SLECOURSE: Annotated Bibliography[*]

October 13, 2012

Abstract

The purpose of this annotated bibliography is to collect papers that could serve as background for concrete SLE courses. At this early stage of a bibliography, we may be well advised on focusing on more general papers that are likely to be useful for different SLE course designs as opposed to highly technical and specialized work. Should it happen that the bibliography contains confusingly many entries, then we can still impose some extra structure on the bibliography, e.g., based on grouping or tagging.

References

[1] Sebastian Erdweg, Tillmann Rendel, Christian Kästner, and Klaus Ostermann. Sugarj: library-based syntactic language extensibility. In *Proceedings of the 26th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2011, part of SPLASH 2011, Portland, OR, USA, October 22 - 27, 2011*, pages 391–406. ACM, 2011. Entry added by *Ralf Lämmel*. Paper available online at *http://www.informatik.uni-marburg.de/~rendel/erdweg11sugarj.pdf*..

This paper describes SugarJ—an approach to enhance Java's library notion with syntactic sugar, hence providing an easy-to-use language extension approach. For instance, XML, Java

[*] http://slecourse.github.com/slecourse

> closures, and XML schemas can be integrated into the Java language in this way. SugarJ leverages SDF for syntax definitions, Stratego for the underlying transformation framework. SugarJ provides an incremental parsing approach so that library imports can affect parsing past the import.

[2] Martin Erwig and Eric Walkingshaw. Semantics first! - rethinking the language design process. In *Software Language Engineering - 4th International Conference, SLE 2011, Braga, Portugal, July 3-4, 2011, Revised Selected Papers*, volume 6940 of *Lecture Notes in Computer Science*, pages 243–262. Springer, 2012. Entry added by *Ralf Lämmel*. Paper available online at *http://web.engr.oregonstate.edu/~erwig/papers/SemanticsFirst_SLE11.pdf*..

> This paper suggests a semantics-centric approach to language design as opposed to a more syntax-based one. Haskell is used as a metalanguage. General language operators are employed to adapt and grow sophisticated languages out of simple semantics concepts.

[3] Jean-Marie Favre, Dragan Gasevic, Ralf Lämmel, and Andreas Winter. Guest editors' introduction to the special section on software language engineering. *IEEE Trans. Software Eng.*, 35(6):737–741, 2009. Entry added by *Ralf Lämmel*. Paper available online at *http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5353438*..

> This is not a technical paper, but rather an extended introduction to a special issue on SLE. This text provides a relatively early but somewhat matured description of the notions SLE and software languages. Also, the papers of the special issue provide a good test harness for discussing these notions.

[4] Markus Herrmannsdoerfer, Sander Vermolen, and Guido Wachsmuth. An extensive catalog of operators for the coupled evolution of metamodels and models. In *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pages 163–182. Springer, 2011. Entry added by *Ralf Lämmel*. Paper available

online at http://www.st.ewi.tudelft.nl/~vermolen/pmwiki/uploads/Main/HerrmannsdoerferVW10.pdf..

> This paper deals with the coupled evolution of metamodels and models (in a MOF/EMF-like context). Such coupling has been studied intensively by the SLE and MDE communities and the present paper aims at a particularly "complete" catalog of operators for coupled transformation.

[5] Ralf Lämmel and Wolfram Schulte. Controllable combinatorial coverage in grammar-based testing. In *Testing of Communicating Systems, 18th IFIP TC6/WG6.1 International Conference, TestCom 2006, New York, NY, USA, May 16-18, 2006, Proceedings*, volume 3964 of *Lecture Notes in Computer Science*, pages 19–38. Springer, 2006. Entry added by *Ralf Lämmel*. Paper available online at http://homepages.cwi.nl/~ralf/ccc/paper.pdf..

> This paper describes a highly systematic and scalable approach to grammar-based testing. The grammar is essentially interpreted in a combinatorial sense and a number of well-defined control mechanisms help with deriving test data of the intented shape, size, and number while also avoiding combinatorial explosion. The approach has been implemented in C# and used for grammar-based testing in the .NET platform. Combinatorial explorations constrasts with randomized test-data generation. This paper provides a broad discussion of related work.

[6] Paul Klint and Ralf Lämmel and Chris Verhoef. Toward an engineering discipline for grammarware. *ACM Trans. Softw. Eng. Methodol.*, 14(3):331–380, 2005. Entry added by *Ralf Lämmel*. Paper available online at http://www.cs.vu.nl/grammarware/agenda/paper.pdf..

> This paper surveys SLE (or grammarware engineering) by making an inventory of grammarware, describing some problems with the proper engineering of grammarware, stating some promises of addressing grammarware engineering more seriously, listing some matured or emerging principles of grammarware engineering, and calling out a list of research challenges.

[7] Lukas Renggli, Tudor Gîrba, and Oscar Nierstrasz. Embedding languages without breaking tools. In *ECOOP 2010 - Object-Oriented Programming, 24th European Conference, Maribor, Slovenia, June 21-25, 2010. Proceedings*, volume 6183 of *Lecture Notes in Computer Science*, pages 380–404. Springer, 2010. Entry added by *Ralf Lämmel*. Paper available online at *http://scg.unibe.ch/archive/papers/Reng10aEmbeddingLanguages.pdf*..

> This paper describes an embedding approach for the implementation of domain-specific languages (DSLs). Specifically, DSLs are modeled as language extensions of the underlying host language. The approach addresses the challenge of providing the language extensions in a manner that they integrate well with the development tools of the host language. The paper presents the extensible system Helvetia which intercepts the compilation pipeline of the Smalltalk host language to seamlessly integrate language extensions.

[8] Davide Di Ruscio, Ralf Lämmel, and Alfonso Pierantonio. Automated co-evolution of gmf editor models. In *Software Language Engineering - Third International Conference, SLE 2010, Eindhoven, The Netherlands, October 12-13, 2010, Revised Selected Papers*, volume 6563 of *Lecture Notes in Computer Science*, pages 143–162. Springer, 2011. Entry added by *Ralf Lämmel*. Paper available online at *http://userpages.uni-koblenz.de/~laemmel/gmfco/paper.pdf*..

> This paper studies coupled evolution in the context of Eclipse Modeling Framework (EMF) and the associated Graphical Modeling Framework (GMF). In this context, we face several coupled models; if the domain model changes, then several other models must co-change. The problem is interesting because of the heterogeneity of the involved models.

[9] Emin Gün Sirer and Brian N. Bershad. Using production grammars in software testing. In *Proceedings of the Second Conference on Domain-Specific Languages (DSL '99), Austin, Texas, USA, October 3-5, 1999*, pages 1–13. ACM, 1999. Entry added by *Ralf Lämmel*. Paper available online at *http://www.cs.cornell.edu/people/egs/papers/kimera-dsl99.pdf*..

This paper shows how grammar-based test-data generation and an accompanying methodology of testing may be highly effective and scalable for testing language-based software. Previously, grammar-based testing was mainly focused on compiler testing. The paper specifically tests the Java virtual machine.

[10] Laurence Tratt. Domain specific language implementation via compile-time meta-programming. *ACM Trans. Program. Lang. Syst.*, 30(6), 2008. Entry added by *Ralf Lämmel*. Paper available online at http://eprints.mdx.ac.uk/5920/1/Tratt-domain_specifci_language_implementation.pdf..

This paper describes an embedding approach for the implementation of domain-specific languages (DSLs). The approach is based on the Converge programming language and uses its compile-time meta-programming facility. The accompanying methodoogy is meant to simplify DSL implementations and improve their quality.