A Novel Method for the Comparison of Graphical Data Models

Katarina Tomičić-Pupek *katarina.tomicic@foi.hr*
University of Zagreb, Faculty of Organization
and Informatics, Varaždin, Croatia

Vjeran Strahonja *vjeran.strahonja@foi.hr*
University of Zagreb, Faculty of Organization
and Informatics, Varaždin, Croatia

Abstract

This paper proposes a methodical approach for data model comparisons based on the graph theory. The proposed new approach is a Graphical Model Comparison Algorithm (GMCA) which includes a procedure, methods and an algorithm that can be used for comparing two data models for similarities. The comparison is based on the structural similarity of graphs representing data models that contain some semantically similar data objects.

Keywords: Data Model Comparison, Graph Theory, Semantic Similarity, Structural Similarity.

1. Introduction

Development and implementation of information systems is reliant on modeling. Two factors – reusability and integration play a significant role in the process of system implementation [9]. Various process, data, and business models are used to align software and organizational structures. In the case that software and organizational structures do not align well, software customization or the redesign of the organizational structure are reasonable alternatives. In order to reuse and implement existing data structures and applications, the compliance of referent and target implementation domains must be achieved, which is best checked by the use of corresponding models. Similarly, for integration, the compatibility of implementation co-domains and interoperability of their information systems must be checked on the appropriate models.

The research presented in this paper proposes that the alignment, compliance, compatibility, and other related properties of business problem domains and their IT co-domains, (which are used in different contexts), can be identified, checked and proven by using the similarity of their corresponding models. Given their importance and stability, we will focus on the data structures and models.

Similarities between IT systems can be defined through the examination of their relevant characteristics. One of these characteristics can be expressed as a measure of the semantic and structural resemblance of data objects used in different IT systems. Why is the semantic and structural resemblance of data objects of interest? Firstly, data objects are related to business rules; since many business domains and systems share similar business rules, they handle similar data sets, which are subsequently used in a similar way in various organizations. This proposition is relevant to the reusability of IT systems. Secondly, rapidly changing technology and business environments call for the continuous improvement of IT systems at various levels: technical, data, process, and business. The existing IT systems need to be upgraded, thus, making the integration possibilities of new and existing data structures very important. Thirdly, graphical data models of IT systems, which comprise relationships between the data objects (thereby representing the business rules), are necessary and useful for identifying and analyzing data relations. Since the structure, constraints, and operations representing data objects or concepts in data models are similar to that of the nodes and edges in graphs, it can be assumed that a comparison of data models based on graph theory is possible.

There are a few formal methods that can be considered as suitable for the graphical comparison of data models [10], [12]. Additional studies have compared the conceptual models of data using descriptive logic [4], and the application of the B-method for comparing relational UML [7]. The concept for the presentation of data models using graphs was researched by many authors [1,2], [5,6] however, there is a lack of any published algorithm or tool to support graphical representation of a data model. On the other hand, some research has focused on investigating and developing models for measuring semantic distance to quantify the distance between two data values using a graph-based approach [8], or determining semantic and schematic similarities between objects in databases, based on a contextual analysis [3].

Influenced by these ideas, the research presented in this paper proposes a new methodical approach. This approach includes procedures that can be used as a tool for comparing two data models for their similarities based on their structure and accounting for the semantic similarity of some data objects.

The fundamental concept is to compare the graphical models as they are (i.e. as graphical representation), without their prior conversion to formal specifications. This implies the use of their basic shape as graphs and the application of graph theory. Despite the development of graph theory and its applications, preliminary research has shown that there are no adequate "ready-to-use" methods and algorithms for this specific purpose. Thus, a new method and algorithm for comparison of graphical data models needs to be defined and implemented.

The new methodological approach explained in this paper is manifested in the fact that the models are compared simultaneously by two criteria: the semantically and structurally, but in one pass through the algorithm and application of the algorithm in a selected tool. Structural comparison is performed only over the contextually/semantically similar elements with the help of directed arcs that describe the use of foreign keys (in relational terms).

2. Graphical Model Comparison: Methods and Process

Before discussing the methods and procedure, the limitations of this method must be explained. These limitations are concerned with the input graphical models for comparison and can be expressed by the following question: When does the comparison of graphical models make sense?

First of all, it should be determined whether graphical models are conceptually comparable, i.e., if they are of the same type and generated on the basis of the same representation concepts. If not, they must be transformed via modeling methods into models that meet this condition. Although our research is based on the Entity-Relationship (E-R) method of graphical representation, the implications and conclusions of our research can be applied to other methods of data modeling with similar concepts and notation.

The next step is to analyze whether models are semantically comparable, i.e., if they contain synonyms or semantically analogous data concepts. Models of a similar shape may describe different matters, which are incomparable, such as the production of fishing nets and the issuance of a loan. In this example, the comparison makes no sense because both models have no semantic analogy. If data models refer to comparable problem domains, then they contain pairs of semantically similar objects. These pairs must be determined first as a prerequisite for further structural analysis.

Determination of the semantic similarity between terms, i.e. semantic relatedness, is essential for various tasks, such as clustering, information retrieval, and synonym extraction, and extends across numerous fields. There are several approaches and methods of determining semantic similarity, but also software tools, which is especially important in order to automate the whole process.

Given the small number of concepts that contain data models (<100 data objects), it is sufficient to use an intuitive method of visualizing the semantic similarity of terms, such as mind map or concept map.

Data models with the previously described pairs of semantically similar data objects (entity types) can be compared in order to explore and establish the structural similarity between them. This method makes sense if the two diagrams have at least two pairs of semantically similar types of entities and at least one relationship between them.

The basic procedure for the application of this method is below:

1. Define pairs of semantically similar data objects from the data models selected for comparison.
2. Translate E-R data models to directed graphs.
3. Create adjacency matrices from directed graphs.
4. Use semantically similar object pairs as permutations for enumerating vertices and extracting adjacency sub matrices.
5. Compare sub matrices and find matching values in matching rows and columns.
6. Interpret matching values as structurally similar relationships.

The process of comparison should include transforming the data models into directed graphs, comparison charts, and the conclusion based on determining the structural similarity between the elements of the graphs.

If it is determined that structural similarity of the directed graphs exists, then it is assumed that the models from which they were derived are also structurally similar.

Based on this assumption, a comparison process and an algorithm for comparing models to identify structural similarities is developed and implemented.

The comparison process is carried out in three phases with a total of 18 specific steps. The process is described in Figure 1. As shown in Figure 1, the steps are interconnected in such a way that the results of one or more steps are used in the next step. The next section gives a brief overview of the Graphical Model Comparison Methodology with an illustration of the algorithm on two graphical data model examples.

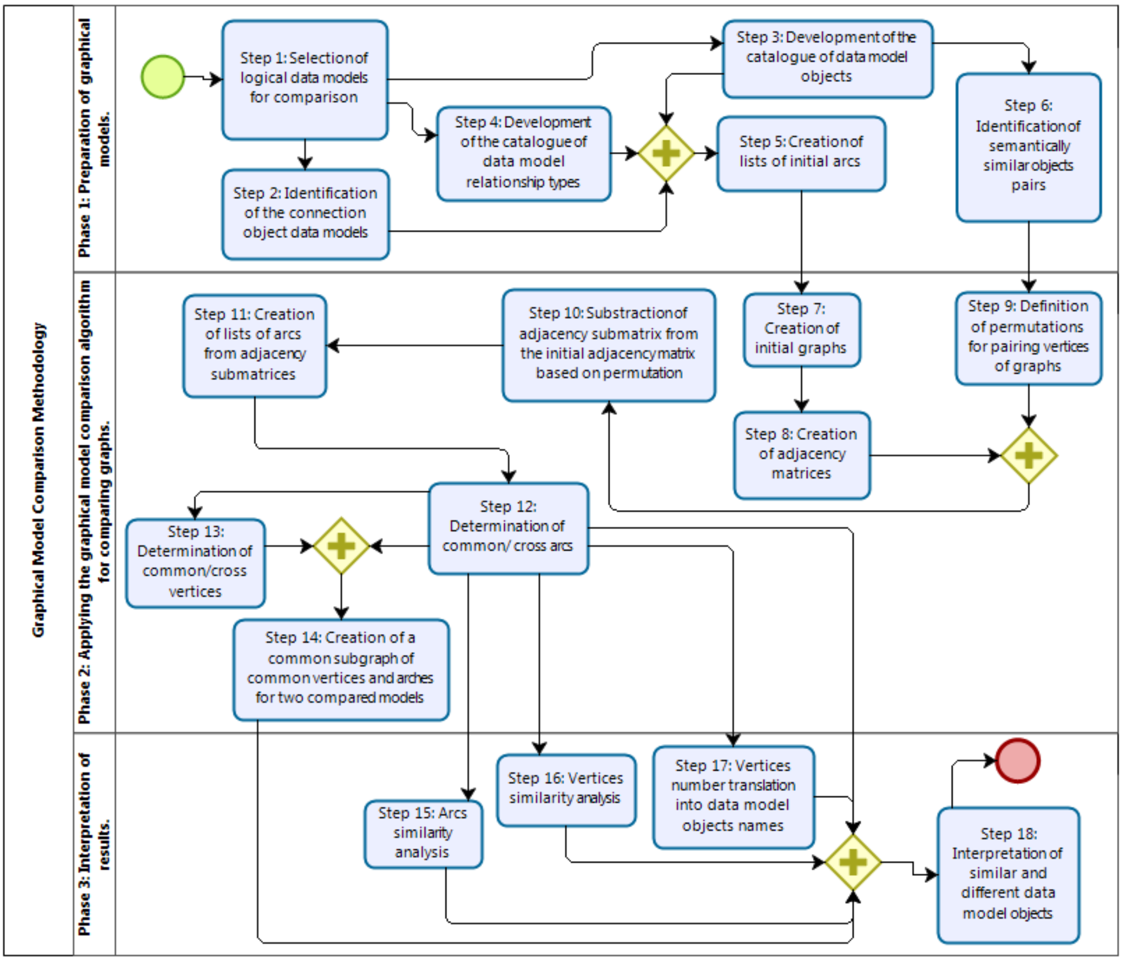
Graphical Model Comparison Methodology
Phase 1: Preparation of graphical models.
Phase 2: Applying the graphical model comparison algorithm for comparing graphs.
Phase 3: Interpretation of results.
Step 1: Selection of logical data models for comparison
Step 2: Identification of the connection object data models
Step 3: Development of the catalogue of data model objects
Step 4: Development of the catalogue of data model relationship types
Step 5: Creation of lists of initial arcs
Step 6: Identification of semantically similar objects pairs
Step 7: Creation of initial graphs
Step 8: Creation of adjacency matrices
Step 9: Definition of permutations for pairing vertices of graphs
Step 10: Substraction of adjacency submatrix from the initial adjacency matrix based on permutation
Step 11: Creation of lists of arcs from adjacency submatrices
Step 12: Determination of common/ cross arcs
Step 13: Determination of common/cross vertices
Step 14: Creation of a common subgraph of common vertices and arches for two compared models
Step 15: Arcs similarity analysis
Step 16: Vertices similarity analysis
Step 17: Vertices number translation into data model objects names
Step 18: Interpretation of similar and different data model objects

Fig. 1. Graphical Model Comparison Process

3. Illustration of the comparison algorithm on two graphical data model examples

Based on the methodology for comparison of graphical data models, an algorithm named Graphical Model Comparison Algorithm (GMCA) was developed and applied. The applicability of the GMCA to Entity-Relationship data models and related data models which are based on binary relations and graphic notation was tested. The algorithm was written in an open source mathematical tool called wxMaxima [11].

3.1. Phase 1: Preparation of graphical models

In order to show how the algorithm works, each step of the methodology is explained and illustrated with an example comparing two data models. The initial phase of the comparison procedure for determining structural similarity two data model examples were selected (Figures 2 and 3). The first phase consists of following steps: selection of logical data models for comparison, identification of the connection object data models, development of the catalogue of data model objects, development of the catalogue of data model relationship types, creation of lists of initial arcs, and identification of semantically similar objects pairs.

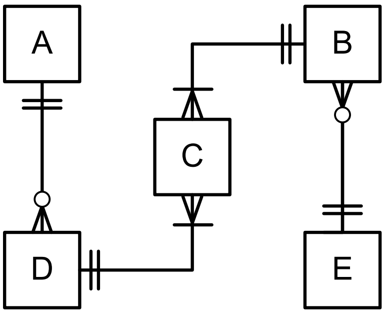

Fig. 2. Example data model 1

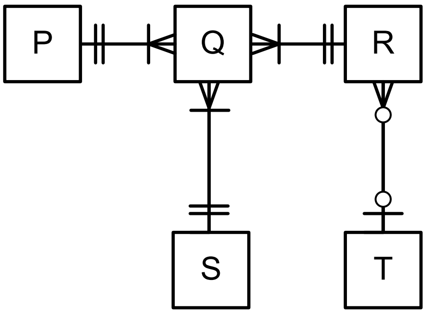

Fig. 3. Example data model 2

For each data model, a list of relationships between objects must be created containing the elements in the following format: [[Source object, target object], relationship type]. Source object refers to the object whose instances are referenced in the target object instances as many times as it is stated in the relationship type.

Data models can contain different relationship types. A subset of common relationship types (1 to many, 1 to 1, many to many) which have been identified as possible relationship types in selected data models, are coded by a number. Assigned codes for relationship types are following: relationship type (0,1):(0,M) is coded by the number 1; relationship type (1,1):(0,M) is coded by 2; relationship type (0,1):(1,M) is coded as 3; and relationship type (1,1):(1,M) is coded by the number 4. This code will be used to describe the weight of arches in the directed graphs.

In order to make the graphs comparable, first identify the pairs of objects which have a semantic similarity, e.g., if object B in data model 1 represents a Client and object R represents Buyer in data model 2, then it can be assumed that these two objects could represent a semantically similar concept in two different data models. In the same way, other semantically similar object pairs are presupposed. These pairs are listed in Table 1. Each pair is coded by a number in the first column. This code number will be used to define permutations of vertices.

Table 1. List of semantically similar pairs of objects

Pair number	Object from data model 1	Object from data model 2
1	E	T
2	D	P
3	C	Q
4	B	R

3.2. Phase 2: Applying the graphical model comparison algorithm for comparing graphs

List of connection from data models 1 and 2 will be used to create and compare lists of vertices and arcs describing graphs derived from data models. These lists are inputs for the algorithm, which will be used to determine the structural similarity of data models.

The second phase of the comparison consists of eight steps, which allow the creation of initial data model corresponding directed graphs (Figures 4 and 5), the creation of adjacency matrices, definition of permutations for pairing vertices of graphs, subtraction of the adjacency submatrix from the initial adjacency matrix based on permutation, the creation of lists of arcs from adjacency sub matrices, determination of common/cross arcs, determination of common/cross vertices and finally the creation of a common subgraph of common vertices and arcs for two compared models (Figure 6).

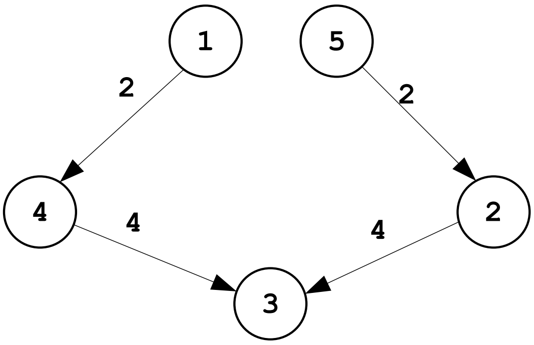

Fig. 4. Graph 1 created from data model 1

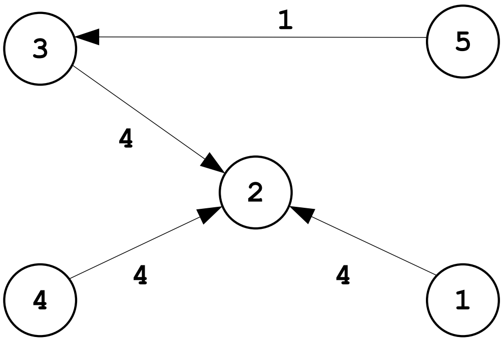

Fig. 5. Graph 2 created from data model 2

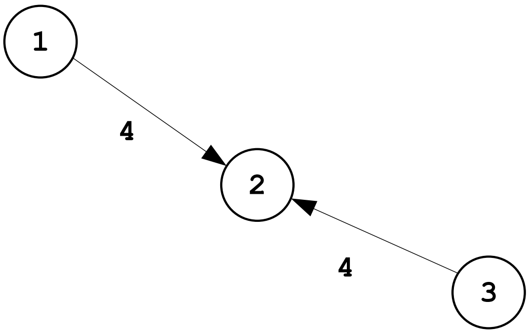

Fig. 6. Common subgraph of common vertices and arcs for two compared models

3.3. Phase 3: Interpretation of results

In this phase, the results are interpreted. This phase consists of four steps: Arcs similarity analysis, vertices similarity analysis, vertices number translation into data model objects' names and the Interpretation of similar and different data model objects. This last step of the methodology refers to the interpretation of the data object similarity by listing data objects for which structural similarity is recognized (if their corresponding vertices are in the common subgraph) and those that are not included in the set of similar data objects.

Comparison results for two compared models are represented by Figure 6 and the matrix from graph intersection are following.

```
┌         ┐
│ 0  4  0 │
│ 0  0  0 │
│ 0  4  0 │
└         ┘        with following measures:
RES: Arc similarity from 1st graph/data model to 2nd :50
RES: Arc similarity from 2nd graph/data model to 1st :50
RES: Vertices similarity from 1st graph/data model to 2nd :60
RES: Vertices similarity from 2nd graph/data model to 1st :60
SIM: Vertices of 1st graph/data model in intersection:[4,3,2]
SIM: Names of vertices of 1st graph/data model in intersection:
[D,C,B]
DIFF: Vertices of 1st graph/data model NOT in intersection:[1,5]
DIFF: Names of vertices of 1st graph/data model NOT in intersection:
[A,E]
SIM: Vertices of 2nd graph/data model in intersection:[1,2,3]
SIM: Names of vertices of 2nd graph/data model in
intersection:[P,Q,R]
DIFF: Vertices of 2nd graph/data model NOT in intersection:[4,5]
DIFF: Names of vertices of 2nd graph/data model NOT in intersection:
[S,T]
```

4. Conclusion

The alignment, compliance, compatibility and other related properties of business problem domains and their IT co-domains, which are used in different contexts, can be identified, checked and proven by using the similarity of their corresponding data models. This paper presents an overview of a methodology and proposes a process for comparing the structural similarity of two data models with semantically similar elements.

The fundamental idea is to compare the graphical models as they are (i.e. as graphical representation), without their prior conversion to formal specification. This means that it is not important which notation (E-R notation or any other) is being used to show relations between objects, but it is essential to include in the comparison the pure existence of relations (to answer the question are two objects related, and is this relation directed). Due to the fact that different types of relations can be seen between objects (e.g. 1:M is a type of relation in E-Rs) the relationship types are being coded for the purpose of the comparison and it cannot be seen as a restriction of the algorithm. Basically, if a relationship between to objects can be described as a directed arc (one or two arcs for bidirectional relations) than there is no need for model translation into another notation). This concept implies the use of their basic shape as graphs and the application of graph theory.

Preliminary research has shown that there are no adequate “ready-to-use” methods and algorithms for this specific purpose. To meet this need a novel method and algorithm was developed, the Graphical Model Comparison Algorithm (GMCA). In order to show how the algorithm works, the algorithm was illustrated with an example of the comparison of two data models.

It is the conclusion of this research that it is possible to determine the similarity of two data models, based on the semantic similarity of pairs of data objects they contain and structural similarity of graphs, by using the proposed procedure and GMCA.
The secondary results of the application of GMCA are differences between compared models.

The applicability of GMCA to Entity-Relationship data models and the related data models which are based on binary relations and graphic notation has currently been confirmed.

During the process of conducting this research, areas that require further studies have been revealed including:

- The comparison of different versions of the same model;
- Performance indicators of the algorithm (like complexity, scalability and run-time);
- Comparison of the “as is” and “to be” models of various kinds;

- Development of a procedure for comparison of more than two data models, based on the GMCA.

Additionally, this research suggests that the application of the methodological approach to process models (which has partially been tested and shown in other papers like "Comparison of simple graphical process models" in Journal of Information and Organizational Sciences), state machines, and other models based on the graphic notation, is a subject for future research.

References

1. Choudhury, S., Chaki, N., Bhattacharya, S.: GDM: a new graph based data model using functional abstraction. Journal of Computer Science and Technology. 21(3), 430-438 (2006)
2. Gyssens, M., Paredaens, J., Van Gucht, D.: A graph-oriented object database model. In: Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS '90), pp. 417-424. ACM, New York, NY, USA (1990)
3. Kashyap, V., Sheth A.: Semantic and Schematic Similarities between Database Objects: A Context-based approach. The VLDB Journal – The International Journal on Very Large Data Bases, 5(4), 276-304 (1996)
4. Keet, C.M.: A formal comparison of conceptual data modeling languages. 13th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'08), Vol-337, pp. 25-39. Montpellier, France (2008)
5. Kim, P.: E-model: event-based graph data model theory and Implementation, PhD Thesis. Georgia Institute of Technology, Atlanta Georgia, USA (2009)
6. Majundar, A.K., Bhattacharya, I., Saha, A.K.: An object-oriented fuzzy data model for similarity detection in image databases. IEEE Transactions on Knowledge and Data Engineering, 14(5), 1186-1189 (2002)
7. Mammar, A., Laleau, R.: From a B formal specification to an executable code: application to the relational database domain. Information and Software Technology, 48(4), 253-279 (2006)
8. Roddick, J.F., Hornsby, K., De Vries, D.: A unifying semantic distance model for determining the similarity of attribute values. In: Proceedings of the 26th Australasian computer science conference - Volume 16 (ACSC '03), Michael J. Oudshoorn (Ed.), pp. 111-118. Australian Computer Society, Inc., Darlinghurst, Australia (2003)
9. Saha, B., Stanoi, I., Clarkson, K.L.: Schema Covering: a Step Towards Enabling Reuse in Information Integration. In: Proceedings of ICDE Conference 2010, pp. 285-296, IEEE, Long Beach, CA (2010)
10. Vatanawood, W., Rivepiboon, W.: Formal Specification Synthesis for Relational Database Model. International journal of intelligent systems, 19, 159-175 (2004)
11. wxMaxima, (http://andrejv.github.com/wxmaxima/index.html, Accesses October 10th 2011
12. Yugopuspito, P., Araki, K.: Transformational Object-Relational Database Model in Formal Methods. Transactions of Information Processing Society of Japan, 42(5), 71-80 (2001)