Metadata Management in Building Bridges between Software

Berati, G.

Abstract - *This article presents a novel approach for constructing converters between software environments and a proposal for a transformation methodology between different software environments, especially between databases modeling software environments. Advantages of the proposed Meta Data Modeling are evidented.*

Introduction

Issues treated in this paper concern the theoretical level efforts to resolve a concrete problem in a project to restructure the cargo system in the airline company KLM (Netherlands). The problem stood in the inability to generate the data model of this project from the program (Casetool-i), which was selected as a modeling tool for of the project. It was really a large amount of information that should be lexicalized and generated to provide the data base model by the Casetool FCO-IM (Casetool selected as standard for the modeling of information in company Atos Origin BMS, the company that should implement the project). This amount was not possible to handle from the toll.

Problem Statement

The problem was in the lexicalization process. The project of the model had been almost completed, but it couldn't be generated. It was needed to restart the project from the beginning in another modeling tool or to transform the project using a bridge to another modeling tool. The resumption of the project was practically impossible because of the high cost, so efforts should focus on the possibility of translation of the model in a format of a tool, which allows the generation of a large-amount data base project.

Being involved in the project for implementation of the converter that will enable the transformation of the model conducted in FCO-IM to a model in ERWIN tool, I suggested the idea of building an intermediate repository.

The steps followed theoretical results and general conclusions were reached, representing an important theoretical research in the field of meta modeling, which, especially in our country, is little known.

Meta Model **(Berg, J. M., Levin, O. & ROUILLARD, J. (Meta-Modeling: Performance and Information Modeling. Http://books.google.com Internet Material Volume 6: Meta-Modeling: Performance and ... System-Level Performance Model and Method))** is the core of structure of any model, i.e. it specifies the way the software maintains data of itself and of its files. Data base, which makes storage of these data, is called repository or nucleus (kernel). So, when it comes to the meta level, it is discussed at the level of data.

Problems with recognition (acceptance) or non-recognition of a file from the software of the same type are well known. Even within software, we have such problems between different versions of it. Anyone has experienced failing to open an MS-WORD file created in MsOffice 2003 in MS-WORD 97. The problem is how repositories store relevant information.

If we pay attention to the structure of the relevant repositories of some different software environments of the same type, we can build bridges to translate files (models) from one to another. Knowing very well repositories of the software environments can help also for other purposes such are

improving versions of the software or rising opportunities to use information provided by the software.

Let's focus our job within family of database analyzing and modeling software environments. Let's try to connect with a bridge two different software tools of this type.

Existing solution and their criticism

In general, conversions are done by directly converting repository data from one enverinment to another one. That might have good impact in complexity of the conversion algorithm, but this way of doing conversion can cause lossing a lot of information that is not relevant from one environment to another one.

Let us see a concrete conversion. Let us convert from **FCO-IM Casetool v4.1** to **Erwin.**

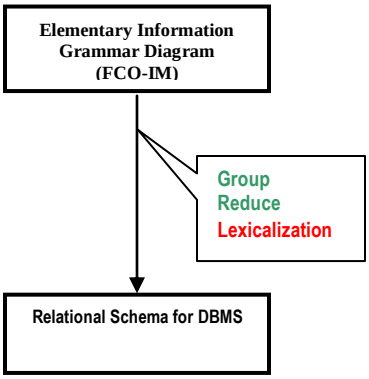

Figure 1

1 - FCO-IM Casetool v4.1
It is a tool which is based on the theory of Facts Oriented Modeling. This tool is very suitable for complex models and very flexible in implementing constraints of the model (Business Rules); it is very convenient in determining the relationship, etc.

However, practice showed that the FCO-IM Casetool has problems in managing large projects. Lexicalization procedure of the large amount of information was not covered by Version 4.1. That process, in this tool is performed by using operating memory (RAM).

The model created by this Casetool we will call Model FCO-IM. The steps that follow in the model, until the database or script is generated, is shown simplified in Figure 1. In Figure 2, is presented the information grammar (IG) and diagrams of grammatical information (IGD) for a very simple example. Also is shown the FCO-IM repository of the model. (**This commentary refers to VAN DER LEK, H., BAKEMA, G. & ZVART, J. P. 2001: Fully oriented Communication - Information Modeling FCO-IM, Page 57-90).**

All necessary information is in tables of the repository. In this paper is shown how and why information will be pumped into an intermediate repository and why an intermediate repository is the optimal solution.

2 - Erwin is a tool based on the theory of entity relationship modeling (ERM). This Software is based primarily on entities and not on facts to model and design a process.

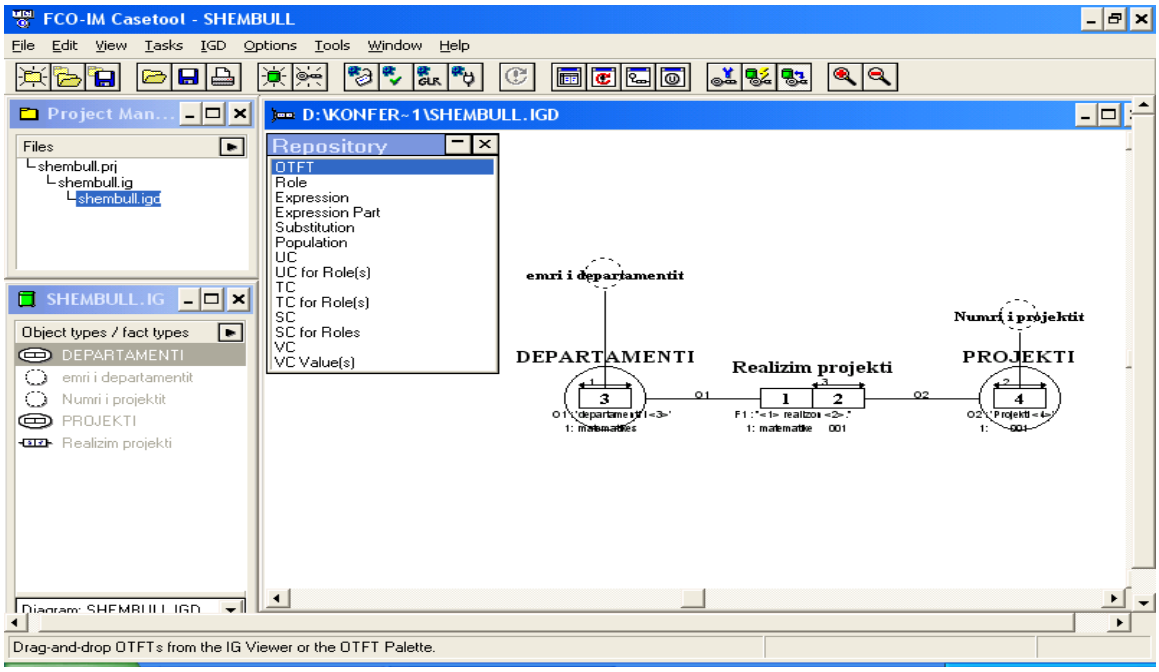

Figure 2

Besides other advantages, models built with this tool can be generated for a very large number of database management systems (DBMS). But, in our case, the most important advantage is the unlimited amount of information that could process this tool. Model constructed with this tool is referred to as the ER Model.

An FCO-IM model grouped and reduced is a masked ER model. For this reason it is theoretically possible that a grouped and reduced model is exported into an intermediate deposit and later on in an ER tools repository.

Essence of the proposed solution

The FCO-IM repository contains a lot of information which cannot be stored directly into repositories of the Entity Relational Tools. This refers to specific information of this tool, such population and some extra constraints. Of course, it would be good that this kind of information not be thrown away. We can store this information in Intermediate Repository, so it is necessary to add tables in Intermediate Repository to keep this information.

Repositories of ER tools (e.g. Erwin) have many tables which hold the tool-specific information and the model information which is automatically generated once the model is created. Therefore, intermediate repository must be a subset of the tables of the repository of the ER tool. Since we are dealing with a relatively limited number of tables, intermediate repository will not be very complicated.

Considering the fact that we have a large amount of information which is stored in different ways in the two environments mentioned above, the idea of building an intermediate deposit of the model information is more flexible then a direct conversion.

Conclusion for building an intermediate repository was reached after I have studied in depth, functioning, semantics, and the structure of the two tools mentioned above.

At the same time, I accurately identified all elements of an ER model and ways of sorting of these elements into its repository comparing to FCO-IM repository.

So, if we model a complex (full) ER model, we can perform a repository which is a wise place for the conversion of the FCO-IM model into the ER model. So, in this direction, ER is the destination of the conversation.

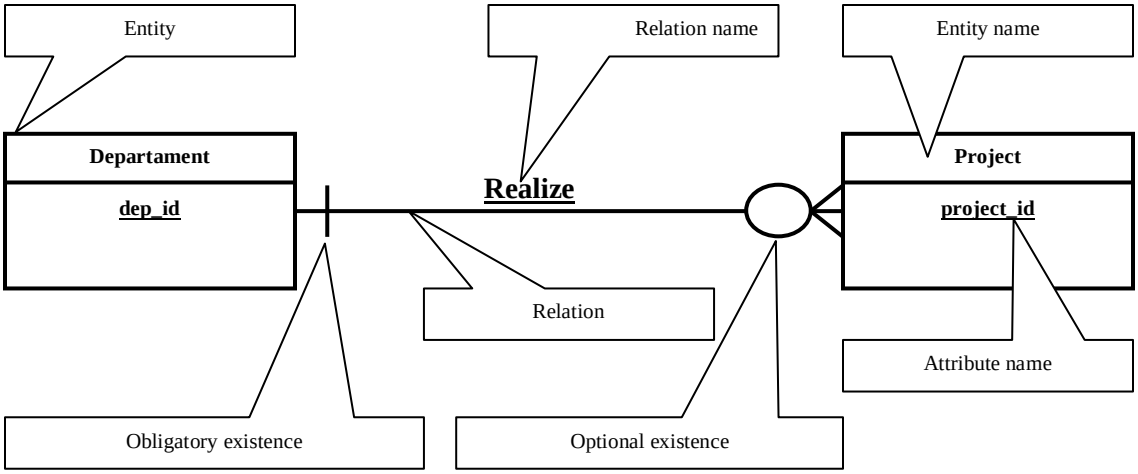

Figure 3

Analysis

Elements of a very simple model will be presented in the ER diagram, as in Figure 3. Of course, this model is not the one that is used to build intermediate repository, because this doesn't have all elements of a model. This example is taken only to show the idea of modeling in the ER and the idea of some elements of the model of the model.

Figure 3 explains what the main elements of a data model are and modeling this model is the way to create the model called data model, which in itself is Intermediate Repository which we are talking about.

Details

Data base which we are building contains a table that will hold all the elements of a model. Let us choose an ER model which is complex in the component elements, so an ER model which has as many different elements such as entities, attributes, one to one relations, one to many relations, optional attributes, mandatory attributes, many constraints, etc.

Entities and relations **(according RAMAKRISHNAN, R. & GEHRKE, J. 2002: Database Management Systems. Chapter 2. Page 25-32)** are those that we will model first, then are the attributes, which are divided into key attributes (primary key, key foreign therefore key attributes) and finally non-key columns (non key attributes).

Below are presented the steps for creating of Intermediate Repository of the model (Figure 4). Issues of importance are:

• Identificagtion of elements of a complex ER model and modeling them
• Verbalization with FCO-IM of the ER model (so we can use FCO-IM to build IR).
• Conducting a draft FCO-IM diagram of the ER model
• Completing the diagram with the relevant conditions (addition of key columns in the diagram, adding non-key columns in the diagram, etc.)
• Making the diagram of the generalization of hierarchies
• Validation of the model through nominalization
• Adding integrity rules
• Generation of the model of the model (intermediate repository).

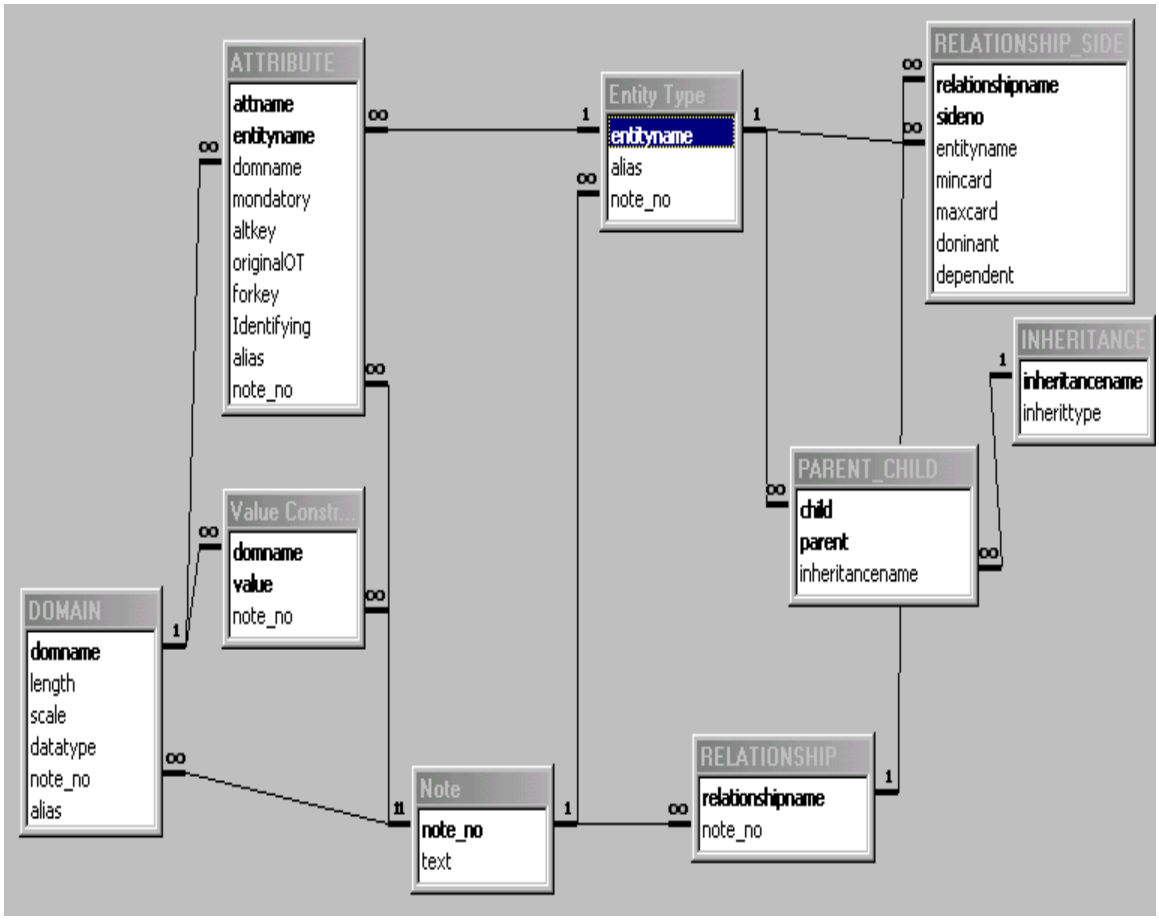

Figure 4. Intermediate Repository in MS Accsess

Environment of FCO-IM tool (Fully Communication Oriented Information Modeling) offers the possibility to explore the repository of the model in which all required information of the model are stored.

Since the semantic of FCO is very readable, we have chosen to implement the meta model of ER model complex in FCO-IM. With this we created a model which will represent in fact the structure of Meta ERM model; of course, only the essential elements of a model. In this model will be included entities, relations, attributes, rules of integration, information recurrences, subtypes, etc. Now in the role of the Entities, will be elements of the Conceptual Model which is selected (e.g. entity, relationship to example above).

The essence of this paper is precisely this model which is the intermediate repository which is an additional step to the process of conversion of the model. This intermediate repository can become an independent mechanism or a part of converter tools. So, now we have an independent environment (with model information form still unspecified), from which one can generate different ER patterns of model. Precisely that is the biggest advantage of this repository.

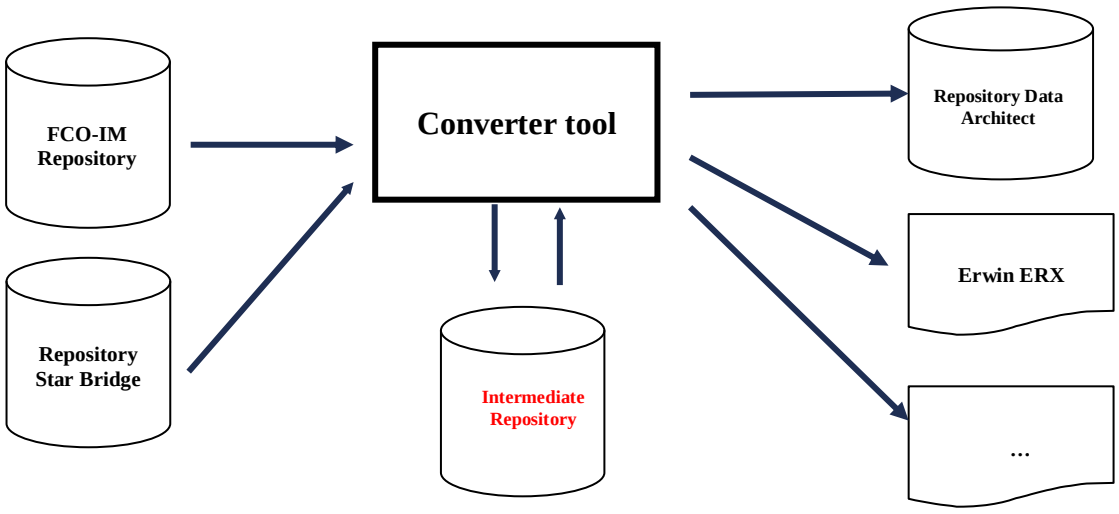

Figure 5. Transformation Schema

Conclusions

Designing intermediate repository enables the creation of easy converter tools. Concretely, we created a converter to surpass the FCO-IM Casetool Lexicalization bug. The tool can convert to a larger number of ER tools (such as: Data Architect, Power Designer, etc.).

Based on these results, one can propose a method for running authentic transformations that can be called the transformation method based on the intermediate repository (Intermediate Repository Based Transformations).

Results of this research have been used in a number of applications in the Albanian software industry, in Shkodra and Tirana.

References

[1] BERGE, J. M., LEVIA, O. & ROUILLARD, J. Meta-Modeling: Performance and Information Modeling. **ISBN:** 079239688X. Publisher: Springer. Volume 6: **Meta-Modeling**: System-Level Performance **Model** and Method), USA

[2] VAN DER LEK, H., BAKEMA, G. & ZVART, J. P. 2001: Fully Communication Oriented – Information Modeling FCO-IM, Page 57-90. Netherlands

[3] RAMAKRISHNAN, R. & GEHRKE, J. 2002: Database Management Systems. **ISBN:** 079239688X. Publisher: McGraw-Hill Professional. Chapter 2. Page 25-32, USA