Evaluation of MetaEdit+ for DSM

Jui Deshpande, Sam Minns, Glen Peek

September 23, 2015

Abstract

In software engineering, there are many approaches for handling the kind of complexities that emerge during development. One such approach is the use of domain-specific languages within the context of model driven development. As model driven development has become increasingly popular, new tools have emerged with the aim of allowing users to quickly, easily, and effectively create domain-specific languages. However, these tools are still far from mainstream. In this paper, we critique the usability of one such metacase tool, MetaEdit+, by defining a domain-specific language for a library system, describing the creation of this language using MetaEdit+, and reviewing our final result.

1 Introduction

In this project we create a simple library management system in MetaEdit+. The solution consists of some basic building blocks, such as books, authors, librarians.

2 Language Design Background

People find DSLs valuable because a well-designed DSL can be much easier to program with than a traditional library. This improves programmer productivity, which is always valuable. In particular it may also improve communication with domain experts, which is an important tool for tackling one of the hardest problems in software development.

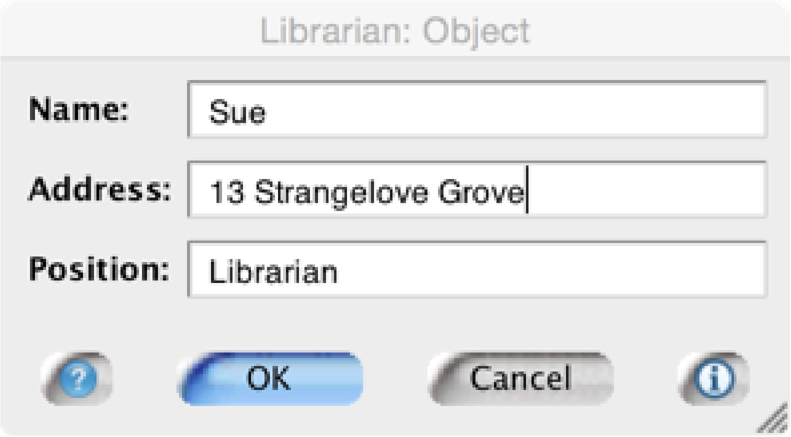
3 MetaEdit+

With MetaEdit+ an experienced developer defines a domain specific language containing the domains concepts and rules in a metamodel, and specifies the mapping from that to code in a domain-specific code generator. For the modeling language implementation, MetaEdit+ provides a metamodeling tool suite for defining the language concepts, rules, symbols, checking reports and generators. Next to the users guide, developers are able to watch several webcasts, read articles, blogs, brochures and even listen to podcasts. Generally, Metacase provides really good and clear documentation, but some points could certainly be refined.

Documentation provided MetaEdit+ offers various DSM resources. This includes a detailed Workbench Users Guide which discusses every aspect of metamodeling in MetaEdit+. All methods provided by the API are described as well. A single point of criticism in the guide is the rather brief explanation of the API methods and the few examples it delivers.

4 Library Language

The Domain Specific Language we have created is an attempt at modelling a Library. We have a chosen the context of a public library as opposed to a specialised facility such as a university or law library. The DSL is naturally constrained by our development tool and as such is comprised of three main entities. Objects represent physical entities such as loanable items, library equipment and people such as library users and employees. Relationships describe methods by which objects within the DSL may interact. Roles define constraints on the types of objects that may mutually INHABIT? TAKE THE PLACE IN THE RELATIONSHIP, ?BE?, ?RESIDE? word please end of a relationship between those objects. Additionally constraints may be defined surrounding the semantics of relationships. These constraints fall into four categories: Connectivity: cardinality and type of some shit about the thing with a thing. Occurrence: constraint surrounding the number of a unique objects may appear in a graph Uniqueness: uniqueness constraints for property values such as ID etc Port: No fucking idea what the shit port was for, probably something to do with ships. Creating and applying constraints allows further development and enrichment of DSL semantics.



Librarian: Object

Name: Sue

Address: 13 Strangelove Grove

Position: Librarian