Graph-based Visualization of Requirements Relationships

Philipp Heim\textsuperscript{1}, Steffen Lohmann\textsuperscript{1}, Kim Lauenroth\textsuperscript{2}, Jürgen Ziegler\textsuperscript{1}

\textsuperscript{1}Interactive Systems and Interaction Design
\textsuperscript{2}Software Systems Engineering
University of Duisburg-Essen, Germany
Motivation

1. Requirements are often interrelated
   - Multiple relationships
   - Relationships of different types:
     • User-defined relations
     • Content relations
     • Shared metadata relations (focus of this talk)

2. Visualizing relationships facilitates
   - Understanding of the requirements themselves
   - Understanding of their dependencies

3. Existing requirements management tools
   - Mainly use lists, tables, trees and matrices
   - Limited capacity to show multiple relationships of different types
Our Approach

- **Graph-based visualization of Requirements Relationships**
  - As extension to existing visualization forms
  - Represents requirements as nodes and relationships as edges
  - Allows for flexible visualization of multidimensional relationships

- **Problems to meet:**
  - Graphs do not scale well to large datasets
  - They get over-cluttered and hence difficult to understand

- **Our solution:**
  - Show a limited set only
  - Focus and context approach
Focus and Context Approach

- Global and local navigation

A: tree view
B: tag cloud
C: result list
D: detail view of the selected requirement (focus)
E: related requirements (context)
Graph-based Visualization

- Direct representation of relations

This can be also seen on the poster

A: Colored rings
B: Multiple relationships
**RDF Graph Transformation (ChainGraph)**

- Indirect representation of shared metadata relations (e.g. the keyword KW: spam)
- Presentation of shared metadata only
- **Direct** representation of shared metadata relations
- Reduced number of edges
- “ChainGraph”
Conclusion and Future Work

- **Pros:**
  - Single visualization (not distributed over several pages)
  - Direct representation of shared metadata
  - Fewer number of crossing edges
  - Following path

- **Cons:**
  - Not suited to visualize large numbers of requirements or large numbers of shared metadata

- **Future Work:**
  - Complete integration into the main system
  - Evaluation of benefits with the help of eye tracking
Thank you for your attention!
Any questions?