Example (Composed Maps): Cancel Reservation
Neil A. Ernst, Yijun Yu and John Mylopoulos (University of Toronto), from “Visualizing Non-Functional Requirements”
Martin S. Feather (California Institute of Technology), Steven L. Cornford (California Institute of Technology), James D. Kiper (Miami University) and Tim Menzies (West Virginia University), from “Experiences using Visualization Techniques to Present Requirements, Risks to Them and Options for Risk Mitigation”
Martin S. Feather (California Institute of Technology), Steven L. Cornford (California Institute of Technology), James D. Kiper (Miami University) and Tim Menzies (West Virginia University), from “Experiences using Visualization Techniques to Present Requirements, Risks to Them and Options for Risk Mitigation”
Martin S. Feather (California Institute of Technology), Steven L. Cornford (California Institute of Technology), James D. Kiper (Miami University) and Tim Menzies (West Virginia University), from “Experiences using Visualization Techniques to Present Requirements, Risks to Them and Options for Risk Mitigation”
Chuan Duan and Jane Cleland-Huang (DePaul University), from “Visualization and Analysis In Automated Trace Retrieval”
Sascha Konrad, Heather Goldsby, Karli Lopez and Betty H.C. Cheng (Michigan State University), from “Visualizing Requirements in UML Models”
Irwin Kwan, Daniela Damian and Margaret-Anne Storey (University of Victoria), from “Visualizing Requirements-Centered Social Network to Maintain Awareness within Development Teams”
Visual Support In Automated Tracing

Automated traceability generates candidate links between requirements and other software artifacts.

An analyst must manually evaluate the results.

Clusters are visualized.

Results are clustered to increase comprehension.

Tag clouds help an analyst understand why a particular artifact is linked to the query.

Visualizations help analysts to make decisions during trace analysis.

Links are displayed within the context of a physical hierarchy.

Jane Cleland-Huang and Rafał Habrat, DePaul University
Visual Analytics for Requirements-driven Risk Assessment

Robin A. Gandhi, Seek-Won Lee
Knowledge-Intensive Software Engineering (KISE) Research Group, College of Computing and Informatics, UNC Charlotte, NC, USA

1. The Position
   - Requirements Visualization plays an important role in dealing with the complexities from:
     - Numerous Dependability Requirements
     - Software-intensive Systems
     - Socio-technical Environments
   - Visual Analytics for Requirements is the creation of abstract visual metaphors based on quantitative and qualitative requirements metrics that help to systematically reason about the related software behavior in a large information space

2. Multi-dimensional Requirements Visualization

3. Visualizing Non-compliance Impact Analysis for Risk Assessment
   - Step 1 (Localization): Locate the most specific formal concept that characterizes the non-compliant requirement categories (Concept C1 for “RS: Monitoring” labeled as 1)
     - Concept C1 Explanation: To assess the risk related to the Threats of Unauthorized Activities that can damage the Asset of Enclave within a DIS Information System by exploiting the Vulnerabilities of Firewall and DIS Mail configuration, collectively evaluate the compliance levels of C&A requirements in the categories of Enclave Boundary Defense and Monitoring for estimating the effectiveness of the suggested “install Firewall and IDS with appropriate configurations” Countermeasure by these requirements to mitigate the Vulnerabilities

4. Visual Metaphors
   - Cohesive Bar Graph
     - Visualization Context: Formal Context
     - Visual Features of a Bar
       - Color: Requirement category presence and compliance levels
       - Height: Requirement category correlation index
       - Width: Risk coverage of a requirement category
       - Orac: Relative criticality
   - Cohesive Arc Graph
     - Visualization Context: Requirement Category
     - Visual Features of an Arc
       - Color: Requirement category compliance level
       - Arc Radius: Requirement influence factor is the degree of similarity between two requirement categories in correlating with other categories in the given scenario

5. References

On Requirements Visualization

Orlena C.Z. Gotel, Francis T. Marchese and Stephen J. Morris

www.systemsguild.com/GuildSite/Robs/Template.html

Contacts: ogotel@pace.edu, fmarchese@pace.edu, sjm@soi.city.ac.uk
Visualizing Aspect-Oriented Goal Models with AoGRL

GRL Model:

AoGRL Model:

Restructure!

With the help of aspect-oriented techniques the major concerns in the GRL model are now better encapsulated. Typical major concerns are stakeholders, non-functional requirements, and solutions (i.e., use cases).

Note that the details of use cases are not shown because they are modeled with Use Case Maps (UCMs) and only linked from the GRL model with URN links.

Legend:

(a) GRL Elements
(b) GRL Links
(c) GRL Satisfaction Levels
(d) GRL Contributions Types
(e) AoGRL
(f) URN

(c) Pointcut marker
(f) URN link
(b) GRL Links
(a) GRL Elements

Legend:

Contribution  Dependency  Correlation  Decomposition

Softgoal  Goal  Task  Resource  Actor  Belief  Actor Boundary

Satisfied  Weakly Satisfied  Weakly Denied  Denied  Undecided  Conflict

Make  Break  Help  Hurt  Some+  Some−  Unknown

Aspect marker
Scenario Map for Visualizing Classified Scenarios
Keitaro Osawa and Atsushi Ohnishi

**Scenario Similarity Map**

- **Different**
  - train ticket reservation
  - goods purchasing
  - claim for the loss on insurance

- **Same**
  - bus ticket reservation
  - another flight ticket reservation
  - cancellation of flight ticket
  - given scenario (flight ticket reservation)

- **System**
  - Same
  - Different
An Improved Fisheye Zoom Algorithm for Visualizing and Editing Hierarchical Models

Tobias Reinhard, Silvio Meier, Martin Glinz
University of Zurich
Visualizing the Analysis of Dynamically Adaptive Systems Using * and DSLs

Peter Sawyer, Nelly Benmamor, Heather J. Goldby, Betty H.C. Cheng, Danny Hughes, Paul Gero

This work has been supported in part by NSF grants EA-0020452, EA-013124N, EIA-9820213, CoT-9909173, and by the Office of Naval Research under Grant N00014-99-1-1394. Aeronautics Corporation, Burtis Corporation/Research, a grant from Micropipette Auto-Universal (MAU) System OS/2000-99.
Visualizing Product Line Domain Variability by Aspect-Oriented Modeling

Reinhard Stolber, Silvio Meier, Martin Glinz
University of Zurich

Commonality

Orthogonal Variability

Traceability

Decision Model

... and how to use your Domain:

1. Do your Product Decisions

2. Receive an automatically generated Application Product
A Visualization Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering

A.M. Sen and S.K. Jain
REV'08 VISUALIZATIONS
Visualizing Informal Communication with FLOW

by Kurt Schneider, Kai Stapel, and Eric Knauss

Security Requirements Example

Information Flow in Related Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Visual Elements</th>
<th>UML Elements</th>
<th>Rational Rose</th>
<th>Initial Study Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOW</td>
<td>Flow diagram</td>
<td>Use case</td>
<td>Use case</td>
<td>Work breakdown</td>
</tr>
<tr>
<td>UML</td>
<td>Activity Diagram</td>
<td>Use case</td>
<td>Use case</td>
<td>Work breakdown</td>
</tr>
<tr>
<td>LENE</td>
<td>Event</td>
<td>Event</td>
<td>Event</td>
<td>Event</td>
</tr>
<tr>
<td>IDEF</td>
<td>Information</td>
<td>Information</td>
<td>Information</td>
<td>Information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>FLOW</th>
<th>UML Activity Diagram</th>
<th>Rational Rose</th>
<th>Initial Study Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Flow Diagram</th>
<th>UML Activity Diagram</th>
<th>Rational Rose</th>
<th>Initial Study Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Activity</td>
<td>Activity</td>
<td>Activity</td>
<td>Activity</td>
<td>Activity</td>
</tr>
<tr>
<td>Event</td>
<td>Event</td>
<td>Event</td>
<td>Event</td>
<td>Event</td>
</tr>
<tr>
<td>Information</td>
<td>Information</td>
<td>Information</td>
<td>Information</td>
<td>Information</td>
</tr>
</tbody>
</table>
Modelling Trust Requirements by Means of a Visualization Language

Innovative Informatics, School of Computing and Technology, University of East London, UK
Kamaljit Kaur Bimrah, Haralambos Mouratidis, David Preston {bimrah, h.mouratidis, d.preston}@uel.ac.uk

AIM:
To develop a reasoning and modelling framework that will enable information system developers to consider trust and its related concepts collectively during the development of information systems.
On Usability in Requirements Trace Visualizations
Stefan Winkler <stefan.winkler-et@fernuni-hagen.de>

Charts and Tables
- tested
- failed
- passed

<table>
<thead>
<tr>
<th>Module</th>
<th>#Req</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>56</td>
<td>15823</td>
</tr>
<tr>
<td>Module 1.1</td>
<td>26</td>
<td>10536</td>
</tr>
<tr>
<td>Module 1.2</td>
<td>30</td>
<td>5287</td>
</tr>
<tr>
<td>Module 2</td>
<td>24</td>
<td>3578</td>
</tr>
</tbody>
</table>

Cross-References
- Req 1: The system shall...
  - Req2
  - Req3
  - Class1
- Req 2: The system shall...
  - Req3
  - Req1
  - Req2
- Req 3: The system shall...
  - Req2
  - Req1
  - Req2

Trace Networks / Graphs
- SysReq1: The system shall...
  - SoftReq5: The system shall...
    - Class 1

User roles, tasks, and access modes
- User wants to complete a task
  - User needs to access the visualization
- User has to choose what (task)
- Visualization results in adequate visualization
- User (role) results in adequate visualization

Making visualizations usable will better expose the benefits of traceability.
Visualizing Use Case Sets as BPMN Processes

Problem Description
Especially in SOA projects but in other projects as well, software has to support business processes. These processes are part of the software requirements as well as the Use Case descriptions. Both models overlap which wastes time for creating and maintaining the software requirements.

Solution
Generate BPMN processes from textual Use Cases
- Generate sequential flows for scenario
- Create decision points for extensions
- Match Pre- and Postconditions of Use Cases
- Join Use Cases

Advantages
- Useful to generate skeleton business processes
- Helps visualizing and identifying dependencies between Use Cases
- Helps validating dependencies between Use Cases
- Helps validating pre- and postconditions of Use Cases

Open Questions for Future Work
- Non-Literal matching of pre- and postconditions
- Also useful for comparing business processes with given Use Cases
- Advanced Tool-Support

Daniel Lübke, Kurt Schneider and Matthias Weidlich
FG Software Engineering, Leibniz Universität Hannover
Hasso Plattner Institute, Potsdam
daniel.luebke@inf.uni-hannover.de
Visualization of Feature Survival in Platform-Based Embedded Systems Development for Improved Understanding of Scope Dynamics

Visualization can improve real industrial projects.

This method for visualizing the scoping process in platform-based development of embedded systems shows the decision process of including or excluding features that are candidates for the next release.

Charts are evaluated in a large-size embedded system platform project and indicate that the visualization of feature survival and scope dynamics can improve the understanding of the decision process of platform scoping in real industrial projects.

**FEATURE SURVIVAL CHART**
The red lines show out-scoped features. The green lines show features in scope (light green for primary flow features and dark green for secondary flow features). The survivors are placed at the top, as the graph is sorted on duration in scope from last baseline.

**FEATURE GROWTH CHART**
The green area represents the number of in-scoped features, red the number of out-scoped and gray the number of undecided.

*Future work includes improving user interaction.*
Visualizing the Impact of Non-Functional Requirements on Variants – A Case Study

Clotilde Rohleder
Clotilde.rohleder@siemens.com
Visualizing Rationale

Need a Zoned Alarm Option

People want alarm while inside house

People want to feel safe

People want alarm while outside house

People want to protect property

Market will support luxury options

Need a configurable electronic Alarm

If Housing Market falls, retail price will fall