Extremo

Automated modelling assistance by integrating heterogeneous information sources

. Á. Mora Segura[1] J. de Lara[1]
P. Neubauer[2] M. Wimmer[2]

[1]Modelling and Software Engineering Research Group
Universidad Autónoma de Madrid

[2]CDL-MINT, Business Informatics Group
TU Wien

2018

Table of Contents

Looking for a way to reuse existing knowledge

Model-Driven Engineering (MDE) uses models as its main assets in the software development process.
High quality models are pivotal for the success of MDE.
Nevertheless, they are mostly built in an unassisted way, with no mechanisms for reusing existing knowledge.

Metamodelling expert vs Domain Expert

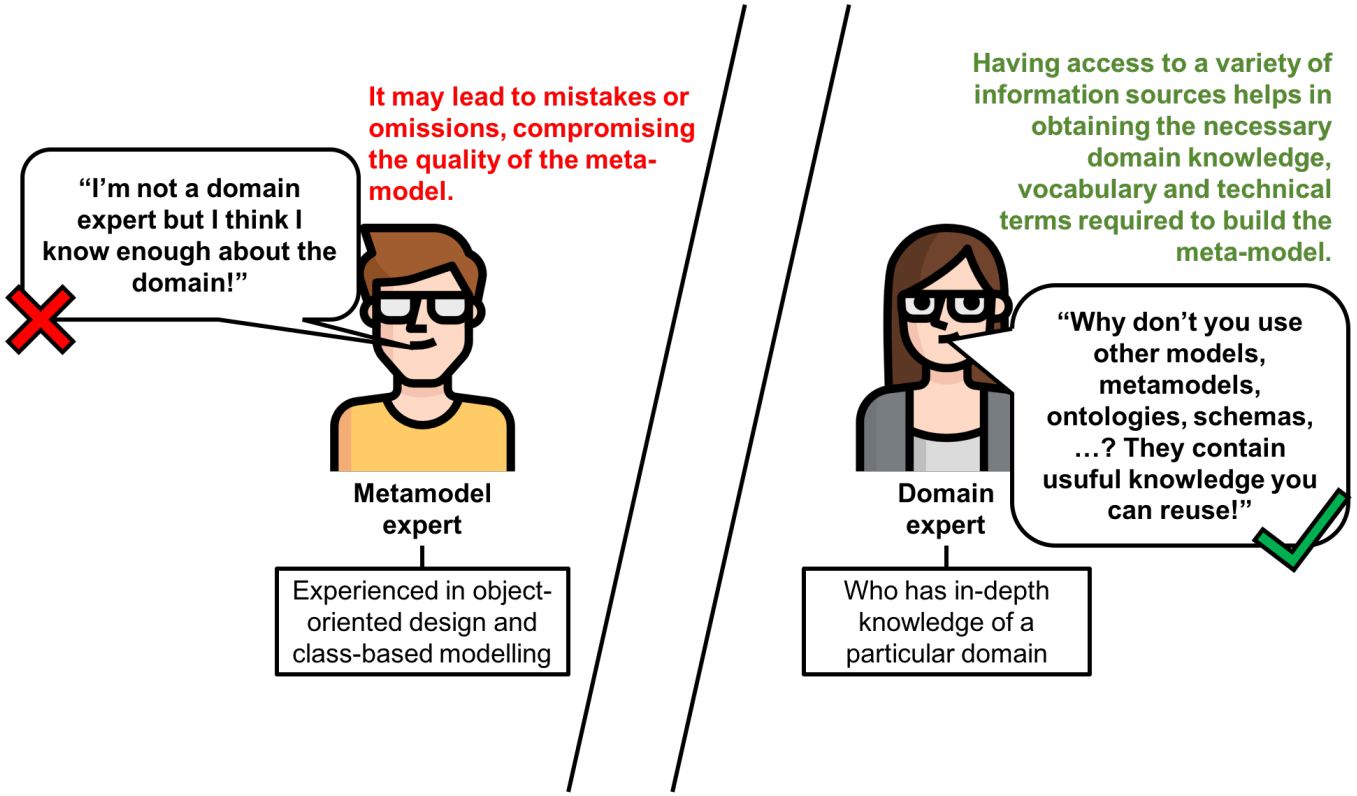

Figure: Motivation: Filling the gap between being a metamodelling expert and a domain expert

Overview

We propose an extensible approach to provide assistance during the modelling process.

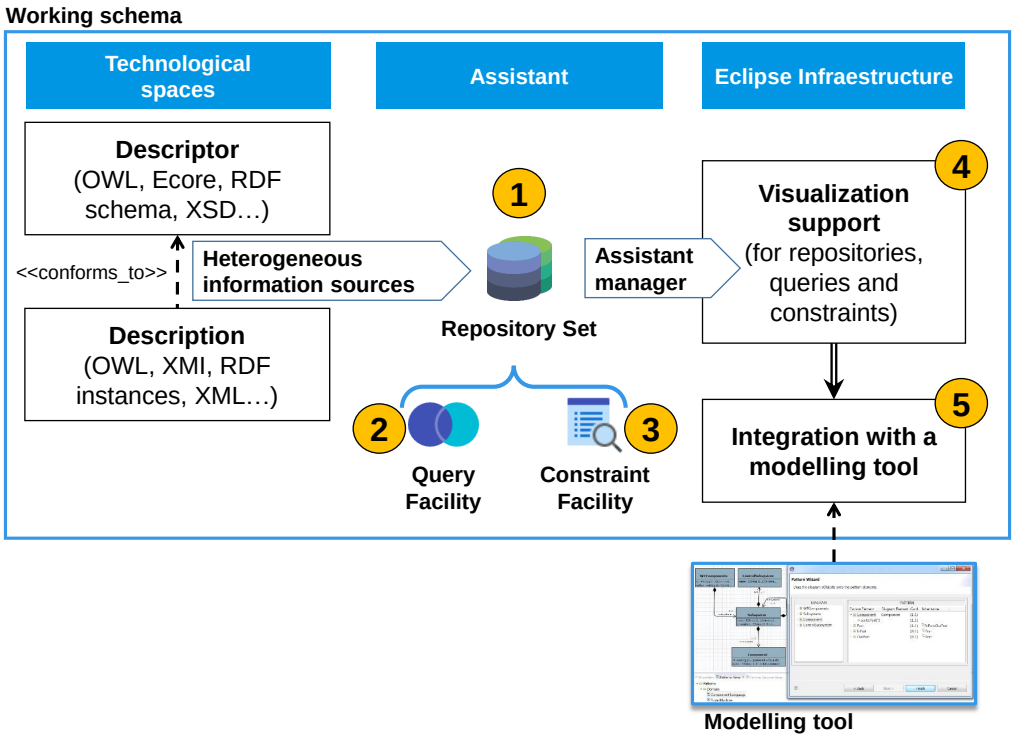

Figure: Overview of the approach

Our approach is useful...

As support for the development of new domain meta-models. This way, domain concepts and vocabulary can be sought in external sources such as XML documents or ontologies.

To create models for a particular domain. In this case, model elements conforming to the same or similar meta-model can be incorporated into the model being built.

To design a “concept” meta-model [1]. A concept is a minimal meta-model that gathers the core primitives within a domain, e.g., for workflow languages.

Our approach is useful...

To aggregate multiple existing models into a model-based product line [2, 3]. In this approach, a description of the space of possible features of a software system is created, typically through a feature model. The choice of features implies the selection of a software model variant. This way, there is a need for understanding an existing family of models and to describe their variability.

To detect "bad smells" [4] or signs of bad modelling practices [5] in a set of resources.

Architecture

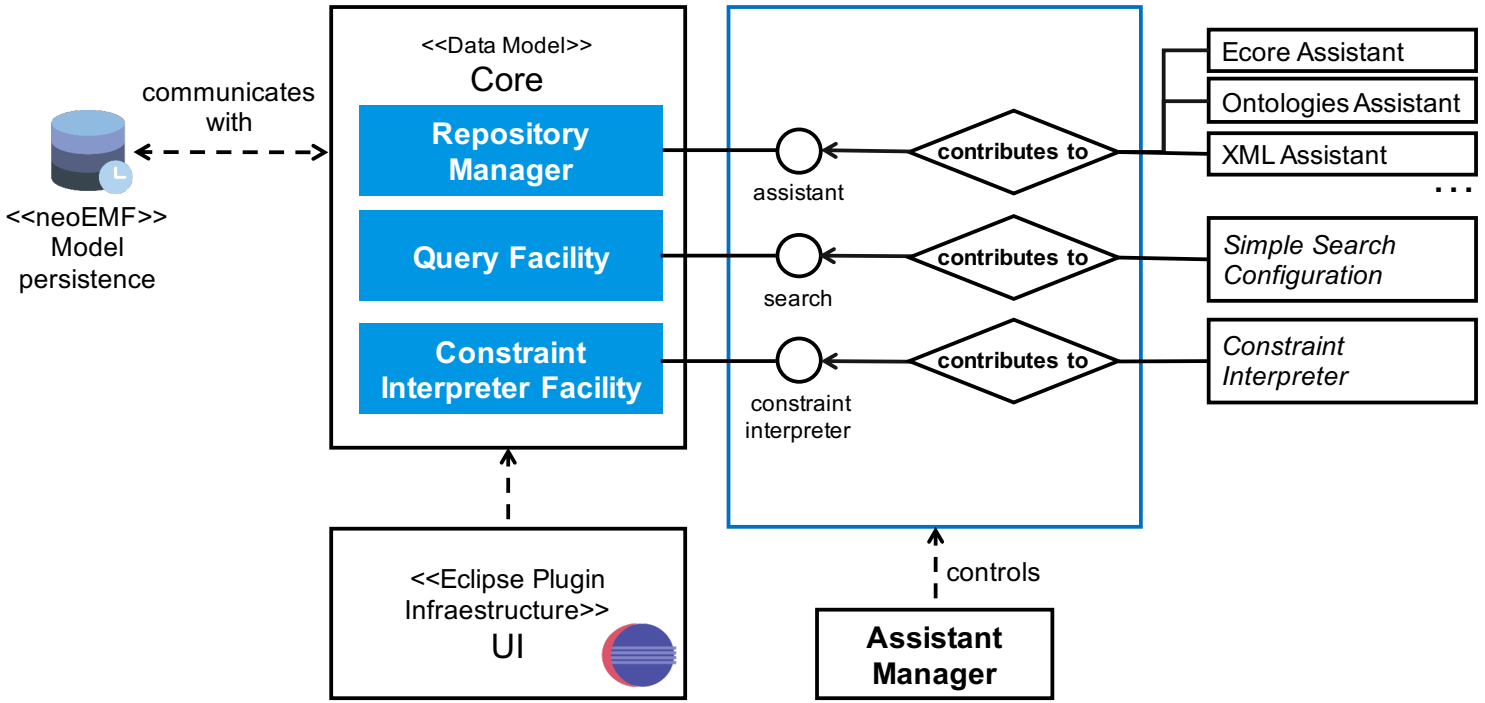

Figure: Architecture of the proposal

Architecture

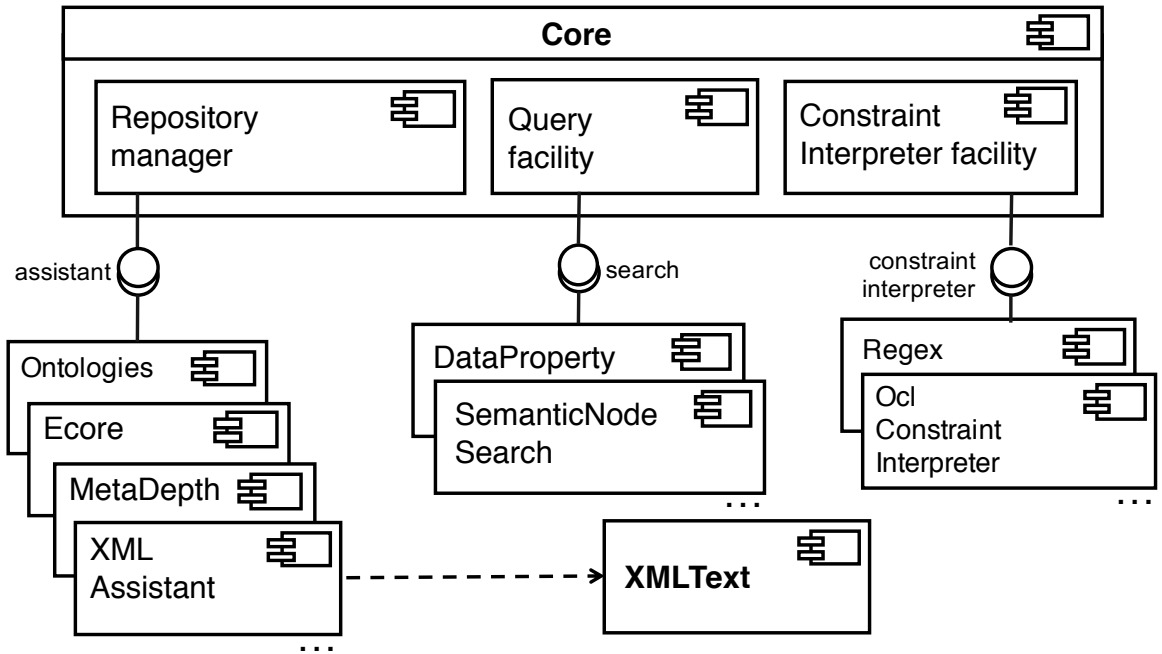

Figure: Architecture of the Core component

Architecture

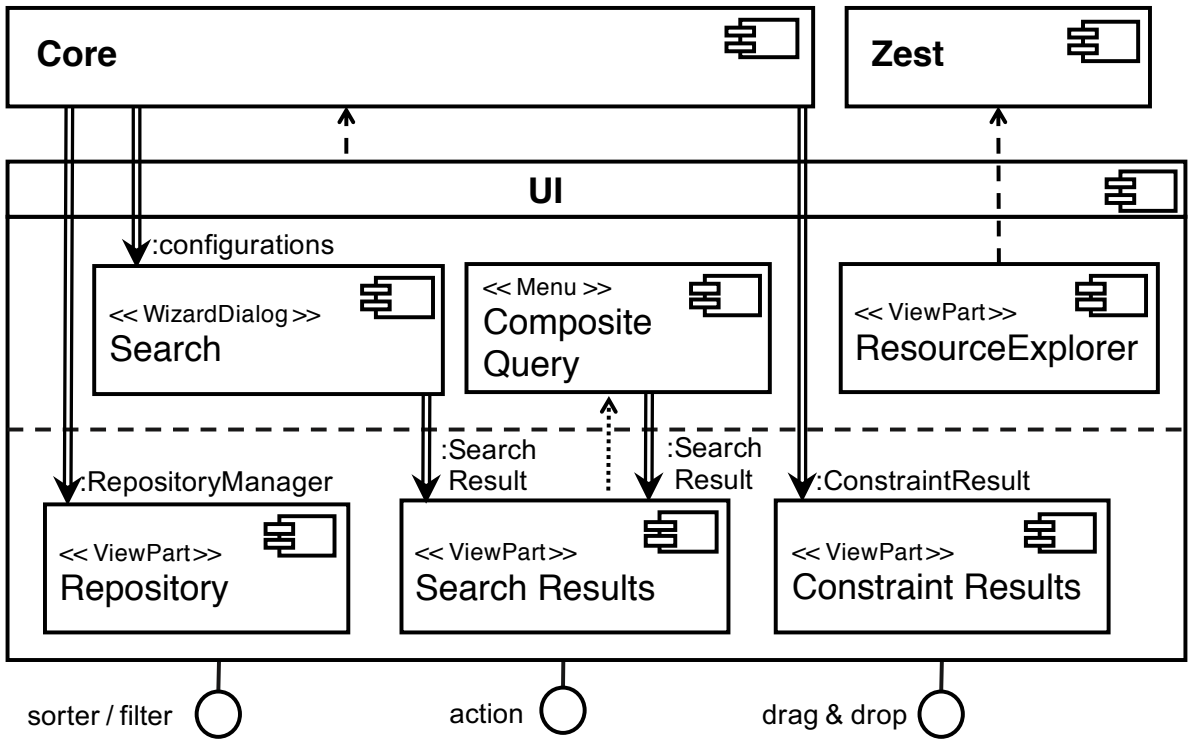

Figure: Architecture of the UI component

The ingredients of a modelling assistance

The handling of heterogeneous sources through a common data model

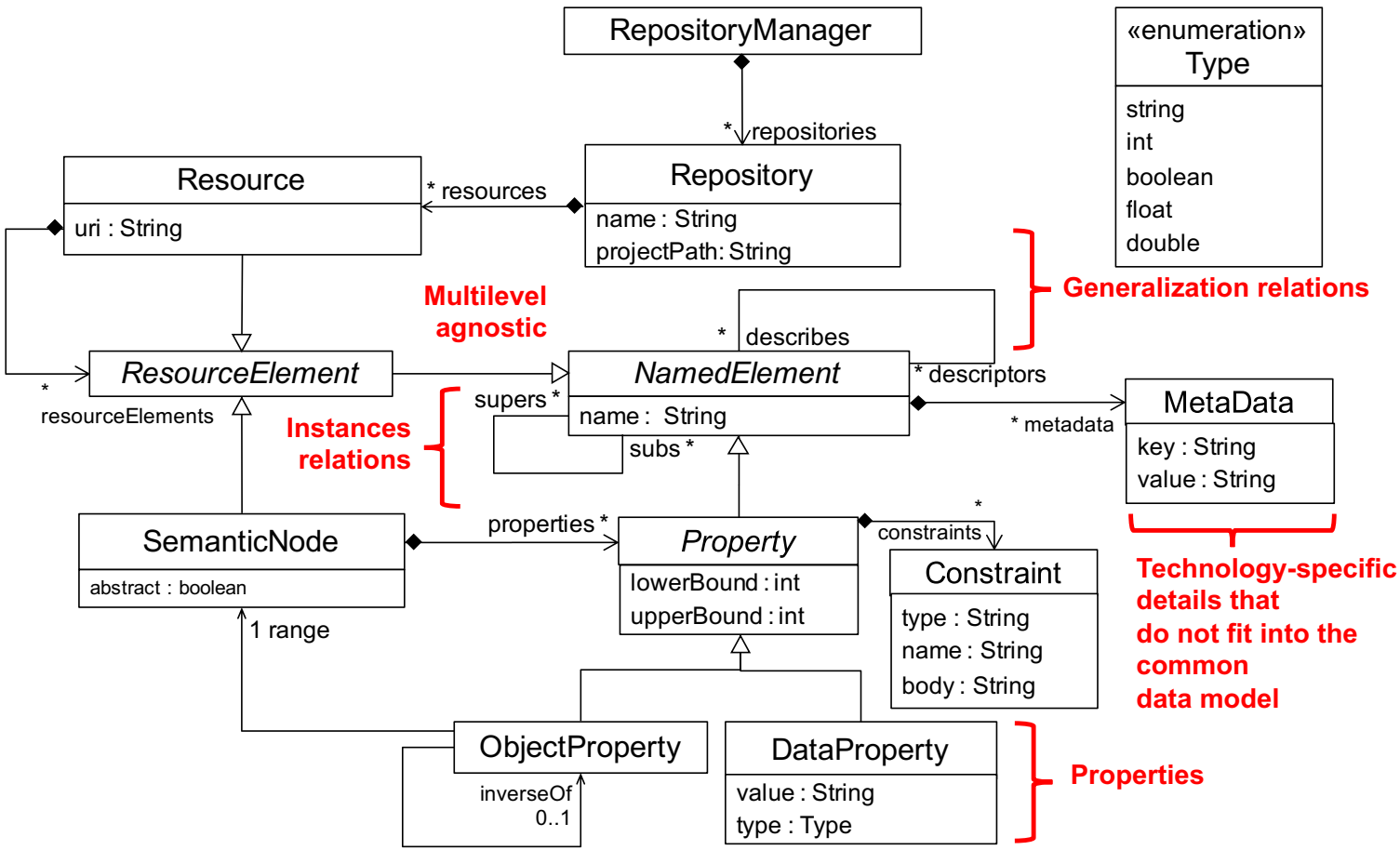

Figure: The common data model

The ingredients of a modelling assistance

The uniform support for queries

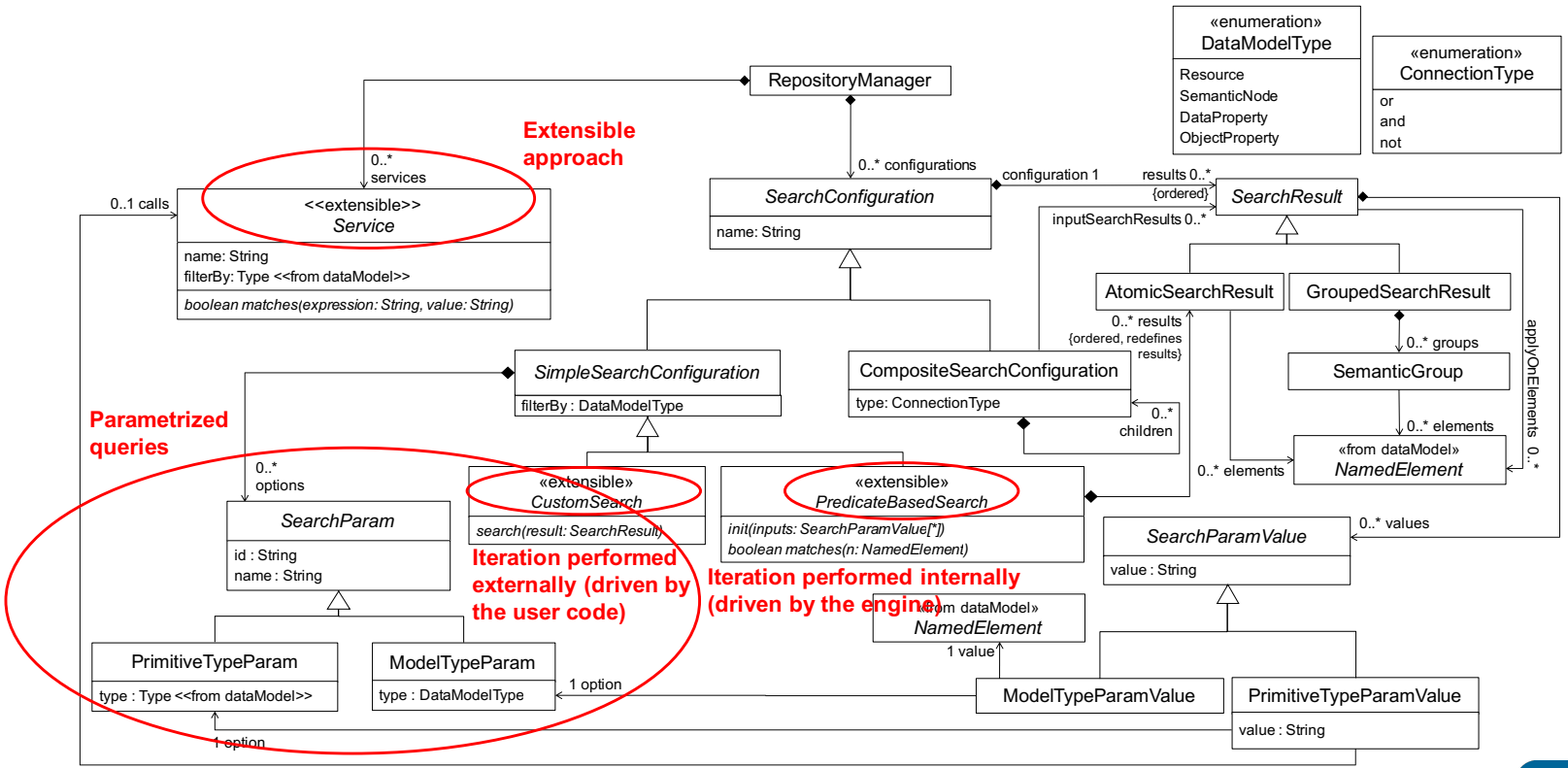

Figure: Meta-model of our extensible query mechanism

The ingredients of a modelling assistance

The managing of constraints

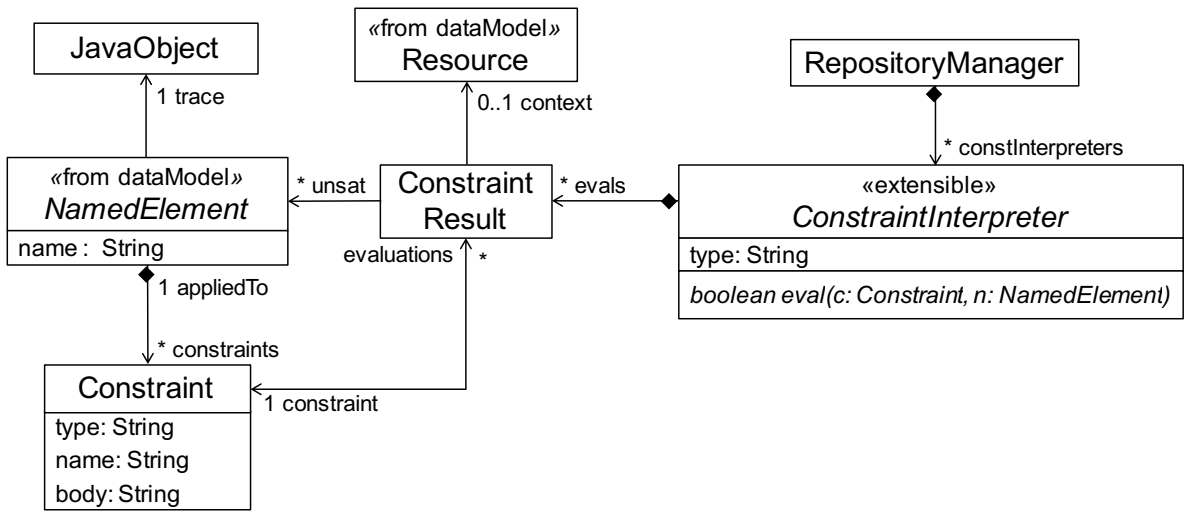

Figure: Meta-model of the extensible constraint handling mechanism

Extremo

Automated modelling assistance by integrating heterogeneous information sources

Computer Languages, Systems and Structures. Volume 53, 2018, Pages 90-120, ISSN 1477-8424

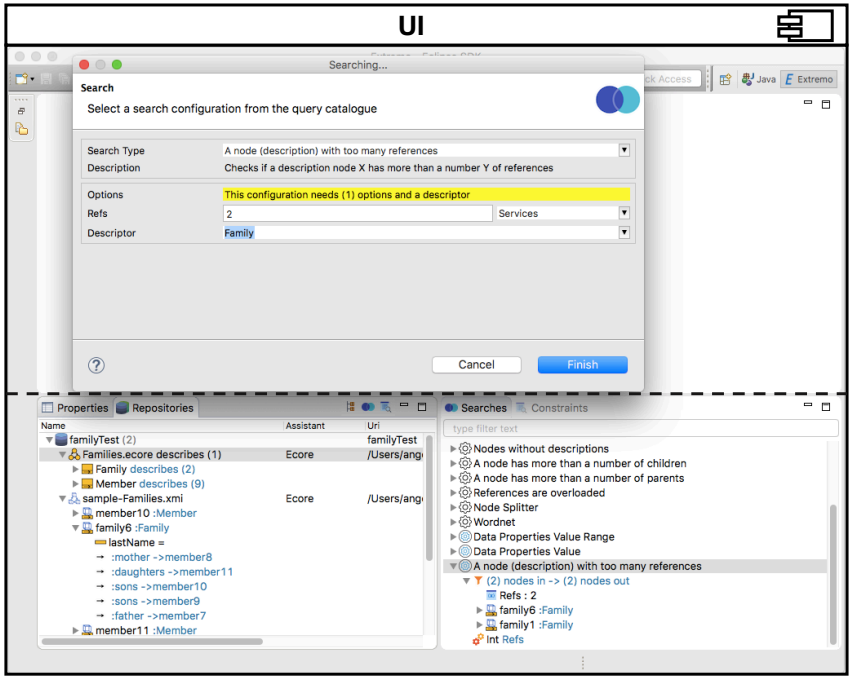

Figure: Extremo in Action: Search Wizard Dialog and views

Preliminary evaluation (DONE)

RQ1: How useful is Extremo to solve practical problems in Language Engineering? Two demonstration cases. In both of them, Extremo was used to support the development of complete DSL implementations for process modelling in the immigration domain and modelling of production systems.

RQ2: How capable is Extremo to represent information coming from different technological spaces? An analytical evaluation to assess how well Extremo's data model covers the space of possible information description approaches.

RQ3: How integrable is Extremo? One of the salient features in Extremo is the possibility of integrating it with third-party tools. Having already successfully integrated a number of tools with Extremo, we intend to assess the reach and limitations of the integration mechanism.

Present work (TO-DO)

Improve the performance of the tool in order to import and query bigger models, using a design for the importers to allow some levels of concurrency.

A study with users. Evaluate the perceived usefulness of Extremo by engineers in order to perform different modelling tasks (e.g., construct or modify a model). Compare the quality of the resulting models, and the effectivenes of the modelling task, with respect to not using assistance.

 J. S. Cuadrado, E. Guerra, J. de Lara, A component model for model transformations, IEEE Transactions on Software Engineering 40 (11) (2014) 1042–1060.

 K. Czarnecki, M. Antkiewicz, Mapping features to models: A template approach based on superimposed variants, in: GPCE, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 422–437.

 A. Polzer, D. Merschen, G. Botterweck, A. Pleuss, J. Thomas, B. Hedenetz, S. Kowalewski, Managing complexity and variability of a model-based embedded software product line, Innovations in Systems and Software Engineering 8 (1) (2012) 35–49.

 N. Moha, Y. Guéhéneuc, L. Duchien, A. L. Meur, DECOR: A method for the specification and detection of code and design smells, IEEE Trans. Software Eng. 36 (1) (2010) 20–36.

D. Aguilera, C. Gómez, A. Olivé, Enforcement of conceptual schema quality issues in current integrated development environments, in: Proc. CAiSE, Vol. 7908 of Lecture Notes in Computer Science, Springer, 2013, pp. 626–640.