A VISUAL DOMAIN-ORIENTED QUERY LANGUAGE

Oles Hodych, Yuriy Prokopiv

Information Systems and Networks Department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE

In recent years the subject of domain-specific languages (DSL) gained a substantial interest in both software research and engineering circles. Domain-specific languages promise to significantly simplify the development and support of software systems. This is achieved by shifting the development paradigm from programming towards modelling using concepts from the problem domain addressed by the system rather than Turing complete programming languages.

The majority of the current approaches to use DSLs pertain to the creation phase of software systems, whereby the actual software systems are the artefacts of this process. In our opinion the process of building and using software systems, especially in case of business applications, should be seamlessly integrated. This would provide domain experts with necessary tools to enhance a live system in order to meet dynamically changing real-life requirements without a tedious and often complex development/deployment cycle that is currently used in the software industry.

This article discusses a visual domain-specific query language (VDSQL) that supports data interaction and composition of business rules as part of a software system. It can be used directly by domain experts. The core purpose of any DSL is to provide a convenient way to interact with an underlying semantic model. VDSQL is an interactive DSL to query the underlying business model and to specify rules for monitoring the state of the system. The basic building components of VDSQL is a set of predefined blocks that can be snapped together in order to form more complex expressions. There are strict compatibility rules between blocks, which take into account their semantic context and type. Compatibility is determined dynamically and indicated to the user during interaction. The provided blocks consist of two main groups: one represents business model metadata such as entities and their properties, another – provides elements to compose expressions (e.g. comparison operators, aggregation functions). User interaction with VDSQL occurs in a workspace with advanced interaction capabilities. This includes features of Zooming User Interface, dynamic semantic compatibility indication when interacting with blocks, and context sensitive fluent interface in a form of a pallet of compatible blocks. An intuitive zoom in/out capability provides a way to include a large number of blocks into the same workspace, which streamlines reuse of pre-built expressions. The context sensitive fluent interface assists users by providing selection choices of only those blocks that are applicable at the specified location – a slot. The use of aggregation functions, conditions, logical operations, and conditional operators greatly facilitates the construction of complex rules.

An important aspect of the proposed approach is the semantic transparency between VDSQL expressions composed by software users and the Entity Query Language (EQL) expressions used by developers in a form of an internal Java DSL during a software construction phase. EQL represents a computation model for VDSQL. Such approach removes the discrepancy between the running system and its code, which greatly facilitates communication between domain experts and software developers.

Keywords – domain-specific language, data querying, projectional editing, HCI.