Abstract

Over the last three decades, an increasing number of languages used for designing and developing software have been created. Software developers gain the benefits of combining multiple programming languages and paradigms in application development, as a result the so-called language engineering approach can be outlined. It involves Domain-Specific Languages (DSLs) and automatic code generation. This paper offers a brief review of the use of DSL as a modeling and programming language and it tight connection with automatic code generation. The evolution of the developed software product requires evolution of the domain-specific language as well. Some of the risks of abandoning of DSLs during development are discussed.

References

- Bentley J. 1986 Little languages. CACM, 29(8), 711–721.
- Evans, E.  2003 Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional
- Debasish, G.  2011 DSLs in Action, Manning Publications
- Fowler M.  2010 Domain Specific Languages, Addison-Wesley Professional
- Fowler M. , Blog http://www. martinfowler. com/bliki/ModelDrivenArchitecture. html (visited 05. 02. 2015)
- Warmer, J. B. , Kleppe, A. G.  2006 Building a Flexible Software Factory Using Partial Domain Specific Models, In: Sixth OOPSLA Workshop on Domain-Specific Modeling (DSM&amp;apos;06)
- Cemosem, G. , Naiburg, E.  2004 The Value of Modelling.  A Technical Discussion of Software Modeling, IBM

**Index Terms**

Computer Science

Information Sciences

**Keywords**

Metaprogramming  DSL  code generation  language engineering  modeling.