2013-10-25

IMPREC USER MANUAL

ImpRec is a prototype tool supporting change impact analysis developed as part of a research project at the Department of Computer Science at Lund University. The tool has been developed as a proof-of-concept of the potential benefits of reusing traceability from previously completed impact analyses. ImpRec is a *recommendation system for software engineering* that uses a combination of *information retrieval-based trace recovery* techniques and analysis of *network* structure to identify potential impact for a given tracker case.

Throughout this manual, terms in **bold** represent components of the GUI. Terms in *italic* are further described in Appendix A - Glossary.

Markus Borg

markus.borg@cs.lth.se

Software Engineering Research Group

Dept. of Computer Science

Lund University

ImpRec builds on research conducted at the industrial excellence center EASE – Embedded Application Software Engineering[1] in research theme D, *Aligning requirements and verification*.

CONTENTS

[1] ease.cs.lth.se

Step by step instructions

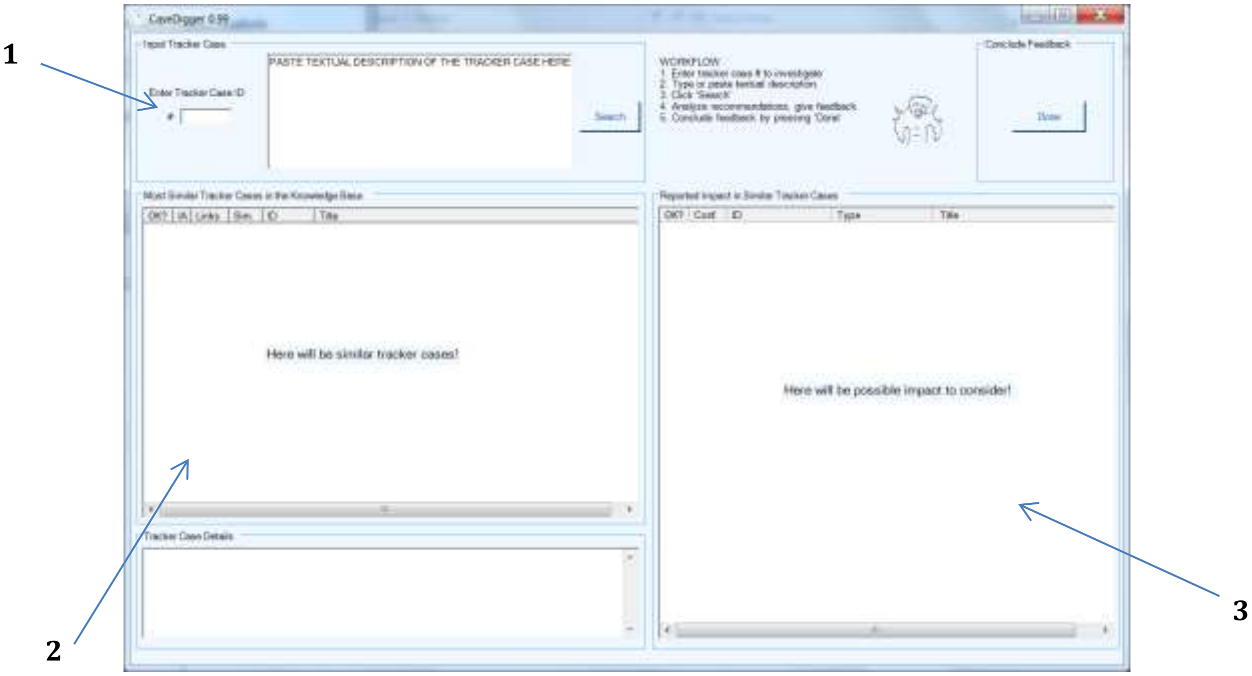
Step 1. ImpRec Startup

The first time ImpRec is started, the *knowledge base* (i.e., an indexed database containing impact analysis reports attached to previous tracker cases) needs to be built. This operation might take a few minutes, but is only required once. Simply follow the dialog, no configurations need to be made.

When the knowledge base is built, the tool is ready to use. Figure 1 displays the GUI at startup.

The main components are the following:

- The **input area** (cf. 1 in Figure 1). In this area the #ID of the investigated tracker case is entered, as well as its textual description. The textual description entered in the text field is used to search for similar tracker cases in the *knowledge base*. Text can be entered freely, however we recommend pasting the tracker case title and/or the tracker case description. You are encouraged to experiment with various textual inputs!
- The **tracker case output area** (cf. 2 in Figure 1) lists the most similar tracker cases in the knowledge base. The number of tracker cases below is restricted to 20 to avoid information overload.
- The **impact output area** (cf. 3 in Figure 1) lists the software artifacts pointed out in impact analysis reports attached to the similar tracker cases in the knowledge base. These items are ranked according to both *textual similarities* and the *network structure* of the knowledge base.


1
CaveDigger 0.99
Input Tracker Case
Enter Tracker Case ID
#
PASTE TEXTUAL DESCRIPTION OF THE TRACKER CASE HERE
Search
Conclude Feedback
Done
Most Similar Tracker Cases in the Knowledge Base
OK? | IA | Links | Sim. | ID | Title
Here will be similar tracker cases!
Tracker Case Details
Reported Impact in Similar Tracker Cases
OK? | Conf. | ID | Type | Title
Here will be possible impact to consider!
2
3

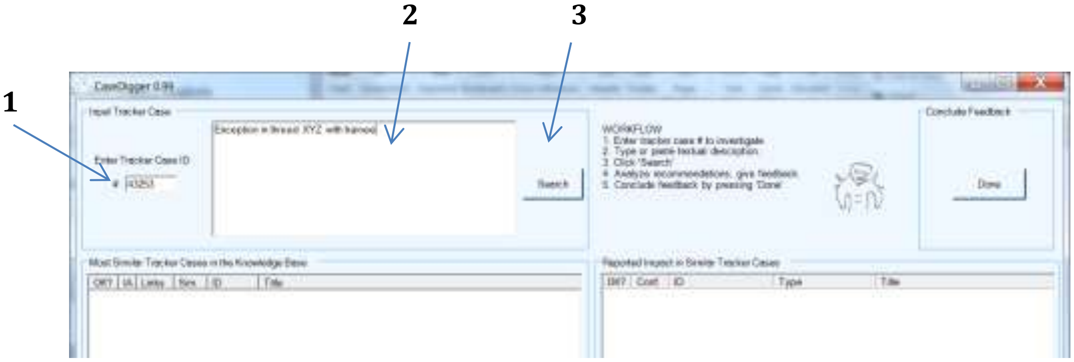
WORKFLOW
1. Enter tracker case # to investigate
2. Type or paste textual description
3. Click 'Search'
4. Analyze recommendations, give feedback
5. Conclude feedback by pressing 'Done'

Figure 1: ImpRec at startup. The tool is ready to use.

STEP 2. CONDUCTING AN IMPACT SEARCH

In the **Input area**, complete the following steps:

1. Enter #ID of the tracker case that is under investigation (e.g., “43253”).
2. Enter description (e.g., “Exception in thread XYZ with trainee”) of the tracker case in the text field (cf. 2 in Figure 2). A typical use case is to paste the title of the tracker case. However, we encourage you to experiment with title + description as well as free text searches.
3. Click the **Search button** (cf. 3 in Figure 2).


2
3
1
CaseDigger 0.99
Input Tracker Case
Enter Tracker Case ID
Search
Conclude Feedback
Done
Most Similar Tracker Cases in the Knowledge Base
CRT | IA | Linky | Sim | ID | Title
Reported Impact in Similar Tracker Cases
Sim? | Conf | ID | Type | Title

# 43253

Exception in thread XYZ with trainee

WORKFLOW
1. Enter tracker case # to investigate
2. Type or paste textual description
3. Click 'Search'
4. Analyze recommendations, give feedback
5. Conclude feedback by pressing 'Done'

Figure 2. Conducting an impact search. Tracker case id and input text have been entered.

4. Results are displayed in the **tracker case output area** and the **impact output area (cf. Figure 3)**.

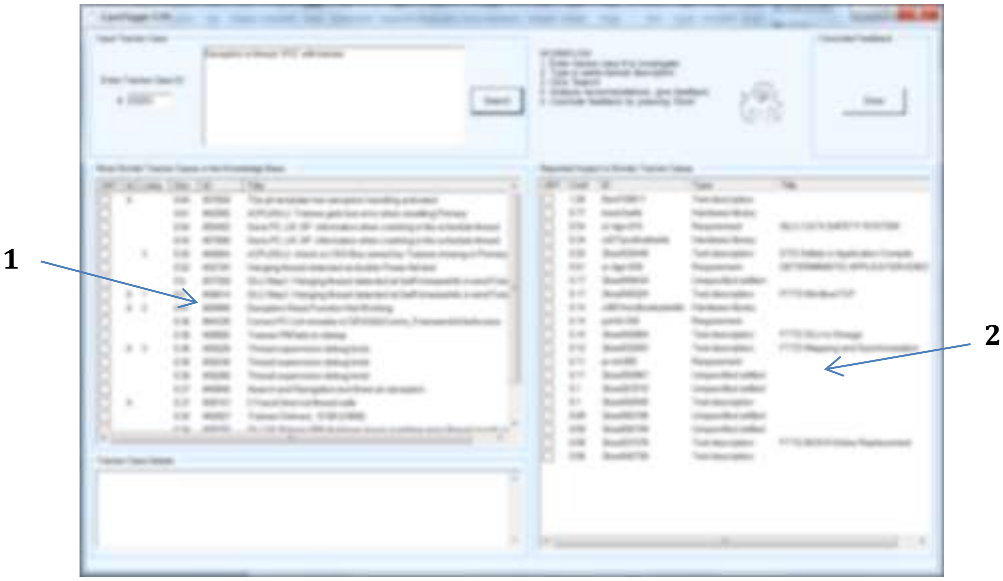


Figure 3. Overview of the search results. The **tracker case output area** (cf. 1) and the **impact output area** (cf. 2) are populated by search results.

Step 3. Investigate Output – Similar Tracker Cases

- In the **tracker case output area** (cf. 1 in Figure 4) you can browse the 20 most similar tracker cases in the knowledge base.
- The **IA column** contains an 'X' if a previous impact analysis report is available.
- The **Links column** displays the number of links to impacted software artifacts identified in the previous impact analysis reports attached to the tracker case.
- The **Sim. column** shows a measure of the *textual similarity* between the text you entered in the **input area** and the text contained in the tracker case. Note that the number is a relative measure, and that limited conclusions can be drawn based on an individual measure.
- If you select a tracker case in the **tracker case output area** (or multiple tracker cases by pressing SHIFT or CTRL), the impacted software artifacts pointed out from the attached impact analysis reports are highlighted in the **impact output area** (see 2 in Figure 4).
- The description of the selected tracker case, as well as its attached impact analysis report if available, is presented in the **Tracker Case Details** area.

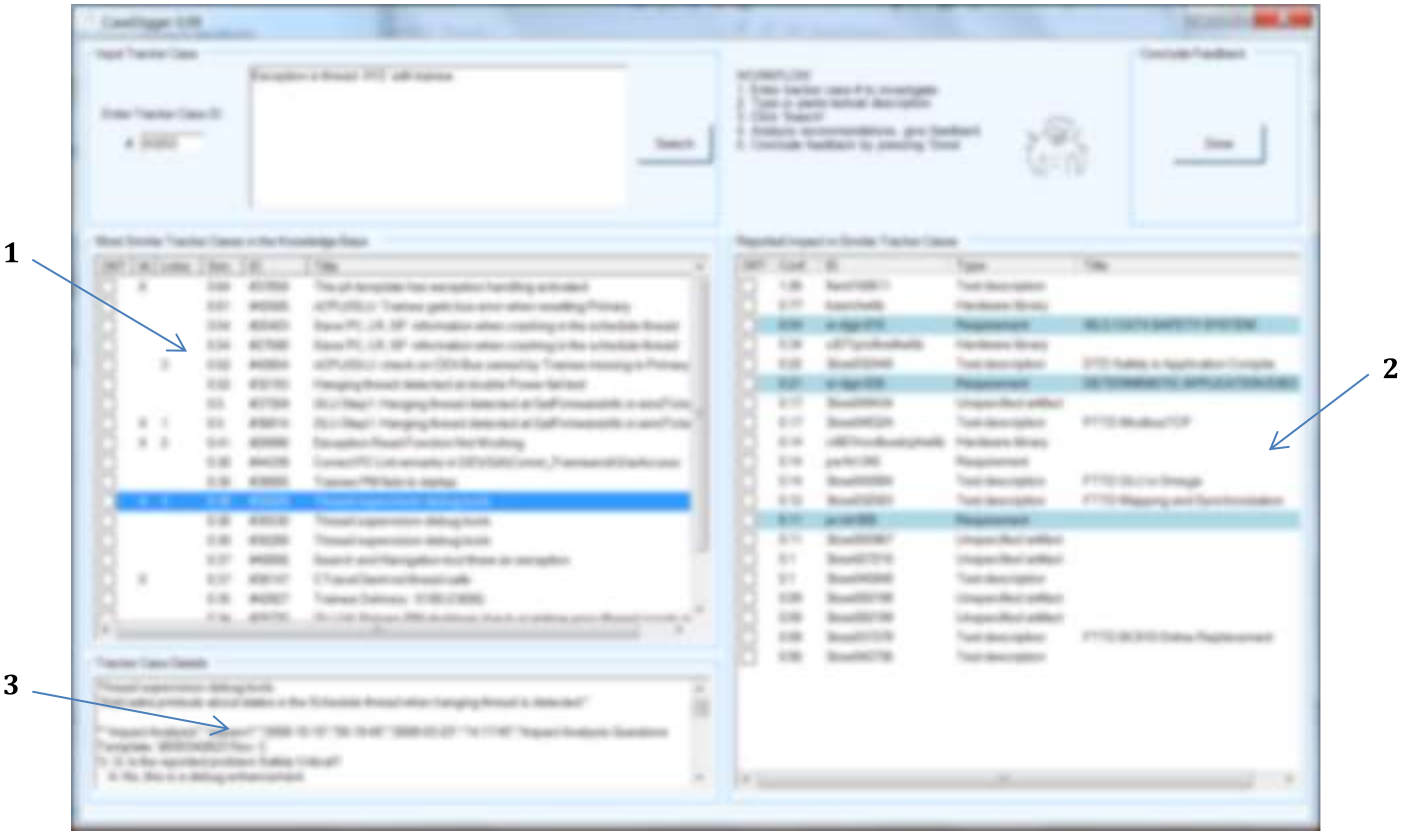


Figure 4. Tracker case #35529 is selected in the **tracker case output area** (cf. 1), the software artifacts previously pointed out as impacted (the three requirements SR-DGN-015, SR-DGN-038, and PR-INT-008) are highlighted in the **impact output area** (cf. 2).

STEP 4. INVESTIGATE OUTPUT – PREVIOUSLY IMPACTED SOFTWARE ARTIFACTS

- In the **impact output area** (cf. 1 in Figure 5) you can browse recommendations for possibly impacted software artifacts to consider for your current impact analysis.
- The number of software artifacts listed in the **impact output area** can be higher than the sum of "Links" column in the **tracker case output area** (cf. 2 in Figure 5). ImpRec searches deeper than those tracker cases listed, by also analyzing previous impact on tracker cases selected as "Related cases" in Merant Tracker.
- The software artifacts listed in the **impact output area** are ranked based on both *textual similarities* and the *network structure* originating from data mining in Merant Tracker. A merged ranking value is presented in the **Conf.** column. It is a relative number, thus limited conclusions can be drawn on these numbers in isolation (as is the case of the **Sim**. **column** in the **tracker case output area**).
- The **Type column** shows the category of the software artifact, if it was deducible from the data mining in Merant Tracker. Otherwise, the software artifact is listed as "Unspecified artifact".
- The **Title column** contains the titles of the software artifacts most often reported as impacted in the impact analysis reports in Merant Tracker. These titles have been manually extracted for your convenience.

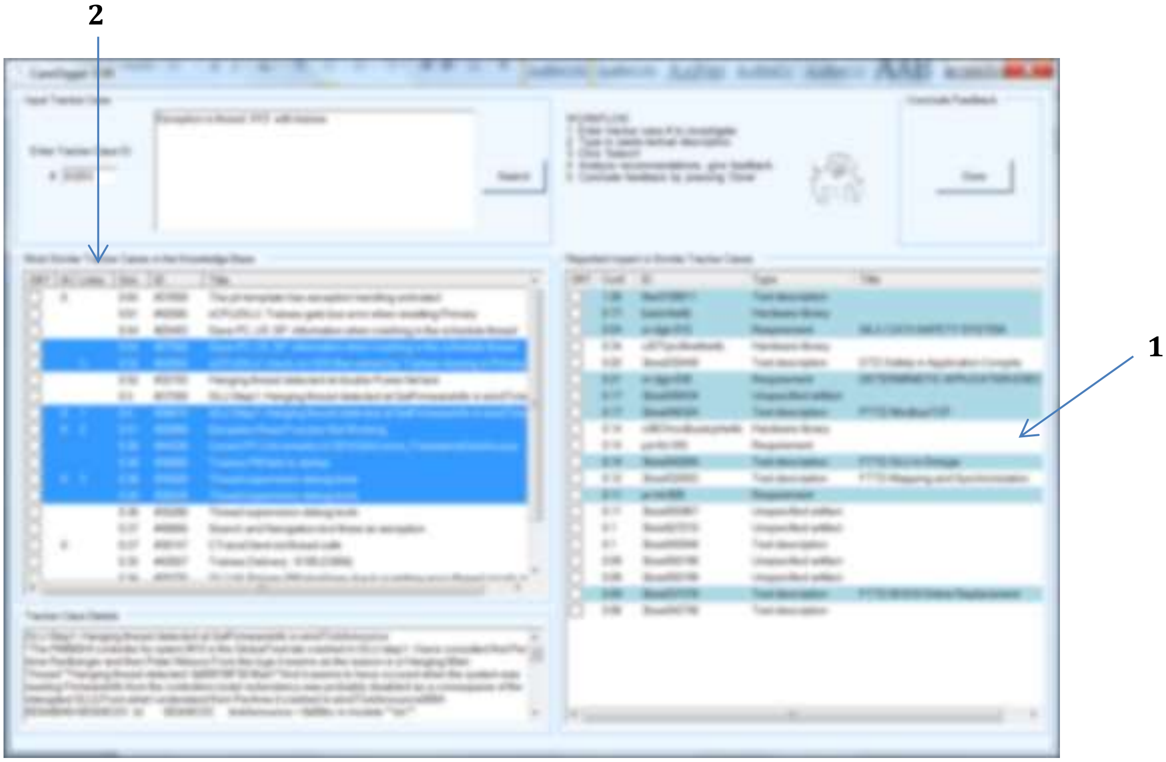


Figure 5. An example of multiple selection in the **tracker case output area** (cf. 1). The corresponding software artifacts are highlighted in the **impact output area** (cf. 2).

Step 5. Provide Feedback to the Research Study

When you are done with the impact analysis, we would greatly appreciate if you could provide feedback to us. Use the **OK? Columns** in the **tracker case output area** and the **impact output area** to signal which recommended artifacts you actually used in your work task. Also, this copies your selection to the Windows clipboard.

1. Check the tracker cases (cf. 1 in Figure 6) in the **tracker case output area** that were meaningful in your impact analysis work task.
2. Check the software artifacts (cf. 2 in Figure 6) that you will point out as impacted by the tracker case you are currently analyzing in the **impact output area**.
3. Click the **Done button** in the upper right corner of the GUI (cf. 3 in Figure 6) to save your feedback to a local file containing user statistics.
4. The IDs of the selected tracker cases and the selected impacte software artifacts are copied to the Windows clipboard.

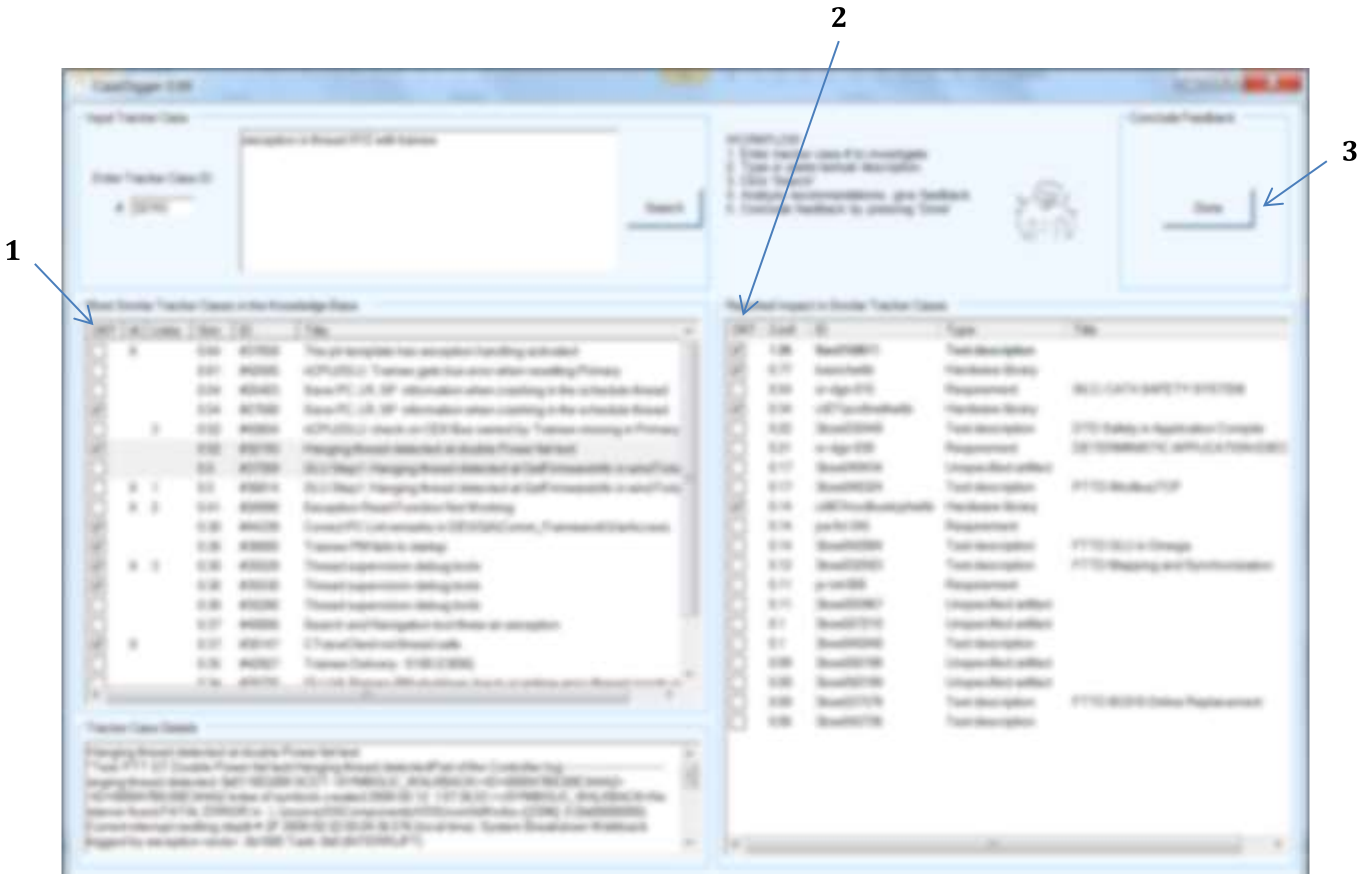


Figure 6. The feedback system in ImpRec. Use the checkboxes to signal appropriate output in the **tracker case output area** and the **impact output area** and confirm the selection by clicking the **Done button**.

APPENDIX A - GLOSSARY

alignment of requirements and verification

Adjustments of requirements and verification activities to coordinate and co-optimize the “two ends of software engineering”. This includes practices, processes and tools organized to improve the coexistence of these often separated areas.

information retrieval-based trace recovery

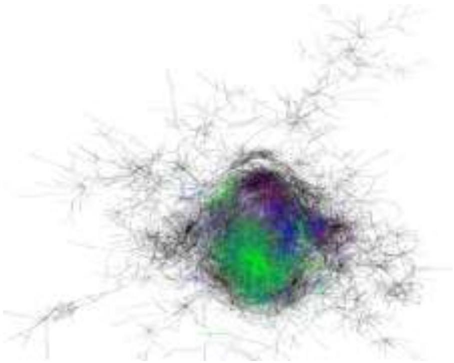
An approach to semi-automatically propose candidate traceability links among existing software artifacts. The approach relies on information retrieval (search) techniques to identify such traceability links based on the textual similarities of software artifacts.

knowledge base

A representation of software artifacts and their relations stored in ImpRec as a semantic network. The information is gathered by data mining in Merant Tracker, using both previous tracker cases and their attached impact analysis reports.

network structure

The knowledge base in ImpRec is represented by a semantic network. The network, shown to the right, contains about 30.000 software artifacts and their complex relations. ImpRec uses the network structure to analyze centrality measures (i.e., importance of software artifacts) and to predict impact propagation. The network structure is an important factor in the ranking of possibly impacted software artifacts.



recommendation system for software engineering

A software application that provides information items estimated to be valuable for a software engineering task in a given context. ImpRec aims at supporting developers conducting change impact analysis with respect to information seeking goals.

textual similarity

ImpRec calculates textual similarity between software artifacts using an open source library for indexing and searching text. The similarity scores between artifacts calculated by Lucene builds on the following:

1. how often individual terms appear in both artifacts (e.g., “download” 3 times, “OLU” 2 times)
2. how rare the overlapping terms are (i.e., terms that appear in all artifacts do not matter much)
3. the total number of terms that appear in both artifacts (e.g., 17 terms occur in both artifacts)
4. the total number of terms in an artifact (i.e., terms that appear in short artifacts matter more)

APPENDIX B – REFERENCES

Bjarnasson E., Runeson P, Borg M, Unterkalmsteiner M, Engström E, Regnell B, Sabaliauskaite G, Loconsole A, Gorschek T, and Feldt R. Challenges and Practices in Aligning Requirements with Verification and Validation: A Case Study of Six Companies, *Empirical Software Engineering*, 2013.

Borg M. Advancing Trace Recovery Evaluation: Applied Information Retrieval in a Software Engineering Context, *Licentiate thesis*, Lund University, 2012.

Borg M, Pfahl D, and Runeson P. Analyzing Networks of Issue Reports, *Proceedings of the 17th European Conference on Software Maintenance and Reengineering*, Genoa, Italy, 2013.

Borg M, and Pfahl D. Do Better IR Tools Improve the Accuracy of Engineers' Traceability Recovery?, In *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*, Lawrence, USA, 2011.

Borg M, Gotel O, and Wnuk K. Enabling Traceability Reuse for Impact Analyses: A Feasibility Study in a Safety Context, *Proceedings of the 7th International Workshop on Traceability in Emerging Forms of Software Engineering*, San Francisco, US, 2013.

Borg M, Runeson P, and Brodén L. Evaluation of Traceability Recovery in Context: A Taxonomy for Information Retrieval Tools, In *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering*, Ciudad Real, Spain, 2012.

Borg M. Findability through Traceability - A Realistic Application of Candidate Trace Links?, In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering*, Wroclaw, Poland, 2012.

Borg M, Wnuk K, and Pfahl D. Industrial Comparability of Student Artifacts in Traceability Recovery Research - An Exploratory Survey, In *Proceedings of the 16th European Conference on Software Maintenance and Reengineering*, Szeged, Hungary, 2012.

Borg M, and Runeson P. IR in Software Traceability: From a Bird's Eye View, In *Proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement*, Baltimore, US, 2013.

Borg M, Runeson P, and Ardö A. Recovering from a Decade: A Systematic Mapping of Information Retrieval Approaches to Software Traceability, *Empirical Software Engineering*, 2013.