Computer Languages, Systems & Structures 53 (2018) 90–120

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Automated modelling assistance by integrating heterogeneous information sources

Mora Segura Ángel[a,*], Juan de Lara[a], Patrick Neubauer[b], Manuel Wimmer[b]

[a] Modelling & Software Engineering Research Group, Universidad Autónoma de Madrid, Spain
[b] Business Informatics Group CDL-MINT, Business Informatics Group, TU Wien, Austria

ARTICLE INFO

Article history:
Received 8 November 2017
Revised 10 January 2018
Accepted 26 February 2018
Available online 5 March 2018

Keywords:
Modelling
(Meta-)modelling
Modelling assistance
Domain-specific languages
Language engineering

ABSTRACT

Model-Driven Engineering (MDE) uses models as its main assets in the software development process. The structure of a model is described through a meta-model. Even though modelling and meta-modelling are recurrent activities in MDE and a vast amount of MDE tools exist nowadays, they are tasks typically performed in an unassisted way. Usually, these tools cannot extract useful knowledge available in heterogeneous information sources like XML, RDF, CSV or other models and meta-models.

We propose an approach to provide modelling and meta-modelling assistance. The approach gathers heterogeneous information sources in various technological spaces, and represents them uniformly in a common data model. This enables their uniform querying, by means of an extensible mechanism, which can make use of services, e.g., for synonym search and word sense analysis. The query results can then be easily incorporated into the (meta-)model being built. The approach has been realized in the Extremo tool, developed as an Eclipse plugin.

Extremo has been validated in the context of two domains – production systems and process modelling – taking into account a large and complex industrial standard for classification and product description. Further validation results indicate that the integration of Extremo in various modelling environments can be achieved with low effort, and that the tool is able to handle information from most existing technological spaces.

1. Introduction

Model-Driven Engineering (MDE) advocates an active use of models throughout the software development life-cycle. Thus, models can be used to specify, analyse, test, simulate, execute, generate code and maintain the software to be built, among other activities [1–3].

Models are sometimes built with general-purpose modelling languages, such as the Unified Modelling Language (UML) [4]. In other cases, modelling is performed using Domain-Specific Languages (DSLs) [5]. DSLs contain tailored domain-specific primitives and concepts accurately representing the abstractions within a domain, which may lead to simpler, more intensional models. The abstract syntax of a DSL is described by a meta-model, which is itself a model. Meta-models are

[*] Corresponding author.
E-mail addresses: Angel.MoraS@uam.es (M.S. Ángel), Juan.deLara@uam.es (J. de Lara), neubauer@big.tuwien.ac.at (P. Neubauer), wimmer@big.tuwien.ac.at (M. Wimmer).
URL: http://miso.es (M.S. Ángel), https://www.big.tuwien.ac.at/ (P. Neubauer)

https://doi.org/10.1016/j.cl.2018.02.002
1477-8424/

typically built using class diagrams, describing the set of models considered valid. Thus, the construction of models and meta-models is a recurrent and central activity in MDE projects [6].

High quality models and meta-models are pivotal for the success of MDE projects. They capture the most important concepts of a domain or describe the features of a system. Nevertheless, they are mostly built in an unassisted way, with no mechanisms for reusing existing knowledge. This situation contrasts with modern *programming* IDEs, which support code completion or provide help for using a given API [7,8]. However, in the MDE field, the modeller normally has the burden of creating the model from scratch. For this reason, modellers would greatly benefit from flexible access and reuse of existing knowledge in a domain. This knowledge might be stored on various technological spaces [9,10], including the modelling technical space, but also the XML, ontologies, and RDF technical spaces.

In order to improve this situation, we propose an extensible approach that provides assistance during the modelling process. In our proposal, we extract the information from an extensible set of different technical spaces. For example, in the XML technical space, DTDs or XML schemas as well as specific XML documents are covered by the assistant; while in the modelling technical space, meta-models and models can be queried. This heterogeneous information is stored in a common data model, so that it can be queried and visualized in a uniform way. The query mechanism is extensible and can make use of services, e.g., for synonym search or word sense analysis. The results of the queries are prioritized and aggregated for all information sources in the repositories and can then be incorporated into the (meta-)model under construction.

We have realized this concept in Extremo and provide an open source Eclipse plugin, which is freely available at the Extremo project website[1]. The web site includes short videos illustrating the main concepts explained in this paper as well as a set of resources, which have been used during the evaluation. Extremo's architecture is extensible and modular by the use of Eclipse extension points, and enables the addition of new information sources and types of queries. The assistant has been designed to be easily integrated with external modelling environments, also through extension points.

We have evaluated our approach under several perspectives. First, we show Extremo's usefulness to create DSLs in two case studies. The first one is in the area of process modelling for immigration procedures and the second is in the area of standard-conforming industrial production systems. We have evaluated its extensibility by describing its integration with a variety of modelling tools, ranging from graphical to tree-based editors. In order to evaluate format extensibility (i.e., the ability to import from new technical spaces), we perform an analytical evaluation of the degree of coverage of the data model. The query mechanism is tested by describing a catalogue of common queries for object-oriented notations. Finally, we address a discussion and the lessons learned from the results of the evaluation.

In comparison with our previous work [11], we provide extensions for a set of different technical spaces that include constraint interpreters and an extensible query mechanism. Moreover, Extremo's internal data model has been extended to handle level-agnostic information, i.e., for an arbitrary number of meta-levels. For example, we have integrated XML schemas and multi-level models [12] as information sources, and integrated Extremo with further modelling and meta-modelling environments. Finally, we report on an evaluation based on process modelling and production systems domain case study, using the eCl@ss standard [13] and provide an analytical evaluation on the generality of the data model we propose.

The rest of this paper is organized as follows. Section 2 provides an overview of the approach and its motivation. Section 3 explains the main parts of the assistant: the handling of heterogeneous sources (Section 3.1), the ability to perform queries on them in a uniform and extensible way (Section 3.2), and the handling of constraints (Section 3.3). Section 4 describes the extensible and modular architecture of the assistant, and how it can be integrated with modelling and meta-modelling tools. Section 5 evaluates the approach under three different perspectives, which include the usefulness for (*i*) language engineering, (*ii*) data extensibility, and (*iii*) integrability with external tools. Section 6 compares with related work, and Section 7 presents the conclusions and lines for future research.

2. Motivation and overview

Many technical tasks in software engineering require from access to knowledge found in a variety of formats, ranging from documents in natural language, to semi-structured and structured data. There is a current trend to make such information readily available and easy to embed in different types of artefacts generated during the software construction process [14]. For example, in the programming community, there are efforts to profit from code repositories, and Q&A sites like StackOverflow to automate coding and documentation tasks [15–17]. Some of these approaches are based on a phase of artefact collection, followed by their preprocessing and storage into a uniform database, which then can be queried using appropriate languages [16]. Following this trend, our objective is to make available to the model engineer a plethora of (possibly heterogeneous) resources that can be queried in a uniform way, and embedded into the model being built.

In general, the task of creating a high quality meta-model is complex because it involves two roles: (*i*) a domain expert, who has in-depth knowledge of a particular domain and (*ii*), a meta-modelling expert, who is experienced in object-oriented design and class-based modelling. Nevertheless, many times, the meta-modelling expert is left alone in the construction of a meta-model, or needs to make a decision based on tacit domain knowledge or under-specified language requirements. In

[1] http://miso.es/tools/extremo.html.