Change Management using CHIMPANC

Martin Ring[1] Jannis Stoppe[1,2] Rolf Drechsler[1,2]
[1]Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany
[2]Group of Computer Architecture, University of Bremen, 28359 Bremen, Germany

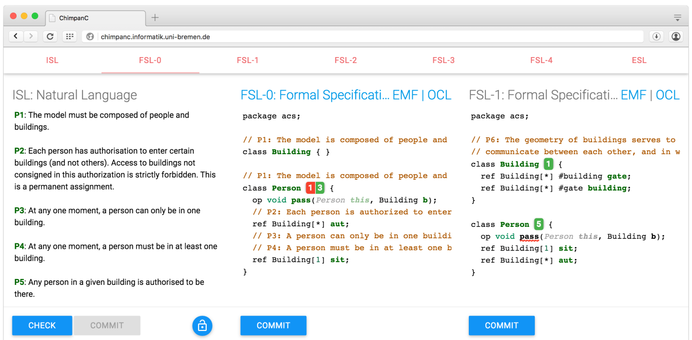

Fig. 1. The CHIMPANC user interface.

One approach to remedy the issue of increasing complexity in the hardware design process is to provide designers with more abstract languages that allow systems to be designed top-down, starting with an abstract model of the system and its requirements. Several of these languages are being used today. Natural language specifications are the most abstract form of describing a system, allowing the designers to use arbitrary language to explain how the system is supposed to behave and be structured. Formal modelling languages such as the UML are built upon a formal definition to avoid the issue of ambiguities in the description. System-level modelling language such as SystemC are the last step before synthesizer HDLs, allowing to build virtual prototypes that can be simulated without actually implementing in the final hardware design.

However, when following this approach, several new challenges arise: firstly, we have to keep the models in the different levels of abstraction *consistent* across the different languages and formalisms involved, secondly, we need a uniform notion of *refinement*, and thirdly, we want to be able to *track changes* and their impacts across the different levels of abstraction.

There are several independent approaches to change management for some of the individual specification levels we described. EMF itself for example offers a toolset to analyse differences between two models [3] and there are entire change management systems for UML [1]. However, these systems share several limitations, the foremost being that there are no semantic connections to external models taken into consideration, leaving the user without knowledge about impacts to other specification layers.

We propose the *Change Impact Analysis and Control Tool* (CHIMPANC) tool to handle these challenges. CHIMPANC extracts the relevant information from the models on the different levels and constructs mappings between them, thus allowing to check consistency and refinements, and moreover calculating the impact of changes. Thus, CHIMPANC ensures that *e.g.* a written specification or documentation is not made obsolete by changes in the implementation without being warned about it.

CHIMPANC is realised as a web interface and can thus either run locally or on a team server, configured for a specific system that is being developed. When users open the application in a browser they get presented a multi column layout representing the different specification layers (Fig. 1).

All extracted model elements are represented as bold identifiers. Mapped model elements appear green. When a user hovers the mouse over such a mapped element the corresponding refinement is visually emphasised (Fig. 2).

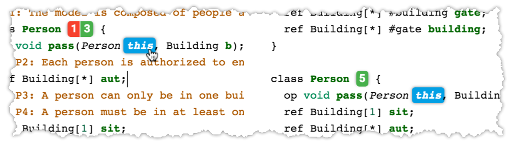

Fig. 2. Highlighting of mappings

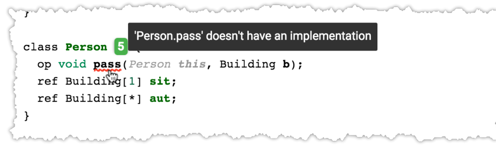

Fig. 3. Highlighting of inconsistencies

Inconsistencies are highlighted with red squiggly underlines. These include abstract models, attributes, references, operations and parameters which are unmapped in a refinement (Fig. 3) as well as mismatching mapped types and inconsistent multiplicities of references.

Content warnings are highlighted with orange squiggly underlines. These are currently only present in natural language where we automatically rate the quality of refinements, using the techniques from [2]. Again, a detailed description of the warning can be obtained by hovering the mouse over the marked element (Fig. 4).

Finally impact warnings appear as orange elements indicating that user attention is required (Fig. 5). An impact can either indicate that a refinement has changed or that the abstraction has been changed or removed.

REFERENCES

[1] Lionel C Briand, Yvan Labiche, and LO Sullivan. Impact analysis and change management of uml models. In *Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on*, pages 256–265. IEEE, 2003.

[2] Mathias Soeken, Nabila Abdessaied, Arman Allahyari-Abhari, Andi Buzo, Liana Musat, Georg Pelz, and Rolf Drechsler. Quality assessment for requirements based on natural language processing. In *Forum on Specification and Design Languages - Proceedings*. o.A., 2014.

[3] The Eclipse Foundation. EMF Diff/Merge. http://www.eclipse.org/diffmerge/, 2012. Accessed: 2015-10-28.

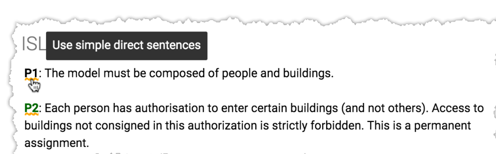

Fig. 4. A content warning in natural language

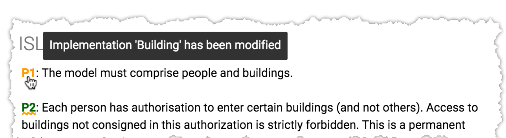

Fig. 5. An impact warning