OthelloPlay – A Plug-in Based Tool for Requirement Formalization and Validation

Roberto Cavada
Alessandro Cimatti
Andrea Micheli
Marco Roveri
Angelo Susi
Stefano Tonetta

Fondazione Bruno Kessler
{cimatti, amicheli, roveri, susi, tonettas, cavada}@fbk.eu

ABSTRACT

Requirement engineering is one of the most important phases in the development process of software and systems. In safety-critical applications, it is important to support the validation of the requirements with formal techniques to identify and remove flaws. However, requirements are often written in textual documents and their formalization and validation is not trivial for non-experts in formal methods. The goal of the OTHELLOPLAY tool is to support formalization of textual requirements and to simplify the use of formal techniques for requirements validation. The tool combines a formal verification engine and the Microsoft Word® editor in a single and consistent environment. A fundamental key in our design approach is a plug-in-based architecture, which uses the Python language in conjunction with a Microsoft Word® Add-In. The user can jump between textual requirements in the Microsoft Word® editor and the corresponding formal requirements model.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*Tools*

General Terms

Design, Verification

Keywords

Requirements formalization and validation, tools as plug-ins

1. OVERVIEW

Complex computer-based systems carry out critical functions in today's intelligent buildings, trains, cars, aircrafts, plant control or spacecrafts. Fundamental steps in the development process of such systems are the specification and validation of requirements. Formal approaches may help with a deep analysis through the precise semantics of the requirements. However, the formalization is typically a difficult task and it is often separated from the validation.

OTHELLOPLAY is a new tool, we are developing, to help users in the formalization of textual requirements in OTHELLO [5], a new language expressive enough to represent various domains of interest, yet allowing efficient reasoning capabilities. The tool incorporates techniques presented in a recent series of papers ([3], [6], and [4]). The work extends the results of the EURAILCHECK project [2] by overcoming some drawbacks of the related tool support [1] (e.g. the traceability and navigability of the linking between natural language requirements written in Microsoft Word® and their formal counterpart in OTHELLO).

The integration of the requisite collection and the formalization tasks within the Microsoft Word® editor was obtained by a hybrid plug-in-standalone approach: a standalone component handles the creation and validation of the formal model, and its linking with the textual requirements; a Microsoft Word® Add-In component supports the collection of the requirements from a document, each consisting of a fragment of text, a name and a type. Each requisite can be *linked* to a number of formal objects. The two components are designed to communicate in both directions in order to offer a seamless formalization procedure to the user. Once a formal model is composed, the user can proceed to the validation phase.

We implemented the Microsoft Word® Add-in in C# using the .NET framework and VSTO (Visual Studio® Tools for Office) . The standalone program is implemented in Python and GTK to ensure portability to other operating systems. For the communication and synchronization between the two components we use the COM technology that enables the interoperability between the Microsoft Word® Add-In and the designer GUI regardless the language used.

The main strength of OTHELLOPLAY is the tight integration with the Microsoft Word® editor: the formalization phase happens from inside the Microsoft Word® environment. The seamless extension of the editor's GUI allows the user to naturally learn how to use the tool and avoids to jump among various applications to interleave editing and formalization of the specifications. Having the formalization tool available while editing, also allows for an early analysis of partial specifications, thus pushing even further the early validation of requirements. In particular, the user can efficiently loops between requirement editing, formalization and validation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
TOPI '11, May 28, 2011, Waikiki, Honolulu, HI, USA
Copyright 2011 ACM 978-1-4503-0599-0/11/05 ...$10.00.

2. REFERENCES

[1] Cavada, R., Cimatti, A., Mariotti, A., Mattarei, C., Micheli, A., Mover, S., Pensallorto, M., Roveri, M., Susi, A., Tonetta, S.: Supporting requirements validation: The eurailcheck tool. In: ASE. pp. 665–667 (2009)

[2] Chiappini, A., Cimatti, A., Macchi, L., Rebollo, O., Roveri, M., Susi, A., Tonetta, S., Vittorini, B.: Formalization and validation of a subset of the European Train Control System. In: ICSE (2). pp. 109–118 (2010)

[3] Cimatti, A., Roveri, M., Susi, A., Tonetta, S.: Object models with temporal constraints. In: SEFM. pp. 249–258 (2008)

[4] Cimatti, A., Roveri, M., Susi, A., Tonetta, S.: Formalizing requirements with object models and temporal constraints. Journal of Software and Systems Modeling (2011)

[5] Cimatti, A., Roveri, M., Susi, A., Tonetta, S.: Validation of requirements for hybrid systems: a formal approach. Submitted to ACM Transactions on Software Engineering and Methodology (2011)

[6] Cimatti, A., Roveri, M., Tonetta, S.: Requirements validation for hybrid systems. In: CAV. pp. 188–203 (2009)