- Documentation

What to Visualize - Categories of Traceability Links

- Action-based view
 - Dependency Links [Pohl 1996]
- Recovery method-based view
 - Lost, Warning, False Positive [Lucia et al 2005], Normal
- Artifact-based view
 - Requirements, Source Code, Specifications, Test cases, etc
- Collaboration view
 - Vertical, Horizontal

Visualization Requirements

1. Integrate with traceability recovery tools, IDE
2. Various data representation formats
3. User views based on stakeholders
4. Customizable views of traceability data
5. Capture and maintain usage history
6. Editable of links and artifacts
7. User query and filtering
8. Others...

TraceViz

The screenshot displays the Eclipse IDE with the TraceViz plugin. The main editor shows the source code of `WeekendCalendarImpl.java`. The Package Explorer on the left shows the project structure. The TraceViz window is open, showing a visualization of dependencies between artifacts. The visualization is divided into four quadrants: Handbook, Developer Guide, Test, and Data. Each quadrant contains colored boxes representing artifacts and their relationships. The Handbook quadrant shows 'First Step', 'Gantt', 'Chart', 'Resource', and 'Miscellaneous'. The Developer Guide quadrant shows 'Introduction', 'Java', 'Eclipse Create Project', and 'Deployment'. The Test quadrant shows 'CalendarDependency', 'Event', 'Hierarchy', and 'Time'. The Data quadrant shows 'Resources', 'Themes', 'xlsfo', and 'xlsl'. A 'Link' table on the right provides details about the selected link, including source and target artifacts, discovery method, creation time, update time, and documentation. A 'Browsing...' table at the bottom right shows a list of links with source, target, and time information.

Element area (points to the Source and Target views)

Link area (points to the Handbook, Developer Guide, Test, and Data views)

Information area (points to the Link table)

User View (points to the View: Artifact-Centric dropdown)

Name	Value
Source	
Target	
Link	
Discovery method	semi
Creation time	
Update time	
Documentation	
Text	This
Time	08-1
Author	Andr
Version	

Source	Target	Time
GanttCalend	Resource	05-
GanttCalend	Event	05-
GanttCalend	Themes	06-
GanttCalend	Calendar	08-

Conclusion and Future Work

- Conclusion
 - Requirements for visualizing traceability links
 - A prototype of visualization tool
- Future work
 - Implementation of the tool
 - Conduct a user study
- Refine the requirements list at TEFSE'05!

For More Details

- Severe Group

<http://www.severe-group.org/>

- Contact Persons

Andrian Marcus amarcus@wayne.edu

Denys Poshyvanyk denys@wayne.edu

Xinrong Xie xxr@wayne.edu

Complete Visualization Requirements

1. Integrate with an IDE
2. Various artifacts
3. Customizable views of traceability data
4. Browse traceability through multiple types of user interactions
5. Capture and maintain browsing history
6. Editable of links and artifacts
7. User query and filtering
8. Integrate with traceability recovery tools
9. Interoperate with other SE tools
10. Various data representation formats
11. Comprehensive configuration management and change tracking facilities
12. Analyze and summarize data on traceability process and links