Java-Based Object Persistence Layer Jakamar

Erki Suurjaak

Abstract

My thesis introduces a concrete solution for achieving object persistence in a Java environment - the object persistence layer Jakamar.

Persistent objects are objects that exist beyond the lifetime of the application; this is typically achieved by storing them in some kind of a data store, most commonly a relational database. A basic task for software is producing, changing and viewing such data. There are several recurring problems that application programmers have to deal with when developing object-oriented applications that use relational databases as a persistence mechanism. The greatest problem is handling changes in the data structure - a simple alteration forces some reworking in both the application persistence logic and the database. Larger changes - e.g. switching the underlying database from one vendor to another - can require a great amount of work. Another concern is application design - the persistence logic of many applications is similar, differing mainly in the exact data structure. It would be favorable to exploit this similarity and produce an automated solution.

I set out to achieve the following goals:

- To propose a reusable generic approach to handling persistence logic - the *Persistence Broker* design pattern, which solves several problems when using relational database management systems in an application environment. This pattern represents a solution where business logic is separated from persistence logic to the degree where persistent objects are not aware that they are being stored and retrieved. The solution is easily customizable for the data structure of different applications.
- To present Jakamar, a component I had written in the Java programming language as an implementation of this pattern. Jakamar is an object persistence layer with the sole purpose of being a building block that is used for developing other applications. It

provides fully automated persistence - the application programmer has no need for writing any database access code, the component handles all data storage, retrieval and deletion by itself.

- To compare using Jakamar with using embedding database access logic directly in program code, and to compare Jakamar with other software products that follow the *Persistence Broker* pattern.

All the three goals were realized. I deemed the *Persistence Broker* to be a well-designed approach to persistence logic, and Jakamar to be a viable software component. Jakamar encapsulated the entire persistence logic of an application, and provided access to persistence via simple operations like *store(object)*, *delete(object)* and *retrieve(criteria)*. In addition to solving the problems listed above, using Jakamar yielded better modularized application structure, faster program development (as the programmer does not need to develop any database access code), and possibility for better performance on object retrieval.

Keywords: automated persistence, persistence layer, object-relational mapping, persistence broker.