Modelibra Software Family

Dzenan Ridjanovic
Université Laval
Québec G1K 7P4, Canada

Abstract -This paper provides a brief overview of Modelibra, the open source software family that is used to develop dynamic web applications based on domain models. The software family consists of a graphical design tool, a domain model framework, a web component framework, a collection of CSS declarations, and XML, database and Java code generators. Modelibra facilitates the definition and the use of domain models in Java. It uses Wicket for application views of a domain model. Wicket is a web framework that provides web components to construct, in an object oriented way, web concepts, such as web pages and page sections. Modelibra interprets the application model and creates default web pages based on the model. A default application may help developers validate and consequently refine the domain model. In addition, Modelibra has a collection of generic web components that may be easily reused in professional web applications to display or update entities.

I. Modelibra

In computer terms, a domain model is a model of specific domain classes that describe the core data and their behaviour [1]. The heart of any software is a domain model. When a model is well designed and when it can be easily represented and managed in an object oriented language, a developer may focus on views of the software and they are what users care about the most.

Modelibra has been designed to help developers in representing and using application domain models in a restricted way. The main restriction of Modelibra, and at the same time its main feature, is that all data must be present in main memory. This and other restrictions of Modelibra minimize the number of decisions that a domain designer must make. This makes Modelibra easy to learn. Modelibra is an Open Source Software (OSS) [2]. It is hosted at JavaForge [3]. Developers of OSS may find Modelibra useful for developing their software around a domain model, for providing an easy installation for their users and for developing a web application to introduce their software to the public audience.

By default, Modelibra uses XML files to save model objects. However, Modelibra allows the use of both relational and object databases. The upgrade of an application from XML data files to a database does not require a single line of programming code to be changed. Modelibra also provides the data migration from XML files to a database. Although the focus of Modelibra is a software application with relatively small amount of data, it provides some advanced features such as transactions and undo. A domain may have several models. One of them may be a reference model where common data, common to all models within a domain, are kept. A domain model may also inherit some of its definitions from another model in the same domain. In Modelibra, a part of the base model may be exported to another model, which can be taken for an off-line work, then returned back to synchronize changes with the base model.

The Modelibra software family consists of a graphical design tool, a domain model framework, a web component framework, a collection of CSS declarations, and code generators for XML configurations, database schemas and Java classes. Modelibra facilitates the definition and the use of domain models in Java. Modelibra interprets the application model and makes it alive as a default web application, which may help developers validate and consequently refine the domain model.

The software closest to Modelibra is Eclipse Modeling Framework (EMF) [4]. The objective of EMF is to provide code generation facility for building tools and other applications based on a structured data model. It has additional components for queries, transactions, model integrity validation and service data objects. It is quite a complex software that is not that easy to learn.

The next section will start with a simple domain model, which will be used to show how a web component is built from the model.

II. Domain model

A domain model is a representation of user concepts, concept properties and relationships between concepts. The easiest way to present a domain model is through a graphical representation. Fig. 1 represents a domain model of a simple web application called Web Links. It features web links that are of interest to certain members.

In our case, the domain model's concepts are: Url, Question, Category, Member, Interest and Comment. Url describes a web link. Urls are categorized. Categories are organized in a tree of subcategories. Question is a frequently asked question together the concept's attributes. The Url concept has only one neighbor, the Category concept. However, the Category concept has four neighbors: Url, Question, Interest and Comment.
Category concepts. A relationship between two concepts is represented by two neighbor directions, displayed together as a line. A neighbor direction is a concept special (neighbor) property, with a name and a range of cardinalities. A neighbor is either a child or a parent. A child neighbor has the max cardinality of N (or a number greater than 1). A parent neighbor has the max cardinality of 1. If a parent neighbor has the min cardinality of 0, the parent is optional.

A concept is represented as a list of entities. The retrieval of entities starts with the entry concepts of the domain model. The retrieval of neighbor entities may start. A child neighbor is represented as a list of entities. A parent neighbor is represented as a single entity. The Url concept is not an entry concept. Hence, entities of the Url concept may be reached only through its parent Category concept. The Interest concept has two parents. Thus, interests may be retrieved either from the Member concept or the Category concept. A concept that has more than one parent is called an intersection concept and has the X sign in the upper right corner of the concept.

Every concept has a predefined property called oid. The oid property is mandatory. It is used as an artificial identifier and is completely managed by Modelibra. Its value is unique universally. In addition, a concept may have at most one user oriented identifier (id) that consists of the concept's properties and/or neighbors. A simple id has only one property. In an entry concept, all entities must have a unique value for the

In the Web Link domain model, the entry concepts are Category, Question, Member and Comment. They all have a darker border and the || symbol in the upper left corner of the concept. Once an entity of the entry concept is retrieved in the list of entities, the retrieval of neighbor entities may start. A child neighbor is represented as a list of entities. A parent neighbor is represented as a single entity. The Url concept is not an entry concept. Hence, entities of the Url concept may be reached only through its parent Category concept. The Interest concept has two parents. Thus, interests may be retrieved either from the Member concept or the Category concept. A concept that has more than one parent is called an intersection concept and has the X sign in the upper right corner of the concept.

Every concept has a predefined property called oid. The oid property is mandatory. It is used as an artificial identifier and is completely managed by Modelibra. Its value is unique universally. In addition, a concept may have at most one user oriented identifier (id) that consists of the concept's properties and/or neighbors. A simple id has only one property. In an entry concept, all entities must have a unique value for the

![Diagram of the Web Link domain model](image)

**Fig. 1.** Web Link domain model