The Multi-Meta Model Aimed for Schema Reuse in Conceptual Database Design

MIRTA BARANOVIĆ, SLAVEN ZAKOŠEK, DAMIR KALPIĆ
Faculty of Electrical Engineering and Computing
University of Zagreb
10000 Zagreb, Unska 3,
CROATIA
mirta.baranovic@fer.hr  http://web.zpr.fer.hr/mirta

Abstract: - In this article a database model, called multi-meta model, aimed for schema reuse in conceptual database design is described. The multi-meta model is based on the entity-relationship model. The corresponding multi-meta database contains the description of existing database models. In the description of models contained in the multi-meta base, the redundancy has been maximally reduced without loss of existing knowledge. In this way, any ambiguity regarding the model description has been eliminated. The restructuring of knowledge about models is simplified, while the model consistencies are preserved. The necessary conditions are defined that have to be satisfied by each database model. If these conditions are not met, the described procedures for the model restructuring have to be applied.

Key-Words: - Database, modelling, meta data, multi-meta database, schema reuse, conceptual design

1 Introduction

In design of a database model, the principal goal is to satisfy the existing and future user requirements. The design has to satisfy usual requirements for data modelling, such as integrity, consistency and non-redundancy. In addition, a special requirement arises - the stability. The stability in this context implies that the existing data structures should be insensitive to changes, which normally affect the database and the corresponding information system throughout its life-cycle. The data model may be enhanced with new elements and new relationships among the elements, but these extensions should not endanger the existing elements and relationships.

To transform the perception of the real world into a conceptual database design is the most sensitive task in development of an information system. The quality of the result heavily depends upon the expertise, intuition and experience of the designer and upon the quality of communication with the users. The final quality of the data model and its stability are a direct consequence of this task. Very often, the designer creates new objects by copying or modifying the already existing ones, used in other databases. The reuse of these properly designed and in practice confirmed elements is surely desirable.

A software tool should help the designer to appropriately select and reuse some existing database parts [5] and to enhance them easily with new creations.

The multi-meta database contains descriptions of the existing database models. It enables an easy retrieval and comparison of the elements stemming from different models. Single models contained in the multi-meta base are represented in a form of an entity-relationship model [2].

In the multi-meta base concept, a single element may belong to multiple databases[5]. It makes the difference to the meta-models applied in semantic integration of heterogeneous databases as reported in [6], [1], where each object belongs strictly to a single scheme. The intention of the proposed method is to stimulate the reuse, so the expectation is that the count of the database schemes sharing the same element should steadily grow.

Descriptions of the models contained in the multi-meta base must be non-redundant. For example, it would not be allowed to describe the connections among the entities concurrently through the definitions of relationships and through foreign keys contained in the entity definitions [4]. A non-redundant description will make the retrieval and comparison easier and it will exclude ambiguity from the model description.

The method based on the multi-meta base detects the semantic and structural similarity among the existing models. The results are applied to remove any duplicates and to define the semantic connections among elements. New models are built from a selection of the already existing ones, new
creations are added, model consistency is checked and some restructuring performed, when necessary. The schemes must be diluted into atomic parts to make the desired selections easier.

2 Multi-meta database model

2.1. Basic model of the meta-base

A model of the meta-base is defined that will contain the description of a database. The database model description consists of descriptions of entities, relationships, attributes and domains. The basic model of the meta-base is presented in Fig. 1.

The entities contain respective descriptions of database components; i.e. ENTITY contains descriptions of the entities etc., as follows:

ENTITY = {EntID, EntName, EntLongName, EntDescr, EntComment}
RELATIONSHIP = {RelID, RelName, RelLongName, RelDescr, RelComment}
ATTRIBUTE = {AtrID, AtrName, AtrLongName, AtrDescr, AtrComment}
DOMAIN = {DomID, DomName, DomLongname, DomDescr, DomComment, DomType}

The relationships REL_ENT, REL_ATT, ENT_ATT, ATT_DOM describe the respective connections among the meta-base entities:
REL_ENT = {RelID, EntID, RoleName, Connectivity, WeakEnt}
REL_ATT = {RelID, AttID, RelKeyPart }
ENT_ATT ={EntID, AttID, EntKeyPart}
ATT_DOM = {AttID, DomID}

Where:
The connectivity of a relationship specifies the mapping of the associated entity occurrences in the relationship [7].

RoleName introduces different roles of an entity in a relationship, what is of special importance in reflexive and in parallel relationships.

RelKeyPart marks an attribute as part of the relationship primary key. The allowed values are TRUE or FALSE.

EntKeyPart marks an attribute as part of the entity primary key. The allowed values are TRUE or FALSE.

2.1.1 Transformation of the meta-base

All the entities in the meta-base share a nearly equal structure. For a model analysis, it is necessary to process all the meta-base entities, regardless to their type or meaning within the model. Therefore, a generalised entity OBJECT is introduced having the entities RELATIONSHIP, ENTITY, ATTRIBUTE, DOMAIN as specialisations. The resulting model is presented in Fig. 2.

For the generalisation, new attributes are introduced: ObjType to enable partitioning and ObjID, as the object identifier. Relational schemes of the entities become:

Figure 2 Meta-base model with generalisation/specialisations

For the generalisation, new attributes are introduced: ObjType to enable partitioning and ObjID, as the object identifier. Relational schemes of the entities become:
2.2. Multi-meta model and multi-meta database

The database that will contain the descriptions of different database models is the multi meta-base, in further text referred to as the mm base. The information contained in the multi-meta base will continuously be analysed, deductions will be carried out and it might be revised. Design of a new model starts with the selection of structures from the existing models described in the mm base. These structures may be subject to revision. The model can be enhanced further on. After the completion of these procedures, the consistency of the new model is verified. The mm base is proposed to enable:

- storing of information about single database models
- deductions, made simple as possible, about similarity among the objects from different data models
- an easy structure revision of existing models, i.e. the revision of data describing the models
- a review of the existing models
- an easy choice of concepts from the existing model to help in the design of a new model

To meet the listed objectives, the redundancy should be reduced.

At the time when a multi-meta model is built, it will be attempted to substitute the concepts from a lower level of abstraction with those concepts from a higher level, which already imply some general rules. In this way, the redundancy will be reduced, the model clarity will be increased, maintenance made easier and the model analysis simplified.

The model in Fig. 3 bears a large redundancy. If an entity, together with its attributes is present in a number of different schemes, it would be described in each of these schemes. This would complicate the schema maintenance and revisions.

By analysing of single schemes, connections will be established among the objects from different planes - databases. At the instance when a connection between the objects from two different databases is established, new objects will be stored redundantly, in each of their home databases.

During the mm model design, the aim is a maximum reduction of redundancy, to make the frequent scheme revisions easy.

The redundancy can be reduced by application of the following rule:

**Rule 1**

The objects, contained in different databases, are stored independently from their home databases.

**Consequence of the Rule 1**

- a new relationship is established to connect the objects to their corresponding databases (BASE_OBJ).

**Rule 2**

A single object, uniquely defined by its identifier, has certain exactly determined semantic and structural properties. There exist no alternative structural properties for any object.

**Consequence of the Rule 2**

- the relationships contained in the relationships REL_ENT, REL_ATT, ENT_ATT and ATT_DOM are independent of their home databases.

This may mean, for example:

- to a given attribute, the same domain is attached in all the databases where this attribute is present
- a given entity has a uniquely defined relational scheme, equal for all the database models where it is present
- a given relationship has a uniquely defined structure (entities involved, the mapping, the relationship attributes) in all the database models where it is present

The model of a mm base, founded upon the rules 1 and 2 is presented in Fig. 4.
For the scope of further redundancy reduction, the interconnection of objects to the databases is considered.

Thanks to the rules 1 and 2, and because all the objects of a database are interconnected, a question arises whether all the database objects have to be explicitly connected to the database. One can suppose that connections to only certain objects are essential, whereby all the other connections can be deduced.

On the ground of an additional analysis of the multi-meta model, some facts have been found and they will be elaborated in some definitions that follow.

Let a scheme MM (or model MM) correspond to the model in Fig. 4 and let a database mm represent the current value of MM. Let mm contain different database descriptions stored in the following relations defined over their respective relational schemes:

- **BASE**
- **OBJECT**
- **RELATIONSHIP**
- **ENTITY**
- **ATTRIBUTE**
- **DOMAIN**

Abbreviations are introduced for attribute names from the model MM:

- #B database identifier (baseID)
- #O object identifier (objID)
- #R relationship identifier (relID)
- #E entity identifier (entID)
- #A attribute identifier (attID)
- #D domain identifier (domID)

**Definition 1**

Let the set \( R_B \) members be the identifiers of all the relationships contained in the database B.

\[
R_B = \{ #R_1, \ldots, #R_p \}
\]

It has to be proved that the membership of all the other objects in the database results from the set \( R_B \) and the sets \( RE, RA, EA \) and \( AD \).

The sets \( RE, RA, EA \) and \( AD \) are invariant in respect to their members’ positioning within different databases. The stored information can be reduced to the data about relationship memberships in databases.

Entities from the database B can be deduced from the sets \( R_B \) and \( RE \). The membership of all the objects in a database is completely described by the membership of relationships.

Relationship BASE_OBJ is substituted by a new relationship BASE_REL. The mapping of the relationship BASE_REL is N:N. The ultimate version of the mm model is presented in Fig. 5.

**Figure 5** The ultimate version of the mm model

The ultimate structure of the mm model is following:

- relational schemes of the entities:
  - BASE = \{ baseID, baseName, baseDescr, baseComment \}
  - OBJECT = \{ objID, objName, objLongName, objDescr, objComment, objType \}
  - ENTITY = \{ entID, objID \}
  - RELATIONSHIP = \{ relID, objID \}
  - ATTRIBUTE = \{ attID, objID \}
  - DOMAIN = \{ domID, objID, domType \}

**3 Description of the database models contained in the mm base**

**3.1 Description of the relationships**

According to [2] and [4] the structure of the relationships can be derived from the keys of the entities involved and from the corresponding
connectivity. In addition, the relationship description can contain some own attributes in the non-key part.

Definition 2
Let $R_i, R_e \in \mathcal{R}$ be a relationship of degree $m$ and let $E_k, k \in (1, \ldots, m)$ be the entities involved in $R_i$. Let $#R_i$ be the identifier of $R_i$. The identifiers of those entities which are connected through $R_i$, are defined by the set $RE_{R_i} = \{ #E_1, \ldots, #E_m \}$. The interconnection of the entities results from the relation $\mathcal{E}(\mathcal{R}_{\text{ENT}})$ of the mm base.

$$RE_{R_i} = \pi_{#E} (\sigma_{#R = #R_i} (\mathcal{E}))$$

Let $RA_{V_i} = \{ #A_1, \ldots, #A_n \}$ contain its own attributes of $R_i$. These own attributes are described in the relation $\mathcal{E}(\mathcal{R}_{\text{ATT}})$ of the mm base.

$$RA_{R_i} = \pi_{#A} (\sigma_{#R = #R_i} (\mathcal{E}))$$

For each entity $E_k, k \in (1, \ldots, m)$ a key is defined. The definition of entity keys results from the relation $\mathcal{E}(\mathcal{R}_{\text{ATT}})$ of the mm base.

Let $K_k$ be the set of identifiers of the key attributes belonging to the entity $E_k$. $K_k = \pi_{#A} (\sigma_{#E = #E_k} (RE_{R_i}))$.

Let $K$ be the set of identifiers of all the key attributes from entities $E_1, \ldots, E_m$:

$$K = \bigcup_{k=1}^{m} K_k$$

where $m$ is the degree of the relationship $R_i$.

Then, it must hold:

$$K \cap RA_{R_i} = K \cap (#A_1, \ldots, #A_n) = \emptyset$$

In other words, the set of own attributes $\{ A_1, \ldots, A_n \}$ of the relationship $R_i$ must not contain any key attributes from the entities involved in $R_i$.

Definition 3
Let $A_{R_i}$ be the set of identifiers of all the attributes of the relationship $R_i$. This set is defined by the set of its own attributes $(RA_{R_i})$ and the set of key attributes (set $K$) from the entities involved in $R_i$.

$$A_{R_i} = RA_{R_i} \cup K$$

A rule deriving from the definitions 2 and 3 can be defined:

Rule 3
In the mm base, the relationship $\mathcal{R}_{\text{ATT}}$ will contain only its own attributes. The set of all the relationship attributes consists of their own attributes and of the key attributes of the involved relationships.

3.1.1. Definition of relationship keys
For every relationship, keys are defined. This definition derives directly from the involved entities and their connectivity.

Definition 4 according to [4]
In a relationship connecting the entities $E_1, \ldots, E_k, \ldots, E_m$ connectivity “1” of an entity $E_k$ means that for any value of all the other entities $E_1, \ldots, E_m$, except $E_k$, there cannot be more than one value of $E_k$.

As the interconnection of entities in a relationship is represented by the entity keys, a functional dependency can be formulated:

$$m \bigcup_{j=1}^{m} K_j \setminus K_k \rightarrow K_k$$

where the sets $K_j$, $(j = 1, \ldots, m)$ define the keys of entities $E_1, \ldots, E_m$.

Definition 5
A relationship has at least one key. If in a relationship multiple entities with connectivity 1 are involved, the relationship will have as many keys as many of such entities are involved.

If in a relationship no entity of connectivity 1 is involved, the relationship key will consist of all the primary keys of the involved entities.

Let $VE_{V_i}$ be a set to describe the interconnection of entities through the relationship $R_i$:

$$RE_{R_i} = \sigma_{#R = #R_i} (\mathcal{E}).$$

- If for a relationship $R_i$ the condition $\sigma_{\text{connectivity} = 1} (RE_{R_i}) = \emptyset$ holds, there will be a single key:

  $$K_{R_i} = \bigcup_{j=1}^{m} K_j$$

  where $K_i$ is the key of the entity $E_i$ and $m$ is the degree of $R_i$.

- If $\sigma_{\text{connectivity} = 1} (RE_{R_i}) \neq \emptyset$, the number of keys will be

  $$\text{card} (\sigma_{\text{connectivity} = 1} (RE_{R_i})).$$

The keys of the relationship are defined: for every $E_k$, $#E_k \in \pi_{#E} (\sigma_{\text{connectivity} = 1} (RE_{R_i}))$ the relationship key is defined:

$$m \bigcup_{j=1}^{m} K_j \setminus K_k$$

where $K_k$ is the key of entity $E_k$, and $m$ is the degree of $R_i$.

From the definition 3 it results that it is possible to uniquely define the keys of a relationship $R_i$, founded upon the definition of the relationship.
Rule 2

The keys of the relationships are not explicitly stored in the \textit{mm} model. They are deduced in a procedure described by the definition 3.

3.2. Description of entities

To simplify the selection and recombination of the objects from the \textit{mm} base, it is desirable that the objects be atomic, i.e. that they are described exclusively by their own properties and that no single database object would simultaneously contain information about several semantically different objects. This requirement is especially important for the entities. Their attribute sets often contain foreign keys and their own relationship attributes for the 1:N mapping. Therefore, the following rule is defined:

Rule 7

In the \textit{mm} base in the relation \textit{EA}, defined over the relational scheme \textit{ENT_ATR}, a single entity is coupled only to its own attributes. All the attributes that in any way relate to the connections with the other entities, must be stored within the relationships.

4 Conclusion

In the \textit{mm} base, every object is connected to the other objects that define its structural properties. An attribute is connected to its domain; entities are connected to its attributes. Exceptions are the domains. They belong to the lowest semantic category and they have no lower objects to describe further their structure. Relationships are connected to the entities involved. For every entity, its connectivity in a given relationship is defined. Relationships are also connected to their own attributes.

A database model is defined by its relationships. From the relationships contained in a database and from their connections to the objects that define their structural properties, all the elements of a database model can be derived.

Single database models are described in the \textit{mm} base in a non-redundant way. For each object, the description of only its own properties is stored. All the other properties, deriving from connections to other objects, are described through connections to these objects. The entities contain only their own attributes. Foreign keys are removed from the entity descriptions. The relationships are described by their own attributes only. The identifiers for all the entities, connected through a relationship, are deduced from the definition of the relationship. The keys of relationships are not formulated explicitly, but they derive from the keys of entities involved and from their connectivity. The rules for construction of relational schemes are defined.

For models not in concordance with the described rules, the adequate prescribed transformations are performed.

References:


