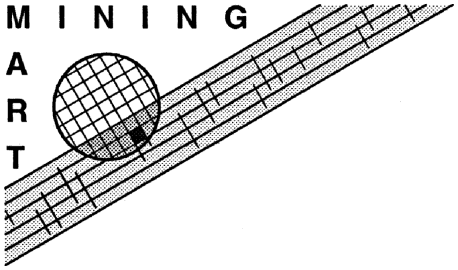

Enabling End-User Datawarehouse Mining
Contract No. IST-1999-11993
Deliverable No. D9

Internet Presentation of Mining Mart Cases

Stefan Haustein
Rolandstrasse 27
D-46045 Oberhausen

University of Dortmund / Swisslife

November 22, 2002

Chapter 1

Overview

The goal of WP 9 was to implemenent Web access to the M4 Model and the Buisiness Layer of sample Mining Mart cases.

The internet presentation of Mining Mart cases has been implemented using the *Information Layer* software (Infolayer[1]), developed at the University of Dortmund.

In a nutshell, the Infolayer is a program that dynamically generates a web presentation from a given ontology definition file. It generates instance lists and input forms for all concepts contained in the ontology. Associations between instances are displayed as hyperlinks. Type information contained in the ontology is used to restrict input forms and to provide convenient selection lists. The content of the given model can be browsed and edited using a regular web browser. The ontology must be provided as an UML class diagram stored in XMI format.

A demonstration server and the UML model containing the M4 Model and the Buisiness Layer are available on the Mining Mart Web pages[2].

[1] http://www.infolayer.org

[2] http://www-ai.cs.uni-dortmund.de/FORSCHUNG/PROJEKTE/MININGMART/index.eng.html

Chapter 2

Infolayer Extension: Database Connections

By default, the InfoLayer software stores all instance information in XML files. Since the M4 cases are stored in database tables, the Infolayer software has been extended with a database access interface, described in more detail in the following sections.

2.1 Database Connections

In order to notify the Information Layer that the instances of a class are stored in a database table, a tagged value named `il-connection` must be added to the corresponding UML class definition. The tagged value denotes the database connection and table name in the following format:

`db:jdbc:`*`jdbc-url`*`;user=`*`username`*`;password=`*`password`*`;table=`*`table`*

Please note that the *jdbc-url*, *username*, *password*, and *table* parameters must be replaced with corresponding actual values. The attribute names of

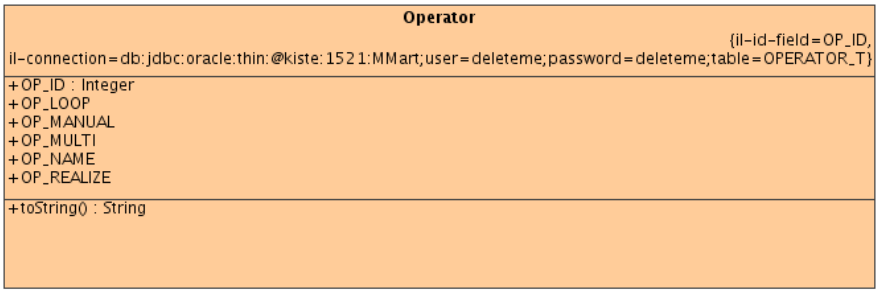

Figure 2.1: Example database table connection for the M4 concept `Operator`

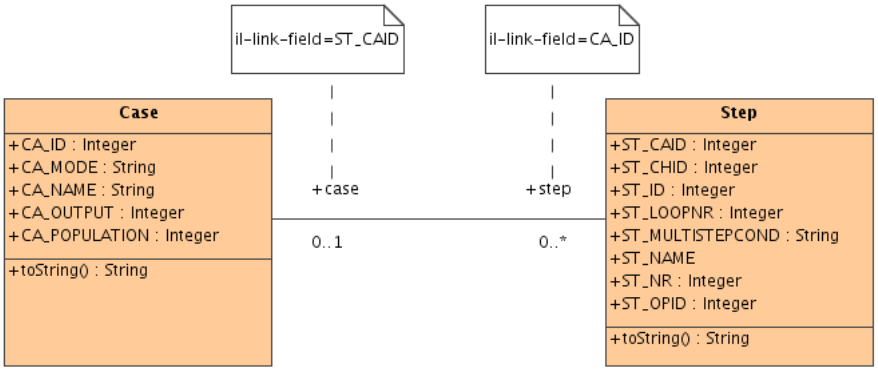

Figure 2.2: Tagged values for providing database implementation details of $1:n$ associations.

the class must match column names of the linked table.

If the table does not contain a column named `id`, or the column named `id` does not contain a unique identifier for each row, a corresponding column must be specified using the tagged value `il-id-field`. Figure 1.1 shows an example specification of a database connection for the M4 concept `Operator`.

2.2 Associations

The Information Layer stores associations between instances in simple lists at the association ends. In relational databases, associations are usually modelled by matching field values in different tables.

2.2.1 1:*n* Associations

The simplest case are 1:n associations. For instance, in the M4 model a single case consists of a set of steps. This association is modelled by the column `ST_CAID` in the step table. All steps in whom the column `ST_CAID` matches a given case id (column *CA_ID*) belong to the case with that id.

Again, tagged values are used in the UML diagram to make this information accessible to the Infolayer. For 1:n associations, it is sufficient to specify the fields defining the association by adding the tagged value named *il-link-field* to both association ends. Figure 1.2 shows the described M4 example. The column name *ST_CAID* is specified in a tagged value named `il-link-field` of the *case* association end. For the other end of the association, the tagged value *CAID* is provided.

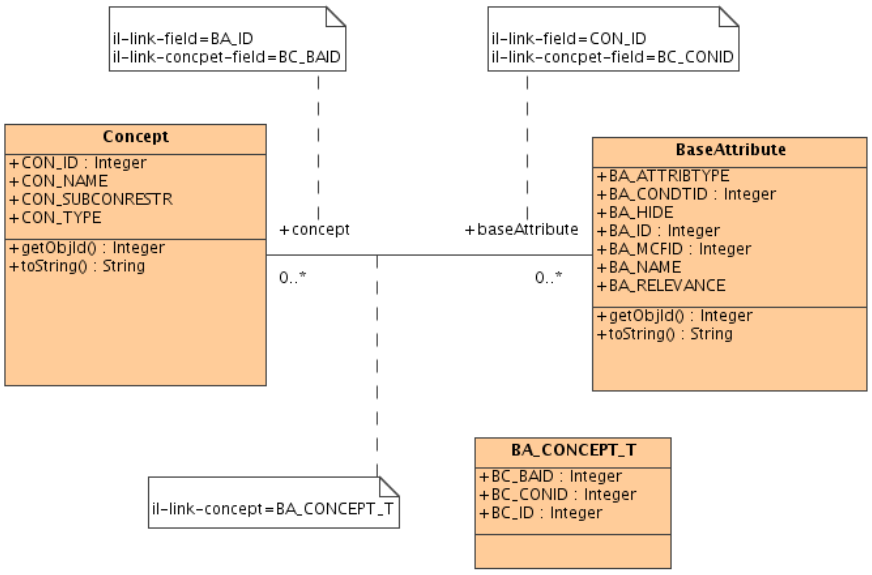

Figure 2.3: An M4 n:m association including the association table and the infolayer annotations

2.2.2 *n*:*m* Associations

While 1:n associations in relational databases can simply be built using the approach described above, for *n*:*m* associations an explicit association table is needed. This additional table links arbitrary numbers of associated rows by storing pairs of row ids from both tables. An example for n:m associations in the M4 model ist the association between base attributes and concepts. On the one hand, a single concept can have a set of associated base attributes. On the other hand, one base attribute can belong to more than one concept. To model this association, the relational M4 implementation uses the table `CONCEPT_T`. Each row of this table links a pair of ids from both associated tables.

To be able to handle this representation of associations, the Information Layer software needs additional information, namely the name of the association table and the names of the columns containing the ids in the association tables. The name of the association table must be stored in the tagged value named `il-link-concept` of the association. Additionally, the tagged values `il-link-concept-field` must be set at the association ends, denoting which column of the association table corresponds to the column given in `il-link-field`. Figure 1.3 shows a corresponding UML example.

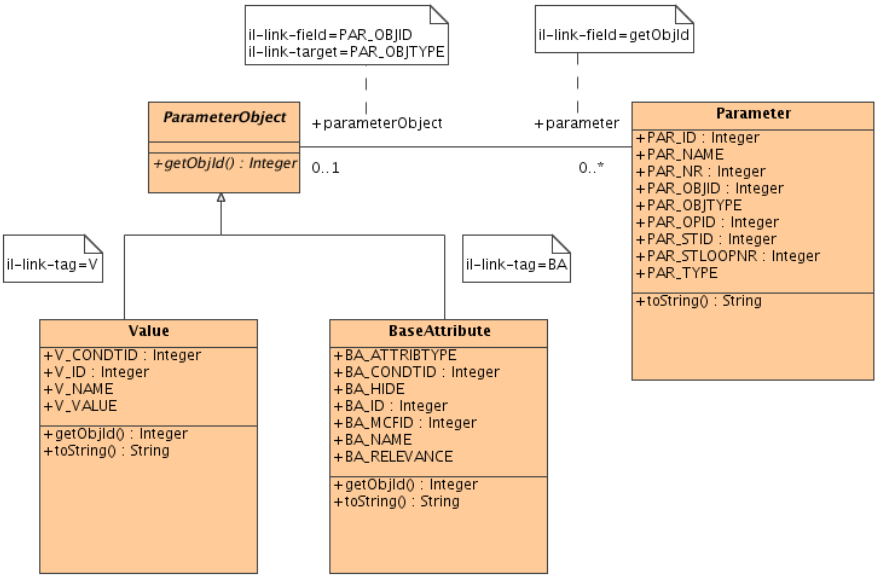

Figure 2.4: A dynamic association between the Parameter class on the one hand and the classes BaseAttribute and Value on the other hand.

2.2.3 Associations linking different tables dynamically

A special case of an association is an association where the table at one association end is not fixed but denoted by a column of the table at the other association end. For example, in the M4 model, a parameter object id can denote rows from different tables, where the target table is contained in the column `PAR_OBJID`.

In order to model this special case in the information layer, all classes corresponding to dynamically associated tables must be subclasses of a common superclass. The association is not drawn to all target classes, but only to the common superclass. The tagged value `il-link-target` must contain the field that denotes the table name. If the content of this field does not contain the target class directly, the target clases must be marked with a corresponding `il-link-tag` tagged value. Figure 1.4 shows a corresponding UML diagram.