

6 References

ALDRICH, J. **ArchJava: Connecting Software Architecture to Implementation**. In Proceedings of the 24th International Conference on Software Engineering - ICSE '02, pags. 187-197. New York, NY, 2002.

ASPECTJ. Available at <http://www.eclipse.org/aspectj/>.

BUSCHMANN, F. et al. **Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing**. In Wiley Software Patterns Series, John Wiley & Sons. 2007.

CERI, S. et al. **What you always wanted to know about datalog**. In IEEE Transactions on Knowledge and Data Engineering, pags. 146-166, vol. 1, no. 1. 1989.

CLEMENTS, P., BACHMANN, F., BASS, L., GARLAN, D., IVERS, J., LITTLE, R., MERSON, P., NORD, R., STAFFORD, J. **Documenting Software Architectures: Views and Beyond**. 2 ed. Addison-Wesley Professional. 2010.

CRUZ, S. and LUCENA, C. **Identificando objetos através de pronome**. 125 f. Tese (Doutorado) - Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, RJ, Brazil. 2003.

EICHBERG, M., KLOPPENBURG, S., KLOSE, K. and MEZINI, M. **Defining and continuous checking of structural program dependencies**. In Proceedings of the 30th International Conference on Software engineering - ICSE '08, pags. 391-400. New York, NY, United States. 2008.

DANTAS, F., GURGEL, A., GARCIA, A. **Towards a Suite of Metrics for Advanced Composition Mechanisms**. In Proceedings of the 2nd International Workshop on Empirical Evaluation of Software Composition Techniques - ESCOT 2011 at ECOOP'11. Lancaster, United Kingdom. 2011.

FIGUEIREDO, E. et al. **Evolving software product lines with aspects: An empirical study on design stability**. In Proceedings of the 30th International Conference on Software Engineering - ICSE '08, pags. 261-270. New York, NY, United States. 2008.

FIGUEIREDO, E. et al. **AJATO: an AspectJ Assessment Tool**. In proceedings of the 20th European Conference on Object-Oriented Programming - ECOOP'06, demo session D9. Nantes, France. 2006.

FOWLER, M. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley. 1999.

GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 1995.

GARCIA, J. et al. **Identifying architectural bad smells**. In Proceedings of the European Conference on Software Maintenance and Reengineering - CSMR '09, pags. 255-258. Washington, DC, United States. 2009.

GARLAN, D. and SHAW, M. **An Introduction to Software Architecture**. Advances in Software Engineering and Knowledge Engineering, pags. 1-39, Singapore. 1993.

GARLAN, D. et al. **Acme: an architecture description interchange language**. In Proceedings of the CASCON'97. pags. 169-183. Toronto, Ontario, Canada. 1997.

GREENWOOD, P. et al. **On the impact of aspectual decompositions on design stability: An empirical study**. In Proceedings of the 21st Conference of Object-Oriented Programming. - ECOOP'07. Springer-Verlag. pags. 176-200. 2007.

GURGEL, A., MACIA, I., GARCIA, A., MEZINI, M., EICHBERG, M., VON STAA, A., MITSCHKE, R. **TamDera: Blending and Reusing Rules for Architectural Degradation Prevention**; In Proceedings of 20th Symposium on the Foundations of Software Engineering (FSE'12). 2012. (in submission).

GURGEL, A., DANTAS, F., GARCIA, A. **Um Estudo de Composições de Padrões de Projeto em CaesarJ**. In Proceedings of the IV Latin American Workshop on Aspect-Oriented Software Development - LA-WASP 2010. pags. 30-36. 2010.

GURGEL, A., DANTAS, F., GARCIA, A. **On-Demand Integration of Product Lines: A Study of Reuse and Stability**. In Proceedings of the 2nd International Workshop on Product Line Approaches in Software Engineering - PLEASE '11 at ICSE'11, 2011; pags 35-39. 2011.

HEALTHWATCHER. Available at <http://www.comp.lancs.ac.uk/~greenwop/tao/>. 2012.

HOCHSTEIN, L. and LINDVALL, M. **Combating architectural degeneration: A survey**. Inf. Softw. Technol. 47. July, 2005.

KNODEL, J. and POPESCU, D. **A Comparison of Static Architecture Compliance Checking Approaches**. In the Proceedings of the 2007 Working IEEE/IFIP Conference on Software Architecture - WICSA'07, pags. 12-12. Mumbai, India. 2007.

LANZA, M. and MARINESCU, R. **Object-Oriented Metrics in Practice**. In Springer-Verlag. New York, NY, United States. 2006.

LI, Z. **Characterizing and Diagnosing Architectural Degeneration of Software Systems from Defect Perspective**. In Electronic Thesis and Dissertation Repository. <http://ir.lib.uwo.ca/etd/30>. 2010.

MACIA, I., GARCIA, J., POPESCU, D., GARCIA, A., MEDVIDOVIC, N. and STAA, A. **Are Automatically-Detected Code Anomalies Relevant to Architectural Modularity?** An Exploratory Analysis of Evolving Systems. In Proceedings of the 11st international conference on Aspect-oriented software development companion - AOSD'12, pags. 167-178. Postdam, Germany. March, 2012.

MACIA, I. et al. **On the Relevance of Code Anomalies for Identifying Architecture Degradation Symptoms**. In Proceedings of the 16th European

Conference on Software Maintenance and Reengineering - CSMR'12. Szeged, Hungary. 2012 (to appear).

MACCORMACK, A., RUSNAK, J. and BALDWIN, C. Exploring **the structure of complex software designs: An empirical study of open source and proprietary code**. In Management Science, 52(7), pags.1015-1030. 2006.

MALEK, S. et al. **Reconceptualizing a family of heterogeneous embedded systems via explicit architectural support**. In Proceedings of the 29th In Proceedings of the 30th International Conference on Software Engineering - ICSE'07, pags. 591-601. 2007.

MARA, L. et al. **Hist-Inspect: A Tool for History-Sensitive Detection of Code Smells**. In Proceedings of the tenth international conference on Aspect-oriented software development companion - AOSD '11, pags. 65-66. 2011.

MARINESCU, R. **Detection strategies: Metrics-based rules for detecting design flaws**. In Proceedings of the 20th International Conference on Software Maintenance - ICSM'04, pags. 350-359. 2004.

MARINESCU, R. et al. **iPlasma: An integrated platform for quality assessment of object-oriented design**. In Proceedings of the 21th International Conference on Software Maintenance - ICSM'05, pags. 77-80. 2005.

MARTIN, R. **Agile Software Development, Principles, Patterns, and Practices**. Prentice Hall. 2002.

MARWAN A. and ALDRICH, J. **Static Extraction and Conformance Analysis of Hierarchical Runtime Architectural Structure using Annotations**. In Proceedings of the 24th international conference on Object-Oriented Programming, Systems, Languages and Applications - OOPSLA'09, pags. 321-340. 2009.

MEDVIDOVIC, N. and RICHARD, T. **A classification and comparison framework for software architecture description languages**. IEEE Transactions on Software Engineering, pags. 70-93, vol. 26, no. 1. 2000.

MERKLE, B. **Stop the software architecture erosion: building better software systems**. In Proceedings of the ACM international conference companion on OOP systems languages and applications companion - SPLASH '10. New York, NY, United States, pags. 129-138. 2010.

MOHA, N. et al. 2010. **DECOR: A Method for the Specification and Detection of Code and Design Smells**. In IEEE Trans. Soft. Eng. 36, pags. 20-36. January, 2010.

MORGAN, C. **A static aspect language of checking design rules**. In Proceedings of the 6th international conference on Aspect-oriented software development companion, pags. 63-72. 2007.

OLIVEIRA, M. **PREViA: An Approach for Visualizing the Evolution of Software Models**. Master Thesis. COPPE/UFRJ, 185p. 2011.

PERRY, D. and WOLF, A. **Foundations for the study of software architecture**. SIGSOFT Software. Eng. Notes 17.pags. 40-52. 1992.

PMD. Available at <http://pmd.sourceforge.net/>.2012.

SANGAL, N. et al. **Using dependency models to manage complex software architecture.** In Proceedings of the 20th on Object-Oriented Programming, Systems, Languages and Applications - OOPSLA'05, pages. 167-176 . 2005.

SEMMLE CODE. Available at <http://semml.com/semmlcode/>. 2012.

SULLIVAN, K., GRISWOLD, W., CAI, Y. and HALLEN, B. **The structure and value of modularity in software design.** In 9th International Symposium on Foundations of Software Engineering - FSE. pages. 99-108. Vienna, Austria. 2001

SOARES, S. et al. **Implementing distribution and persistence aspects with AspectJ.** In Proceedings of the 17th on Object-Oriented Programming, Systems, Languages and Applications - OOPSLA'02, pages. 174-190, 2002.

SWI-PROLOG. Available at <http://www.swi-prolog.org>. 2012.

TAMDERA. Available at www.les.inf.puc-rio.br/opus/tamdera/fse12/. 2012.

TAYLOR, R., MEDVIDOVIC, N. and DASHOFY, E. **Software Architecture: Foundations, Theory, and Practice.** Wiley Publishing. 2009.

TERRA, R. and VALENTE, T. **A Dependency Constraint Language to Manage Object-Oriented Software Architectures.** In Software: Practice and Experience, v. 32, n. 12, John Wiley & Sons. 2009, pages. 1073-1094.

TOGETHER. Available at <http://www.borland.com/us/products/together/>. 2012.

UBAYASHI, N. et al. **Archface: A Contract Place Where Architectural Design and Code Meet Together.** In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10, pages 75-84. New York, NY, United States. 2010.

TROPASHKO, V. and BURLESON, D. **SQL Design Patterns: Expert Guide to SQL Programming.** Rampant TechPress. 2007.

VERBAERE, M., GODFREY, M. and GIRBA, T. **Query Technologies and Applications for Program Comprehension.** In Proceedings of the 16th IEEE Int'l Conference on Program Comprehension - ICPC'08. IEEE CS Press. pages. 185-288. 2008.

WERNER, C. et al. **Evoltrack: A Plug-in-Based Infrastructure for Visualizing Software Evolution.** In Proceedings of the 1st Brazilian Workshop on Software - WBVS. São Paulo, SP, Brazil. 20011.

XTEXT. <http://www.eclipse.org/Xtext/>. 2012.

Appendix A

BNF TamDera Grammar

The BNF description uses the bold font to display terminal symbols and the first characteres of non-terminal symbols are shown in upper-case format. The symbols '[A]', '(A)+' and '()A'*' respectively impose the cardinalities: optional (0 or 1); at least one; and zero or more to a the symbol A.

ConceptDeclaration	::=	concept ConceptId [ConceptInheritance] { [ConceptMapping] [ThresholdVariableList] (AntiDriftRule)* (AssignmentThreshold)* }
ConceptMapping	::=	name: STRING
	::=	parent: STRING
AntiDriftRule	::=	Metric Operator (Value VariableId)
	::=	ConstraintSetId
Metric	::=	LOC CBO LOCM CC DIT ...
Operator	::=	> < = ≤ ≥
AntiErosionRule	::=	only ConceptList can -DependType ConceptList
	::=	ConceptList cannot - DependType ConceptList
	::=	ConceptList must -DependType ConceptList
CptList	::=	ConceptId (, ConceptId)*
DependType	::=	derive invoke depend create declare handle
ConceptInheritance	::=	extends ConceptId
ConstraintSetDecl	::=	constraintset ConstraintSetId { (AntiDriftRule)+ }
CptList	::=	ConceptId (, ConceptId)*
ThresholdVariableList	::=	thresholds: VariableId (, VariableId)*
AssignmentThreshold	::=	assign VariableId to VALUE
ConceptId	::=	STRING
ConstraintSetId	::=	STRING
VariableId	::=	STRING
Value	::=	NUMBER