Welcome from the Tool Demonstration Chairs

Holger M. Kienle
hkienle@acm.org
http://holgerkienle.wikispaces.com

Mircea Lungu
lungu@iam.unibe.ch
http://lungu.org/mircea

I. MOTIVATION

The Tool Demonstration Track at CSMR 2012 provides an opportunity for researchers to present novel research tools for software maintenance and reengineering. Eight tools which aim to aid software developers and maintainers in their work have been selected for presentation and demonstration. The tools are meant to be explored and applied by interested users and to inspire fellow researchers. Tool developers can thus get feedback on how to further improve upon their tools. This track also provides an outlet to publish about research embodied in tools. This was recognized by the track’s call:

Since tools are central to research in software maintenance and reengineering, tool demonstrations will have a prominent role within the conference and will add to the visibility of the associated research. Whereas a scientific paper is intended to give the background information and point out the scientific contribution of a new software engineering approach, the tool demonstration provides a good opportunity to show how the scientific approach has been transferred into a running tool prototype.

To further increase the visibility of the presented tools, developers were encouraged to provide a screencast and make it available on the web.

II. RELATED EFFORTS

Tool demonstrations tracks are by now an established event for many conferences related to software engineering. Examples closely related to this track are the tool demo tracks of ICSM, WCRE, ICSE and FSE. Tool demonstrations tracks are nevertheless considered complementary to the main technical track. One indication of this is that significantly less pages are allocated to the corresponding papers —typically, no more than four pages. For this track the upper limit is 6 pages (2 extra pages may be purchased), which enables authors to present their tool in more detail than is usually the case. We believe that the importance of tool demonstrations will increase further.

III. SELECTION PROCESS

For this track 8 out of 17 submissions have been accepted. Reviews were conducted by the track chairs and program committee, taking into account the submitted paper as well as the screencast or demonstration description. To reach the final decisions, papers received two or three reviews followed by discussions as needed.

The program committee, which helped in evaluating and providing feedback to the authors, was composed of international experts and sufficiently large to have no more than four papers per reviewer, thus allowing thorough, high-quality reviews. While this track does not give out a best tool demonstration award, we wish to acknowledge that the following submission received the highest marks by the reviewers:

BugMaps: A Tool for the Visual Exploration and Analysis of Bugs, by Andre Hora, Cesar Couto, Nicolas Anquentil, Marco Tulio Valente, Stephane Ducasse, Bhatti Usman and Julio Martins

By analyzing both the submissions that were accepted and the ones that were rejected, and in hoping to provide guidance to authors submitting to this track in the future, we would like to point out a number of features that in our experience make a paper a clear winner for the tool demonstrations track: a strong motivation of why the tool is relevant to the conference, a clear identification of novel tool features in relation to existing tools, a detailed description of the tool including screenshots and explanations, a well-informed identification of previously published related work.

IV. ACCEPTED TOOLS

Table I provides an overview of the accepted tools as of January 2012. Links to screencasts of the tools (if available) can be found at http://csmr2012.sed.hu/tooldemonstration.

One of the interesting observations that we made when looking at the submitted and accepted papers is that researchers are still interested in supporting programmers in their lower-level programming tasks: tools for visualization, debugging, static analysis and test coverage are closely related to code. Out of the 8 accepted demonstrations three are targeting the Eclipse platform showing that it is a good platform for maintenance tools.

In the following, each tools is briefly introduced.

Data from the image

---

Welcome from the Tool Demonstration Chairs

Holger M. Kienle
hkienle@acm.org
http://holgerkienle.wikispaces.com

Mircea Lungu
lungu@iam.unibe.ch
http://lungu.org/mircea

I. MOTIVATION

The Tool Demonstration Track at CSMR 2012 provides an opportunity for researchers to present novel research tools for software maintenance and reengineering. Eight tools which aim to aid software developers and maintainers in their work have been selected for presentation and demonstration. The tools are meant to be explored and applied by interested users and to inspire fellow researchers. Tool developers can thus get feedback on how to further improve upon their tools. This track also provides an outlet to publish about research embodied in tools. This was recognized by the track’s call:

Since tools are central to research in software maintenance and reengineering, tool demonstrations will have a prominent role within the conference and will add to the visibility of the associated research. Whereas a scientific paper is intended to give the background information and point out the scientific contribution of a new software engineering approach, the tool demonstration provides a good opportunity to show how the scientific approach has been transferred into a running tool prototype.

To further increase the visibility of the presented tools, developers were encouraged to provide a screencast and make it available on the web.

II. RELATED EFFORTS

Tool demonstrations tracks are by now an established event for many conferences related to software engineering. Examples closely related to this track are the tool demo tracks of ICSM, WCRE, ICSE and FSE. Tool demonstrations tracks are nevertheless considered complementary to the main technical track. One indication of this is that significantly less pages are allocated to the corresponding papers —typically, no more than four pages. For this track the upper limit is 6 pages (2 extra pages may be purchased), which enables authors to present their tool in more detail than is usually the case. We believe that the importance of tool demonstrations will increase further.

III. SELECTION PROCESS

For this track 8 out of 17 submissions have been accepted. Reviews were conducted by the track chairs and program committee, taking into account the submitted paper as well as the screencast or demonstration description. To reach the final decisions, papers received two or three reviews followed by discussions as needed.

The program committee, which helped in evaluating and providing feedback to the authors, was composed of international experts and sufficiently large to have no more than four papers per reviewer, thus allowing thorough, high-quality reviews. While this track does not give out a best tool demonstration award, we wish to acknowledge that the following submission received the highest marks by the reviewers:

BugMaps: A Tool for the Visual Exploration and Analysis of Bugs, by Andre Hora, Cesar Couto, Nicolas Anquentil, Marco Tulio Valente, Stephane Ducasse, Bhatti Usman and Julio Martins

By analyzing both the submissions that were accepted and the ones that were rejected, and in hoping to provide guidance to authors submitting to this track in the future, we would like to point out a number of features that in our experience make a paper a clear winner for the tool demonstrations track: a strong motivation of why the tool is relevant to the conference, a clear identification of novel tool features in relation to existing tools, a detailed description of the tool including screenshots and explanations, a well-informed identification of previously published related work.

IV. ACCEPTED TOOLS

Table I provides an overview of the accepted tools as of January 2012. Links to screencasts of the tools (if available) can be found at http://csmr2012.sed.hu/tooldemonstration.

One of the interesting observations that we made when looking at the submitted and accepted papers is that researchers are still interested in supporting programmers in their lower-level programming tasks: tools for visualization, debugging, static analysis and test coverage are closely related to code. Out of the 8 accepted demonstrations three are targeting the Eclipse platform showing that it is a good platform for maintenance tools.

In the following, each tools is briefly introduced.
### Table I
**ACCEPTED TOOLS**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Keywords</th>
<th>Availability</th>
<th>Platform/OS</th>
<th>License</th>
<th>Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BugMaps</td>
<td>Bug visualization, historical analysis, software quality</td>
<td>Download</td>
<td>Moose</td>
<td>MIT</td>
<td>Advanced prototype</td>
</tr>
<tr>
<td>ChEOPSJ</td>
<td>Testing, change-centric software development</td>
<td>Not public</td>
<td>Eclipse</td>
<td>–</td>
<td>Early prototype</td>
</tr>
<tr>
<td>Deterministic Replay in QEMU</td>
<td>Reverse engineering, dynamic analysis, virtual machines</td>
<td>Not public</td>
<td>Windows XP,7</td>
<td>–</td>
<td>Advanced prototype</td>
</tr>
<tr>
<td>Eclipse Array Explorer</td>
<td>Data structure visualization, visual debugging</td>
<td>Download</td>
<td>Eclipse</td>
<td>CC-BY</td>
<td>Mature</td>
</tr>
<tr>
<td>Eclipse Tracer</td>
<td>Visual debugging, program tracing, software visualization</td>
<td>Download</td>
<td>Eclipse</td>
<td>CC-BY</td>
<td>Advanced prototype</td>
</tr>
<tr>
<td>SECONDA</td>
<td>Evolution, software ecosystems, community</td>
<td>On request</td>
<td>Unix-like OS</td>
<td>Open source (TBD)</td>
<td>Early prototype</td>
</tr>
<tr>
<td>USE</td>
<td>OOSD, modeling, validation</td>
<td>Download</td>
<td>JDK 6</td>
<td>GNU GPL</td>
<td>Industrial strength</td>
</tr>
<tr>
<td>Web2MexADL</td>
<td>Documentation, maintenance</td>
<td>Download</td>
<td>JDK 6</td>
<td>GNU GPL v3</td>
<td>Advanced prototype</td>
</tr>
</tbody>
</table>

**BugMaps**<sup>1</sup> provides mechanisms to automate the process of retrieving and parsing software repositories to map bugs reported in bug-tracking platforms to defects in the classes and provides a set of interactive visualizations for reasoning about bugs. It supports software developers in answering questions such as: What are the modules involved in bug-fixing? What is the lifetime of a bug? What is the period that a module has presented more bugs?

**ChEOPSJ**<sup>2</sup> (Change and Evolution Oriented Programming Support for Java) sits in the background of Eclipse and can log the changes that a programmer makes to a software system. Additionally the changes and their dependencies can be used to find which tests are relevant for a given change. As such the feedback cycle for developer testing can be made much shorter.

**Deterministic Replay in QEMU**<sup>3</sup> is a whole-system non-intrusive deterministic replay of programs’ execution. It allows using deterministic replay for the purposes of reverse debugging and dynamic analysis (e.g. execution trace collection). Our tool supports deterministic replay for IA-32, IA-64, and ARM platforms.

**Eclipse Array Explorer**<sup>4</sup> allows developers to explore large arrays efficiently. The tabular view provided by this tool offers an overview of the values of all fields in all elements in the array. It is possible to expand inner objects to explore the values of their fields along with the outer fields and there are histograms and bar charts to examine the distributions of numerical of categorical values.

**Eclipse Tracer**<sup>5</sup> allows developers to adapt selected Eclipse break-points and watch-points into trace-points while debugging Java programs. Instead of stopping the execution, trace-points only record information about the current state when they are hit. The “silent” hits can be visualized over a time-line for post-mortem examination of the recorded information. Thus, the tool offers a solution in the middle between full tracing and stop-and-go debugging. Multiple traces can be visualized over time to reveal temporal relations.

**SECONDA**<sup>6</sup> (Software Ecosystem Analysis Dashboard) is a stand-alone tool for visualization and statistical analysis of (open source) software ecosystems and their development community, taking into account both product quality metrics and social aspects. Global analysis of the entire ecosystem as well as local analysis from the viewpoint of individual projects or developers is supported.

**Web2MexADL**<sup>7</sup> recovers software architecture from existing Java web systems supporting the MexADL verification approach. It helps to verify the maintainability of software systems. Key contributions are a probabilistic approach for the generation of architecture documentation and a flexible implementation based on machine-learning tools.

**USE**<sup>8</sup> (A UML-based Specification Environment) is a system for the specification of information systems. It is based on a subset of the Unified Modeling Language (UML). A model can be animated to validate the specification against non-formal requirements. USE can improve the model quality and the design of information systems at an early stage of the development process. The USE monitor plugin (an advanced prototype) allows to validate formal models of a monitored system against its implementation and can be used to detect differences between both.

## V. Conclusions

This year’s demonstration track was competitive with an acceptance rate just below 50%. The selected tools are focused on working with the code and promise interesting demonstrations and stimulating discussions. We hope you will enjoy reading the papers and, where available, watch the screencasts and try out the tools.

---

<sup>1</sup>http://rmod.lille.inria.fr/web/pier/software/BugMaps

<sup>2</sup>http://win.ua.ac.be/~qsoeten/other/cheopsj/

<sup>3</sup>http://www.ispras.ru

<sup>4</sup>http://www.cvast.tuwien.ac.at/projects/visualdebugging/ArrayExplorer

<sup>5</sup>http://www.cvast.tuwien.ac.at/projects/visualdebugging/EclipseTracer

<sup>6</sup>http://informatique.umons.ac.be/genlog/

<sup>7</sup>http://code.google.com/p/web2mexadl/

<sup>8</sup>http://www.db.informatik.uni-bremen.de/projects/use/