

Software Architecture Tool Demonstrations

Henry Muccini
Dipartimento di Informatica
University of L'Aquila, Italy
henry.muccini@di.univaq.it

Bedir Tekinerdogan
Department of Computer Engineering
Bilkent University, Ankara, Turkey
bedir@cs.bilkent.edu.tr

ABSTRACT

In this paper, we describe the short summary of the tool demonstrations session at WICSA/ECSA 2012. The session aimed to attract both tools in practice and research tools. We describe the targeted topics for the tool support, and report on the program.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architecture –Domain-Specific Architecture, Languages

General Terms

Management, Documentation, Design

Keywords

Software Architecture, Tools, Demo

1. INTRODUCTION

WICSA/ECSA is the premier conference of practitioners and researchers interested in the topics related to software architecture. Since its inception, WICSA/ECSA has functioned as a working conference, where researchers meet practitioners and where practicing software architects can explain the problems that they face in their day-to-day work and so influence the future of the field. Next to traditional conference keynotes and paper sessions WICSA/ECSA includes interactive working sessions where practitioners and researchers discuss their experiences to understand the current state of the field, and identify opportunities to make a difference in the future.

The WICSA/ECSA tool demonstrations track provides an opportunity for both practitioners and researchers to present and discuss the most recent advances, experiences, and challenges in the field of software architecture by means of live tool presentations.

Tools can belong in two distinct categories:

- *Tools used in practice*, either from commercial vendors or open-source projects. These demonstrations should focus on the practical applications within the different activities of software architecting and should particularly show how they advance the current state of the practice. New ideas and

features are particularly welcome.

- *Research tools* from academic or industrial research environments. These demonstrations are intended to highlight underlying scientific contributions and show how scientific approaches have been transferred into a working tool.

Both categories may range from early prototypes to in-house or pre-commercialized products.

Tool support for software architecture design can focus on different issues. To identify the topics we could relate to the concepts as defined in the ISO/IEC 42010 standard for architecture description [5]. The simplified model is shown in Figure 1.

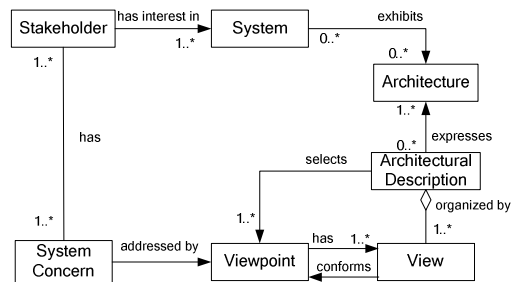


Figure 1. Conceptual model for Architectural Views based on ISO/IEEE/IEC Standard

Architecture is described based on stakeholder concerns. Each of the stakeholders' concerns impacts the early design decisions that the architect makes. Architecture is described using different architecture views according to the stakeholders' concerns. An architecture view is a representation of a set of system elements and relations associated with them to support a particular concern. Having multiple views helps to separate the concerns and as such support the modeling, understanding, communication and analysis of the software architecture for different stakeholders. Architecture views conform to viewpoints that represent the conventions for constructing and using a view. Architecture viewpoints are organized and structured within an architectural framework.

Based on this overview of concepts, among others, the following topics for tool support for software architecture design can be identified:

- Automated support for modeling concerns and traceability among concerns
- Automated support for tracing architectural elements to requirements/concerns

Copyright is held by author/owner(s).

WICSA/ECSA 2012, August 20-24, Helsinki, Finland.
ACM 978-1-4503-1568-5/12/08

- Automated support for tracing architectural elements to design/code
- Automated analysis of consistency among architecture views
- Generating code from design/code
- Automated support for checking the consistency among architecture and code
- Model-based testing based on architecture views
- Automated measurement of quality of architecture
- Automated support for collaborative architecture design
- Automated design rationale management

2. ORGANIZATION AND PROGRAM

Submissions of papers for tool demonstrations had to adhere to:

- the conference proceedings format at <http://www.wicsa.net/> and had to be submitted through the Easychair conference submission system.
- have a length of maximum 4 pages, describing the technology or approach, how it relates to other industrial or research efforts, including references, and what the expected benefits are.
- have an appendix (not included in the page count) that provides a brief description of how the presentation will be conducted, information on tool availability, maturity and the web-page for the tool (if one exists).
- state clearly the tool category: in-practice or research. In-practice tool demonstrations should also state whether they opt to be considered for inclusion in the proceedings.
- be accompanied by a video (with audio commentary) of maximum 5 minutes length and of high resolution (details of the tool's functionality should be clearly visible). The video should be uploaded on a video server (e.g. youtube) and the link should be mentioned in the submission.

For the tool demonstrations the following tool proposals were accepted for presenting the tool:

- Chris Chedghey, "Restructure101 – Improving the modularity of an existing codebase Tools Used in Practice" [2].

Summary: The paper aims to tackle complexity and evolution of code through using higher level architecture abstractions. The tool Restructure101 provides support to retrofit a hierarchical, acyclic module structure to an unstructured code-base, with minimal impact on the working code. For this the Levelized Structure Map, or LSM is used as a visual model of the code-base. The tool provides filtering and abstraction capabilities to support automatic analysis and interactive manipulation.

- Marcin Nowak and Cesare Pautasso, "Software Architecture Warehouse: live and collaborative architectural decision making" [3]

Summary: The paper presents the Software Architecture Warehouse, a tool for collaborative decision making support. The tool can be used both by individual designers as well as co-located and distributed design teams. Using the tool the designers can collaborate, build consensus and achieve shared understanding of the design issues. The tool is based on a decision meta-model that enables sharing and importing knowledge from different architectural decision management tools.

- Hyesun Lee, Jin-Seok Yang and Kyo C. Kang, "VULCAN: Architecture-Model-Based Software Development Workbench" [4]

Summary: The paper presents the workbench called VULCAN for supporting the feature-oriented software product line engineering life cycle [3]. The workbench consists of various tools that provides a set of architecture patterns based on which reusable components can be integrated. The integration of reusable the components can be automated using model-based generators.

More details about the tools can be found in the corresponding papers. Further, each of the tool presentations are also present as videos through the corresponding links [2][3][4].

The tool demonstrations have given a presentation that was scheduled into the conference program during the WICSA/ECSA industry day on August 22nd. In addition, there was also a demonstration area open to attendees at scheduled times during the conference, during which demonstrators are expected to be exhibiting their tools.

3. ACKNOWLEDGMENTS

We would like to thank the tool demo presenters and the participants to the demonstrations.

4. REFERENCES

- [1] WICSA/ECSA Tool Demonstrations, <http://wicsa2012.soberit.hut.fi/wp/assets/Call-for-Tool-Demos.pdf>, 2012.
- [2] Chris Chedghey, "Restructure101 – Improving the modularity of an existing codebase Tools Used in Practice", Short video (5 min): <https://vimeo.com/41214310>; Full tutorial (15 min): <https://vimeo.com/25631199>
- [3] Marcin Nowak and Cesare Pautasso, "Software Architecture Warehouse: live and collaborative architectural decision making", Video: <http://goo.gl/82mZ3>.
- [4] Hyesun Lee, Jin-Seok Yang and Kyo C. Kang, "VULCAN: Architecture-Model-Based Software Development Workbench", Video: <http://selab.postech.ac.kr/xe/?mid=selab>
- [5] [ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010). (identical to ANSI/IEEE Std1471–2000), July 2007.