Generic Model in Modelling Analytical Data Model

Mladen Varga
Faculty of Economics & Business – Zagreb
Trg J. F. Kennedya 6, Zagreb, Croatia
mvarga@efzg.hr

Abstract. *The paper presents an algorithm for generating analytical data model from collected business data requirements, which are represented by entities of interest, their associations and attributes. The requirements are described in generic form, which is capable to accept any model regardless their purpose. The algorithm finds proper entities which serve as analytical dimensions and finally generate analytical or dimensional data model.*

Keywords. Generic model, analytical data modelling, multidimensional data modelling

1. Introduction

It is supposed that business data requirements are collected and represented by a set of entities of interest, their associations and attributes; and represented by a data model in appropriate graphical representation, such as entity-relationship (ER) [5] or class diagram [4]. The data model may be represented in classical ER representation but the better and easier way is using generic model structure. The generic model structure [3] is used as the structure capable to describe any data model regardless of its purpose.

The aim of the paper is to sketch an algorithm of conversion an initial business data model into the data model which is capable to fulfil analytical data processing requirements. The problem is considered [6, 7] in last decade but the generic model structure wasn't used in representing input data model.

The basic terms are introduced in Section 2, algorithm for generating analytical data model is presented in Section 3, and possible usage of generated algorithm is discussed in Section 4.

2. Basic terms

An entity is a thing or object of significance, whether real or imaginated, about which information needs to be known or held [1]. A strong entity, such as Person, Product and Course, exists independently of any other entity.

Association provides a means of linking various entities in a meaningful way [2]. For example, the entities Person and Organization are associated. The connection between two (or more) entities in both directions is called relationship and can be considered as an associative (or intersection) entity, which is a weak entity. Entity-relationship theory distinguishes unary (special type of binary), binary and n-ary relationship. For our purpose it is sufficient to identify child-parent associations; or in relational database terminology functional dependencies from child entities to parent entities. These are inter-entities functional dependencies, which are used to describe inter-entities structure in generic model.

An attribute is a descriptive property of an entity, both strong and associative. Attributes may be of the type ID (an identifier or key attribute), DESC (descriptive non-measure nominal or ordinal attribute), and MEASURE (additive, semi-additive or non-additive measure attribute).

Generic model paradigm [3] is used for describing initially collected data model and also generated analytical data model. Some of the important principles of the generic model concept are [3]: (1) Entity types should represent, and be named after, the underlying nature on an object, not the role they play in a particular context; (2) Activities and associations should be represented by entity types (not relationships); (3) Candidate attributes should be suspected of representing relationships to other entity types. The generic model structure that is used to describe initial business data model requirements is represented in Figure 1.

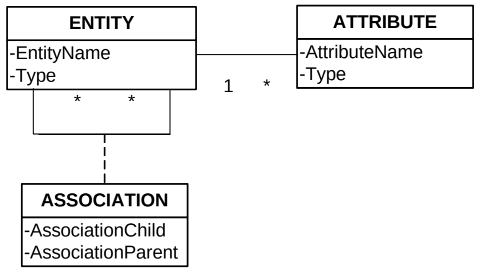

Figure 1. Generic model

3. Algorithm for conversion of input data model into analytical data model

The idea of the algorithm is to convert the input business data model into an analytical (dimensional) data model suitable for dimensional data analysis used in data warehousing.

The first steps, A and B, of the algorithm are model requirements steps in which the collected business requirements, i.e. business rules, are modelled through entities, associations and attributes. This initial business data model may be operational, analytical, or even unknown.

In the next steps, D1 to D6, the proposed algorithm converts the initial model into the analytical (dimensional) model.

3.1. Model requirements steps

A. Define initial list of strong entities and their attributes

In this step an initial model is settled by a list of strong entities and their attributes. The entities, important to the business and as they are used by users, are recognized. All their attributes must be inherent, i.e. there must be no attribute that represents an association of an entity to another entity. In this step the attributes of all entities, both strong and associative, have to be defined.

The results of the algorithm are illustrated by the simplified hospital example consisting of two hospital processes: Patient examination after patient's coming to hospital and Patient transfer to ward after his/her medical examination. The Patient examination process' entities are PATIENT (patient's data), TOWN (data of the town which is patient coming from), ADMISSION (patient's admission to the hospital data), DOCTOR (data of the doctor who diagnoses patient) and DIAGNOSIS (international classification of diseases). The Patient transfers to ward process' entities are PATIENT, ADMISSION, DOCTOR (data of the doctor who will cure the patient) and WARD (data of the ward where the patient is transferred). Table 1 shows the initial list of strong entities of the hospital example.

Table 1. Step A: initial list of strong entities

ENTITY NAME	TYPE
PATIENT	STRONG
TOWN	STRONG
ADMISSION	STRONG
WARD	STRONG
DOCTOR	STRONG
DIAGNOSIS	STRONG

The attributes of initial strong entities are shown in Table 2.

Table 2. Step A: list of attributes of initial strong entities

ATTRIBUTE NAME	TYPE
PATIENT.Patient#	ID
PATIENT.Name	DESC
PATIENT.Sex	DESC
PATIENT.Age	DESC
TOWN.Town	ID
TOWN.County	DESC
ADMISSION.Admission#	ID
ADMISSION.AdmissionDate	DESC
ADMISSION.AdmissionType	DESC
WARD.Ward#	ID
WARD.WardName	DESC
WARD.CostPerDay	MEASURE
DOCTOR.Doctor#	ID
DOCTOR.DoctorName	DESC
DIAGNOSIS.Diagnosis#	ID
DIAGNOSIS.DiagnosisName	DESC
DIAGNOSIS.DiagnosisCategory	DESC

B. Define association between strong entities including associative entities

In this step all associations between strong entities have to be defined. An association is the connection between entities which we are interested in.

All associations will be depicted by inter-entity functional dependencies. For all binary relationships with relationship attributes and for all n-ary relationships an associative (or intersection) entity has to be opened. Functional dependencies from associative entity to all strong entities involved in the relationship serve as inter-entity functional dependencies. Also, all attributes of the associative entity have to be added to the list of attributes.

The associations used in the example are as follows. Each ADMISSION entity represents data of a patient's admission to hospital. Therefore, the association from ADMISSION to PATIENT is represented by functional dependency ADMISSION$\rightarrow$PATIENT, where symbol $\rightarrow$ shows functional dependency from ADMISSION to PATIENT. In entity-relationship terminology this is M:1 relationship between ADMISSION and PATIENT, where ADMISSION is child entity and PATIENT is parent entity. The admission activity is performed by one DOCTOR (ADMISSION→DOCTOR). The patient comes from one town (PATIENT→TOWN).

The fact that the patient is diagnosed is described by an association entity ADMISSION_DIAGNOSIS with associations ADMISSION_DIAGNOSIS→ADMISSION, ADMISSION_DIAGNOSIS→DOCTOR (who diagnoses patient) and ADMISSION_DIAGNOSIS →DIAGNOSIS. The ADMISSION_DIAGNOSIS entity represents a ternary relationship between entities ADMISSION, DOCTOR and DIAGNOSIS. The attributes of ADMISSION_DIAGNOSIS entity are DiagnosisType and DiagnosisDate.

The fact that the patient is transferred to ward is described by association entity ADMISSION_WARD with associations ADMISSION_WARD→ADMISSION, ADMISSION_WARD→WARD and ADMISSION_WARD→DOCTOR. The attributes of ADMISSION_WARD entity are FromDate and ToDate.

Table 3. Step B: list of associations

ASSOCIATION CHILD	ASSOCIATION PARENT
PATIENT (comes from)	TOWN
ADMISSION (belongs)	PATIENT
ADMISSION (is done by)	DOCTOR (admission)
ADMISSION_WARD	ADMISSION
ADMISSION_WARD	WARD
ADMISSION_WARD	DOCTOR (in cure)
ADMISSION_DIAGNOSIS	ADMISSION
ADMISSION_DIAGNOSIS	DOCTOR (diagnosis)
ADMISSION_DIAGNOSIS	DIAGNOSIS

Two associative entities, ADMISSION_WARD and ADMISSION_DIAGNOSIS, are added to the list of entities as ASSOC entities, as shown in Table 5. Their associations are shown in grey in Table 3. Their attributes are also added to the list of attributes, as it is shown in grey in Table 4.

At the end of model requirements steps the initial business data model is described by Tables 5, 3 and 4, which are implementation of the generic model from Fig. 1. We may conclude that the initial data model shown in Fig. 2 is suitable for operational or transactional purposes.

3.2. Analytical modelling steps

D1. Define candidate dimension attributes

Previously defined attributes are now examined. Attributes candidates for generating new strong entity, i.e. new dimension in the analytical model, are distinguished. This is manual task in which the analyst concludes which DESC attribute is capable to generate new strong or dimensional entity.

Candidate dimension attributes in the example are Sex, Age, AdmissionDate, DiagnosisCategory, FromDate, ToDate and DiagnosisDate, depicted by ✓ symbol in Table 4.

Table 4. Steps B and D1: list of attributes

ATTRIBUTE NAME	TYPE	CD
PATIENT.Patient#	ID	
PATIENT.Name	DESC	
PATIENT.Sex	DESC	✓
PATIENT.Age	DESC	✓
TOWN.Town	ID	
TOWN.County	DESC	
ADMISSION.Admission#	ID	
ADMISSION.AdmissionDate	DESC	✓
ADMISSION.AdmissionType	DESC	
WARD.Ward#	ID	
WARD.WardName	DESC	
WARD.CostPerDay	MEASURE	
DOCTOR.Doctor#	ID	
DOCTOR.DoctorName	DESC	
DIAGNOSIS.Diagnosis#	ID	
DIAGNOSIS.DiagnosisName	DESC	
DIAGNOSIS.DiagnosisCategory	DESC	✓
ADMISSION_WARD.FromDate	DESC	✓
ADMISSION_WARD.ToDate	DESC	✓
ADMISSION_DIAGNOSIS.DiagnosisType	ID	
ADMISSION_DIAGNOSIS.DiagnosisDate	DESC	✓

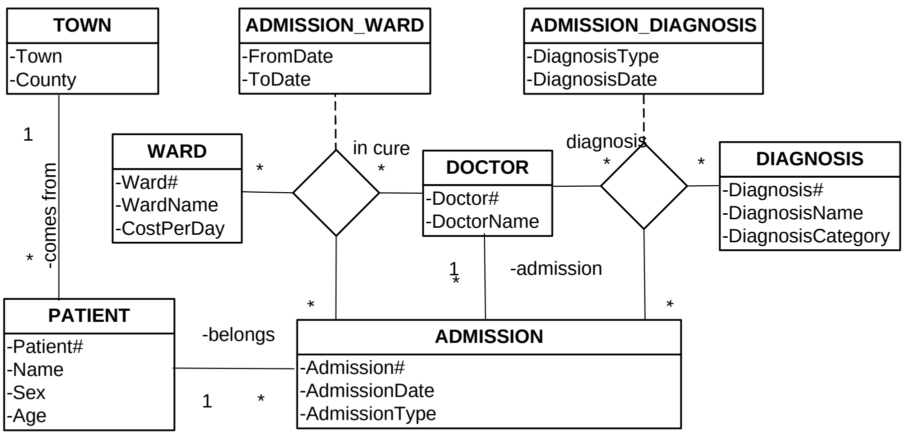

Figure 2. Initial business data model

D2. Define new entities from candidate dimension attributes

Attributes detected as candidate dimension attributes are the bases for generating new strong (dimensional) entities. Each candidate dimension attribute becomes the key attribute (identifier) in the new entity, which may be named after it. The same attribute must be pruned from the entity in which it was detected as candidate dimension attribute. Also, a new association must be established (see step D3).

If it is necessary to describe new entity in detail, new attributes may be manually added to the new entity.

The result of step D1 of the example is shown in Table 5 and Table 6.

Table 5. Step D2: extended list of entities

ENTITY NAME	TYPE
PATIENT	STRONG
TOWN	STRONG
ADMISSION	STRONG
WARD	STRONG
DOCTOR	STRONG
DIAGNOSIS	STRONG
ADMISSION_WARD	ASSOC
ADMISSION_DIAGNOSIS	ASSOC
SEX	STRONG
AGE	STRONG
ADMISSION_DATE	STRONG
DIAGNOSIS_CATEGORY	STRONG
FROM_DATE (admission on ward)	STRONG
TO_DATE (admission on ward)	STRONG
DIAGNOSIS_DATE	STRONG

Table 6. Step D2: list of attributes

ATTRIBUTE NAME	TYPE
PATENT.Patient#	ID
PATIENT.Name	DESC
SEX.Sex	DESC
AGE.Age	DESC
TOWN.Town	ID
TOWN.County	DESC
ADMISSION.Admission#	ID
ADMISSION_DATE.AdmissionDate	DESC
ADMISSION.AdmissionType	DESC
WARD.Ward#	ID
WARD.WardName	DESC
WARD.CostPerDay	MEASURE
DOCTOR.Doctor#	ID
DOCTOR.DoctorName	DESC
DIAGNOSIS.Diagnosis#	ID
DIAGNOSIS.DiagnosisName	DESC
DIAGNOSIS_CATEGORY.DiagnosisCategory	DESC
FROM_DATE.FromDate (admission on ward)	DESC
TO_DATE.ToDate (admission on ward)	DESC
ADMISSION_DIAGNOSIS.DiagnosisType	ID
DIAGNOSIS_DATE.DiagnosisDate	DESC

D3. Define new association based on new entities

New generated entities must be associated to the existing entities. From each attribute detected as candidate dimension in the previous step D2 a new association must be established from the new to the existing entity.

New associations detected in the example are shown in grey in Table 7.

Table 7. Step D3: new associations based on new entities

ASSOCIATION CHILD	ASSOCIATION PARENT
PATIENT (comes from)	TOWN
ADMISSION (belongs)	PATIENT
ADMISSION (is done by)	DOCTOR (admission)
ADMISSION_WARD	ADMISSION
ADMISSION_WARD	WARD
ADMISSION_WARD	DOCTOR (in cure)
ADMISSION_DIAGNOSIS	ADMISSION
ADMISSION_DIAGNOSIS	DOCTOR (diagnosis)
ADMISSION_DIAGNOSIS	DIAGNOSIS
PATIENT	SEX
PATIENT	AGE
ADMISSION	ADMISSION_DATE
DIAGNOSIS	DIAGNOSIS_CATEGORY
ADMISSION_WARD	FROM_DATE
ADMISSION_WARD	TO_DATE
ADMISSION_DIAGNOSIS	DIAGNOSIS_DATE

D4. Define new association based on transition rule

New associations also may be generated by performing the transition rule on the set of previously defined associations. By transition rule, if A→B and B→C then A→C. The transition rule is performed on all ASSOCIATION CHILD → ASSOCIATION PARENT functional dependencies.

New associations generated in the example by transition rule are shown in grey in Table 8.

D5. Unite identical entities

After generating, some of the new entities may be similar or the same, and may be united into one entity. In step D5 similar entities have to be detected and united.

In most cases this will appear in variants of the entity DATE, for example entities ADMISSION_FROM_DATE and ADMISSION_TO_DATE can be substituted by one entity DATE.

In the example, the entities ADMISSION_DATE, FROM_DATE, TO_DATE and DIAGNOSIS_DATE which are variants of date may be united in the entity DATE. At the same time the key attributes (identifiers) and all non-key attributes as well of united entities must be united.

Table 8. Step D4: new associations based on transition rule

ASSOCIATION CHILD	ASSOCIATION PARENT
PATIENT (comes from)	TOWN
ADMISSION (belongs)	PATIENT
ADMISSION (is done by)	DOCTOR (admission)
ADMISSION_WARD	ADMISSION
ADMISSION_WARD	WARD
ADMISSION_WARD	DOCTOR (in cure)
ADMISSION_DIAGNOSIS	ADMISSION
ADMISSION_DIAGNOSIS	DOCTOR (diagnosis)
ADMISSION_DIAGNOSIS	DIAGNOSIS
PATIENT	SEX
PATIENT	AGE
ADMISSION	ADMISSION_DATE
DIAGNOSIS	DIAGNOSIS_CATEGORY
ADMISSION_WARD	FROM_DATE
ADMISSION_WARD	TO_DATE
ADMISSION_DIAGNOSIS	DIAGNOSIS_DATE
ADMISSION_WARD	PATIENT
ADMISSION_WARD	DOCTOR (admission)
ADMISSION_WARD	ADMISSION_DATE
ADMISSION_WARD	TOWN
ADMISSION_WARD	SEX
ADMISSION_WARD	AGE
ADMISSION_DIAGNOSIS	PATIENT
ADMISSION_DIAGNOSIS	DOCTOR (admission)
ADMISSION_DIAGNOSIS	ADMISSION_DATE
ADMISSION_DIAGNOSIS	DIAGNOSIS_CATEGORY
ADMISSION_DIAGNOSIS	TOWN
ADMISSION_DIAGNOSIS	SEX
ADMISSION_DIAGNOSIS	AGE

The results of uniting the entities are shown in Tables 9 and 10.

Table 9. Step D5: list of entities after uniting entities

ENTITY NAME	TYPE
PATIENT	STRONG
TOWN	STRONG
ADMISSION	STRONG
WARD	STRONG
DOCTOR	STRONG
DIAGNOSIS	STRONG
ADMISSION_WARD	ASSOC
ADMISSION_DIAGNOSIS	ASSOC
SEX	STRONG
AGE	STRONG
DATE	STRONG
DIAGNOSIS_CATEGORY	STRONG

D6. Transition of measure attributes from strong entities into association entities

In previously defined strong entities some new attributes may be the MEASURE attributes. They have to be transited to the association entity which is connected to. Thus, if association entity associates strong entity with MEASURE attribute, then MEASURE attributes have to be transited to this associative entity.

Table 10. Step D5: associations after uniting entities

ASSOCIATION CHILD	ASSOCIATION PARENT
PATIENT (comes from)	TOWN
ADMISSION (belongs)	PATIENT
ADMISSION (is done by)	DOCTOR (admission)
ADMISSION_WARD	ADMISSION
ADMISSION_WARD	WARD
ADMISSION_WARD	DOCTOR (in cure)
ADMISSION_DIAGNOSIS	ADMISSION
ADMISSION_DIAGNOSIS	DOCTOR (diagnosis)
ADMISSION_DIAGNOSIS	DIAGNOSIS
PATIENT	SEX
PATIENT	AGE
ADMISSION	DATE (admission date)
DIAGNOSIS	DIAGNOSIS_CATEGORY
ADMISSION_WARD	DATE (admission to ward from date)
ADMISSION_WARD	DATE (admission to ward to date)
ADMISSION_DIAGNOSIS	DATE (diagnosis)
ADMISSION_WARD	PATIENT
ADMISSION_WARD	DOCTOR (admission)
ADMISSION_WARD	DATE (admission date)
ADMISSION_WARD	TOWN
ADMISSION_WARD	SEX
ADMISSION_WARD	AGE
ADMISSION_DIAGNOSIS	PATIENT
ADMISSION_DIAGNOSIS	DOCTOR (admission)
ADMISSION_DIAGNOSIS	DATE (admission date)
ADMISSION_DIAGNOSIS	DIAGNOSIS_CATEGORY
ADMISSION_DIAGNOSIS	TOWN
ADMISSION_DIAGNOSIS	SEX
ADMISSION_DIAGNOSIS	AGE

In the example, shown in Table 11, the measure attribute WARD.CostPerDay in transited to ADMISSION_WARD.CostPerDay.

Table 11. Step D6: list of attributes after transition of measure attributes

ATTRIBUTE NAME	TYPE
PATIENT.Patient#	ID
PATIENT.Name	DESC
SEX.Sex	DESC
AGE.Age	DESC
TOWN.Town	ID
TOWN.County	DESC
ADMISSION.Admission#	ID
ADMISSION.AdmissionDate	DESC
ADMISSION.AdmissionType	DESC
WARD.Ward#	ID
WARD.WardName	DESC
ADMISSION_WARD.CostPerDay	MEASURE
DOCTOR.Doctor#	ID
DOCTOR.DoctorName	DESC
DIAGNOSIS.Diagnosis#	ID
DIAGNOSIS.DiagnosisName	DESC
DIAGNOSIS_CATEGORY.DiagnosisCategory	DESC
ADMISSION_WARD.Date	DESC
ADMISSION_DIAGNOSIS.DiagnosisType	ID

The final data model of the example is described by Tables 9, 10 and 11 in the generic model description. The results are also shown in the classical entity-relationship representation by two entity-relationship diagrams (Fig. 3. and Fig. 4.) Both diagrams show analytical or dimensional data models.

The model of Fig. 3 represents the analytical dimensional model (or star model; or cube) of accommodation of patient's admission on hospital wards.

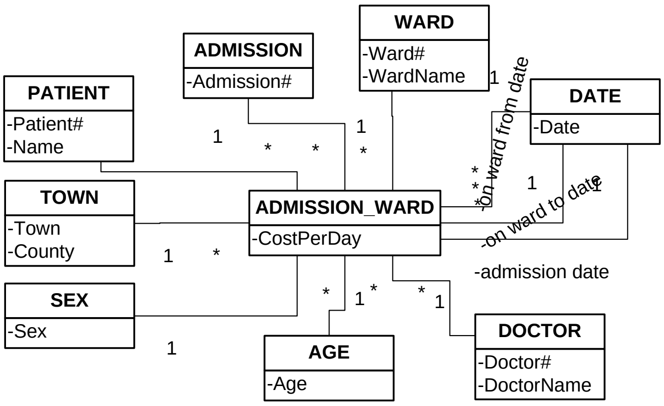

Fig. 3 Analytical dimensional model: Accommodation of patient's admission on wards

The model in Fig. 4 represents the analytical dimensional model of diagnosing patients in hospital.

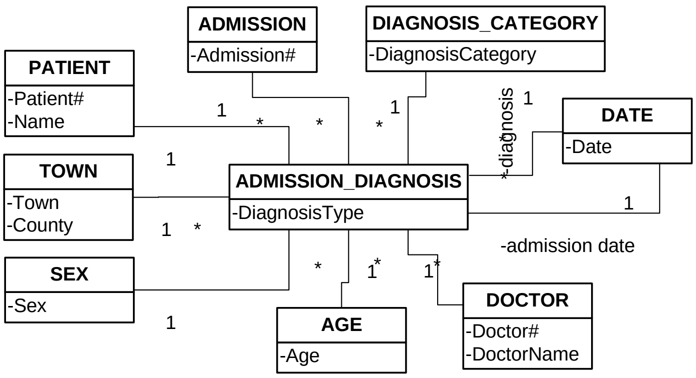

Fig. 4: Analytical dimensional model: Diagnosing of patient

4. Conclusion

The paper presents the outline of the algorithm for generating an analytical (dimensional) data model from the input data model. The input data model may represent collected business data requirements and may be of any kind: operational, analytical or even unknown. It is described in generic form and therefore easily changeable.

The presented algorithm may be used for conversion of any data model into analytical (dimensional) data model.

5. References

[1] Barker, R. CASE*Method; Entity Relationship Modelling, Addison-Wesley Publishing Company, Wokingham, 1990.

[2] Martin, J., Odell, J. Object-oriented Methods; A Foundation. Prentice-Hall, Englewood Cliffs, 1995.

[3] West, M. Developing High Quality Data Models, EPISTLE http://www.matthew- west.org. uk/documents/princ03.pdf [2007-01-15]

[4] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language; User Guide. Addison-Wesley Professional, 1998.

[5] Chen, P. The Entity-Relationship Model; Toward a Unified View of Data. ACM Trans. on Database Syst. 1,1, 9-36, 1976.

[6] Golfarelli, M., Maio, D., Rizzi, S. Conceptual Design of Data Warehouses from E/R Schemes. Proc. of the Hawaii Int. Conf. on System Science, 1998.

[7] Varga, M. A Procedure of Conversion of Relational into Multidimensional Database Schema. Journal for Computing and Information Technology (CIT), Vol. 10, No. 2, 69-84, 2002.