Chapter XIX

A Concept-Based Query Language Not Using Proper Association Names

Vladimir Ovchinnikov, Lipetsk State Technical University, Russia

ABSTRACT

This chapter is focused on a concept-based query language that permits querying by means of application domain concepts only. The query language has features making it simple and transparent for end-users: a query signature represents an unordered set of application domain concepts; each query operation is completely defined by its result signature and nested operation’s signatures; join predicates are not to be specified in an explicit form, and the like. In addition, the chapter introduces constructions of closures and contexts as applied to the language which permits querying some indirectly associated concepts as if they are associated directly and adopting queries to users’ needs without rewriting. All the properties make query creation and reading simpler in comparison with other known query languages. The author believes that the proposed language opens new ways of solving tasks of semantic human-computer interaction and semantic data integration.

INTRODUCTION

Conceptual models serve for application domain modeling as opposed to means of system implementation modeling. A conceptual model does not concern implementation details and describes an application domain’s essence. Conceptual models underlie conceptual query languages that are meant for querying schemas of the models (here and throughout the chapter, a model is considered to be a mean of modeling, and a schema
is considered to be a result of modeling). The languages have dual use. On the one hand, conceptual queries play the key role in constraint formalization—any constraint can be formulated as a query and an assertion upon it. On the other hand, the queries can be used for requesting data from an information system wrapped by a conceptual schema. In both cases, conceptual query transparency and simplicity are very important.

Aiming at more transparency and simplicity of conceptual queries, the author proposes Semantically Complete Query Language (SCQL) (Ovchinnikov, 2004b, 2005b; Ovchinnikov & Vahromeev, 2005). The language is founded on the semantically complete model (SCM) (Ovchinnikov, 2004a, 2005b, 2004c), the main property of which is semantic completeness that endows the model and query language with their names. A schema of the model is a set of application domain concepts, concept associations, and constraints defined over. The semantic completeness property implies a SCM schema does not include associations describing interrelation of application domain concepts differently; in other words, each association describes semantics of concept interrelation completely (more precise definition will be given in the section, Restrictions Imposed on Underlying Model). The main consequence of the property is that associations are based on unique (within a schema) concept sets; an association is identified with a set of underlying concepts, and not a proper name. As a result, SCQL is created that uses concept sets for referring to associations. The language permits querying by means of application domain concepts completely; proper names of associations are not used within it.

There are several other properties of SCQL that resulted in more simplicity and transparency of its queries: each query operation is completely defined by its signature and nested operations’ signatures; a signature of any query is an unordered set of application domain concepts; and join predicates have not to be specified in an explicit form. In addition, this chapter introduces conceptions of closures and contexts as applied to the language. The conceptions permit querying some indirectly associated concepts as if they are associated directly and adopting queries to users’ needs without rewriting. All the properties make query creation and reading simpler in comparison with other known query languages, which will be proved in the subsequent sections. The author believes that all these properties and others discussed permit usage of the language by end-users who are not specialists in information technologies (IT).

The chapter considers restrictions imposed on an underlying model by SCQL, the way of referring to associations within it, the structure of SCQL expressions, and the context mechanism. All ideas are illustrated using the running example introduced in the next section. Finally, the chapter shows the ways of application and development of the query language.

**QUERY SIMPLIFICATION METHODS**

The Impact of Ideology on the Organizational Adoption of Open Source Software
www.igi-global.com/article/impact-ideology-organizational-adoptions-open/3385?camid=4v1a