

Natural Language Generation within Model-Driven Architecture

Introduction

There is often a gap between the requirements and the code of a system, both as in specification versus implementation [9, 3] but also in the knowledge needed to understand them [2]. One attempt to bridge the gap is given by Model-Driven Architecture [8, 6] where software models are used as an intermediate step. The idea is that the models should both formalise the requirements and explain the code.

Motivation

Testing the software models or accessing their contents requires an understanding of object-oriented design, knowledge about specific software models and experience of using tools for modelling software [2]. On the other hand, textual descriptions of the models are suitable for stakeholders without the necessary expertise in software models [4].

Aim

Natural language is a common way of sharing information, allowing stakeholders to contribute to the validation of the software models, resulting in software that better meets their requirements and expectations. By generating natural language descriptions of the structure and behaviour of the models we enable a process that can accommodate the input from the stakeholders incrementally with early and continuous feedback on the development process [1]. The model-to-text transformations will be done within the framework of Natural Language Generation [12].

We will use executable and translatable software models [5, 10] in our work. In contrast to UML [7] these models have all the properties that are necessary for automatic translation into efficient code [13]. But instead of translating the software models into code (or text) we will translate them into a linguistic model that takes care of ensuring texts that are grammatically correct. The linguistic model will be defined by Grammatical Framework [11].

References

- [1] S. Ambler. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [2] J. Arlow, W. Emmerich, and J. Quinn. Literate Modelling - Capturing Business Knowledge with the UML. In *Selected papers from the First International Workshop on The Unified Modeling Language UML'98: Beyond the Notation*, pages 189–199, London, UK, 1999. Springer-Verlag.
- [3] J. Bosch. Architecture in the age of compositionality. In M. Babar and I. Gorton, editors, *Software Architecture*, volume 6285 of *Lecture Notes in Computer Science*, pages 1–4. Springer Berlin, Heidelberg, 2010.
- [4] D. Firesmith. Modern requirements specification. *Journal of Object Technology*, 2(2):53–64, 2003.
- [5] S. J. Mellor and M. Balcer. *Executable UML: A Foundation for Model-Driven Architectures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [6] S. J. Mellor, S. Kendall, A. Uhl, and D. Weise. *MDA Distilled*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
- [7] OMG. OMG Unified Modeling Language (OMG UML) Infrastructure Version 2.3. <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF>. Accessed 11th September 2010.
- [8] OMG. MDA. <http://www.omg.org/mda/>, Accessed January 2011.
- [9] D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. *SIGSOFT Softw. Eng. Notes*, 17:40–52, October 1992.
- [10] C. Raistrick, P. Francis, J. Wright, C. Carter, and I. Wilkie. *Model Driven Architecture with Executable UMLTM*. Cambridge University Press, New York, NY, USA, 2004.
- [11] A. Ranta. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford, 2011.
- [12] E. Reiter and R. Dale. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3:57–87, March 1997.
- [13] T. Siljamäki and S. Andersson. Performance benchmarking of real time critical function using BridgePoint xtUML. NW-MoDE'08: Nordic Workshop on Model Driven Engineering. Reykjavik, Iceland, August 2008.