1.	Record Nr.	UNINA9910418322703321
	Autore	Horn Tassilo
	Titolo	A functional, comprehensive and extensible multi-platform querying and transformation approach / / Tassilo Horn
	Pubbl/distr/stampa	Berlin/Germany, : Logos Verlag Berlin, 2016 Berlin, Germany : , : Logos Verlag Berlin GmbH, , [2016] ©2016
	Descrizione fisica	1 online resource (xv, 454 pages) : illustrations; digital file(s)
	Disciplina	005.1028
	Soggetti	Computer science
	Lingua di pubblicazione	Tedesco
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Thesis approved by the doctoral committee of the Faculty 4: Computer Science at the University Koblenz-Landau in fulfillment with the requirements for obtaining the degree Doktor der Naturwissenschaften (Dr. rer. nat.). -- Title-page verso.
	Nota di bibliografia	Includes bibliographical references and index.
	Sommario/riassunto	This thesis is about a new model querying and transformation approach called FunnyQT which is realized as a set of APIs and embedded domain-specific languages (DSLs) in the JVM-based functional Lisp-dialect Clojure. Founded on a powerful model management API, FunnyQT provides querying services such as comprehensions, quantified expressions, regular path expressions, logic-based, relational model querying, and pattern matching. On the transformation side, it supports the definition of unidirectional model-to-model transformations, of in-place transformations, it supports defining bidirectional transformations, and it supports a new kind of co-evolution transformations that allow for evolving a model together with its metamodel simultaneously. Several properties make FunnyQT unique. Foremost, it is just a Clojure library, thus, FunnyQT queries and transformations are Clojure programs. However, most higher-level services are provided as task-oriented embedded DSLs which use Clojure's powerful macro-system to support the user with tailor-made language constructs important for the task at hand. Since queries and

transformations are just Clojure programs, they may use any Clojure or Java library for their own purpose, e.g., they may use some templating library for defining model-to-text transformations. Conversely, like every Clojure program, FunnyQT queries and transformations compile to normal JVM byte-code and can easily be called from other JVM languages. Furthermore, FunnyQT is platform-independent and designed with extensibility in mind. By default, it supports the Eclipse Modeling Framework and JGraLab, and support for other modeling frameworks can be added with minimal effort and without having to modify the respective framework's classes or FunnyQT itself. Lastly, because FunnyQT is embedded in a functional language, it has a functional emphasis itself. Every query and every transformation compiles to a function which can be passed around, given to higher-order functions, or be parametrized with other functions.