1.	Record Nr.	UNINA9910483137003321
	Titolo	Software Language Engineering : Second International Conference, SLE 2009, Denver, CO, USA, October 5-6, 2009 Revised Selected Papers / / edited by Mark van den Brand, Dragan Gasevic, Jeff Gray
	Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2010
	ISBN	1-280-38595-2 9786613563873 3-642-12107-1
	Edizione	[1st ed. 2010.]
	Descrizione fisica	1 online resource (390 p. 126 illus.)
	Collana	Programming and Software Engineering, , 2945-9168 ; ; 5969
	Classificazione	DAT 310f DAT 350f SS 4800
	Altri autori (Persone)	BrandMark van den GasevicDragan GrayJeff (Jeffrey G.)
	Disciplina	005.1
	Soggetti	Software engineering Compilers (Computer programs) Computer simulation Electronic data processing - Management Computer science Software Engineering Compilers and Interpreters Computer Modelling IT Operations Computer Science Logic and Foundations of Programming
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Bibliographic Level Mode of Issuance: Monograph
	Nota di bibliografia	Includes bibliographical references and index.
	Nota di contenuto	Keynotes -- Eating Our Own Dog Food: DSLs for Generative and Transformational Engineering -- If MDE Is the Solution, Then What Is the Problem? -- Regular Papers -- Language Evolution in Practice: The History of GMF -- A Novel Approach to Semi-automated Evolution of DSML Model Transformation -- Study of an API Migration for Two XML

APIs -- Composing Feature Models -- VML* – A Family of Languages for Variability Management in Software Product Lines -- Multi-view Composition Language for Software Product Line Requirements -- Yet Another Language Extension Scheme -- Model Transformation Languages Relying on Models as ADTs -- Towards Dynamic Evolution of Domain Specific Languages -- ScalaQL: Language-Integrated Database Queries for Scala -- Integration of Data Validation and User Interface Concerns in a DSL for Web Applications -- Ontological Metamodeling with Explicit Instantiation -- Verifiable Parse Table Composition for Deterministic Parsing -- Natural and Flexible Error Recovery for Generated Parsers -- PIL: A Platform Independent Language for Retargetable DSLs -- Graphical Template Language for Transformation Synthesis -- A Role-Based Approach towards Modular Language Engineering -- Language Boxes -- Declarative Scripting in Haskell -- An Automated Process for Implementing Multilevel Domain Models -- Domain-Specific Metamodelling Languages for Software Language Engineering -- Generating Smart Wrapper Libraries for Arbitrary APIs -- Closing the Gap between Modelling and Java.

Sommario/riassunto

We are pleased to present the proceedings of the Second International Conf- ence on Software LanguageEngineering (SLE 2009). The conference was held in Denver, Colorado (USA) during October 5-6, 2009 and was co-located with the th 12 IEEE/ACM International Conference on Model-Driven Engineering L- th guages and Systems (MODELS 2009) and the 8 ACM International Conf- ence on Generative Programming and Component Engineering (GPCE 2009). TheSLEconferenceseriesisdevotedtoawiderangeoftopicsrelatedtoarti'cial languages in software engineering. SLE is an international research forum that brings together researchers and practitioners from both industry and academia to expand the frontiers of software language engineering. SLE'sforemostmissionis to encourageand organizecommunicationbetween communities that have traditionally looked at software languagesfrom di'erent, more specialized, and yet complementary perspectives. SLE emphasizes the f- damental notion of languages, as opposed to any realization in speci'c technical spaces. In this context, the term "software language" comprises all sorts of - ti'cial languages used in software development, including general-purpose p- gramming languages, domain-speci'c languages, modeling and meta-modeling languages, data models, and ontologies. Software language engineering is the application of a systematic, disciplined, quanti'able approach to the devel- ment,use, andmaintenanceoftheselanguages. TheSLEconferenceisconcerned with all phases of the lifecycle of software languages; these include the design, implementation, documentation, testing, deployment, evolution, recovery, and retirement of languages. Of special interest are tools, techniques, methods, and formalisms that support these activities. In particular, tools are often based on, or automatically generated from, a formal description of the language.