

1. Record Nr.	UNISA996466271003316
Autore	Jorges Sven
Titolo	Construction and Evolution of Code Generators [[electronic resource] ] : A Model-Driven and Service-Oriented Approach // edited by Sven Jörges
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2013
ISBN	3-642-36126-9
Edizione	[1st ed. 2013.]
Descrizione fisica	1 online resource (XXIV, 246 p. 75 illus.)
Collana	Programming and Software Engineering ; ; 7747
Classificazione	SS 4800
Disciplina	005.4/5
Soggetti	Programming languages (Electronic computers) Software engineering Application software Programming Languages, Compilers, Interpreters Software Engineering Computer Appl. in Administrative Data Processing
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	The state of the art in code generation -- Extreme model-driven development and jABC -- The Genesys framework -- Code generators for jABC -- Verification and validation of code generators -- Domain-specific code generators for EMF -- Service-oriented combination of code generation frameworks.
Sommario/riassunto	Automatic code generation is an essential cornerstone of model-driven approaches to software development. Currently, lots of techniques are available that support the specification and implementation of code generators, such as engines based on templates or rule-based transformations. All those techniques have in common that code generators are either directly programmed or described by means of textual specifications. This monograph presents Genesys, a general approach, which advocates the graphical development of code generators for arbitrary source and target languages, on the basis of models and services. In particular, it is designed to support incremental language development on arbitrary metalevels. The use of models

allows building code generators in a truly platform-independent and domain-specific way. Furthermore, models are amenable to formal verification methods such as model checking, which increase the reliability and robustness of the code generators. Services enable the reuse and integration of existing code generation frameworks and tools regardless of their complexity, and at the same time manifest as easy-to-use building blocks which facilitate agile development through quick interchangeability. Both, models and services, are reusable and thus form a growing repository for the fast creation and evolution of code generators.

---