Mile Pavlić
Pedagogical Faculty Rijeka - Informatics Dept.

UDK: 681.327.2
Original Scientific Paper

Problems in the Application of Generalized Abstractions

According to the methodology for planning an information system, the basic concept used in describing complex systems is the principle of abstraction. This paper analyzes a generalized abstraction which is often a natural description of the system. A problematic variant of a generalized tree is defined in which the following are valid: a great number of subtypes exist, the number of subtypes is unknown as well as the attributes of known and unknown subtypes. The paper sets out the problems and gives one solution by substituting any generalized abstraction with the model made with the help of a series of classificatory and aggregative abstractions.

The paper shows that one coordinate of the modeling space which is assumed to be basic and orthogonal in the modeling space can be carried out through complex relations between the other two referential coordinates, in other words, classification and aggregation are two extreme variants (coordinates) of modeling the system and generalization is a defined function in the given coordinates.

Besides making a theoretical contribution, the paper also contributes in a practical manner, because in the implementation of problematic variants of generalization it gives an overall solution to the model without loss of information.

Key words: generalization, problematic generalization, methodology, abstraction, aggregation, classification, E-R model, relational model.

1. Introduction

According to the methodology of planning information systems, one of the standard methods of data modeling is the Entity-Relationship (ER) method which belongs to the semantically-rich group of methods. It is installed in almost all CASE tools. According to it, the principle of abstraction is used in order to master the description of the complexities of the system. Three types of abstraction exist: classificatory, generalized and aggregative.

Information systems today have mainly been developed through fourth generation software based on the relational method. This method does not have a supportive concept of generalization. Therefore, the problem is to develop IS which requires implementation of generalization. Analysts faced with generalization do not have any elaborate mechanism for translating semantically-rich generalized concepts of ER language into semantically-poor language of relational and other data models.

This paper offers a solution to the problem, translating an overall generalization into other abstractions without loss of information.

2. Problems with generalization

There are some entities in a real system which we monitor and describe with a generalized tree. In Figure 1. an example is given of a generalized entity type ship. Generalization is such a description where some individual appearance of entities which have the same attributes, participate in the same relations and on which the same operations can be applied, are monitored as an entity class. Such entity which individually appears is named subtype (or subclasses) in relation to the entity type and the entity type itself its supertype (or superclasses). Attributes, relations and operations of the supertype can be applied to the subtype but not vice versa.

We define a general starting entity type (T) with a set of attributes (A) and n appearing (synonym: an Instance) as: T(A,n).

We define generalization (G) entity type T in a set of m subtypes in: Tg, T1, T2,...Tm such as:

$$G(T(A,n)) = G(Tg,T1, T2,...Tn) =$$
$$Tg(B,n) \text{ "S" } U(T1(C1,n1),T2(C2,n2),...Tm(Cm,nm)) \qquad (1)$$

or abbreviated G(T). A graphic presentation of generalization is given in Figure 2.

We say that generalization G over entity type T is a generalized operation "S" which specializes the initial entity type T with attributes A in a set of entity types: Tg, T1, T2,...Tm.

Tg has the same number of appearances as T but a smaller number of attributes for the attribute C set. Tg has a set of B attributes. T1 has a set of C1 attributes transferred from the initial entity type T, T2 has a set of C2 attributes transferred from T,... and Tm has a set of Cm attributes transferred from T. The union of attributes C1, C2...Cm is equivalent to the set of C attributes.

Equally A = B U C, in other words the union of sets of attributes B and C is equivalent to the set of attributes of the initial entity type T.

It is also stated that the number n appearance of a new entity type Tg is equivalent to the number n appearance of the initial entity type T. Tg is called the supertype. It is also true that the number of appearances of types of objects, T1, T2...Tm is equivalent to the random numbers n1, n2,...nm whereby ni<=n is valid. T1, T2,...Tm are called subtypes of entity type Tg.

An operation of generalizations (marked with "S") on types of objects Tg, T1, T2,...Tm leads to the generalized tree G, in other words a chosen real system G(T(A,n)) is equivalent to the system model given with equation (1)

The complex operation S can be defined as two functions S1 and S2.

S1 is a function which combines the appearances of entity type Tg with the appearances of entity types T1, T2,...Tn.

S1(Tg) = union (T1,T2,...Tn)

S2 is an inverse function which combines the appearances of entity types T1, T2,...Tn with the appearances of entity type Tg.

S2(Ti) = subset (Tg)

Through an analysis of functions S1 and S2 we can determine the number of individual appearances of entities from co-domain for one entity number of domain. We extract the extreme values and write them in the form of an arranged pair (MIN, MAX) which we call the cardinalities of generalization.

The cardinalities of generalization of subtypes T1, T2,...Tm towards supertype Tg are always minimal (1:1), in other words at least one and at the most one individual appearance of the supertype belongs to one individual appearance of subtype.

The cardinalities of generalization from supertype to subtype are:

version 1. - (0,1) i.e. to one individual appearance of supertype belongs either none or at most one individual appearance of subtype (exclusive generalization) or

version 2. - (0,M) i.e. to one individual appearance of supertype belongs either none or at most many individual appearances of subtype (multiple generalization).

It is possible that the lower limit of cardinalities of generalization from supertype to subtypes is not 0 but 1. If the upper limit is equal to 1 and the lower limit is equal to 0, then the union of subtypes is the subset of appearances in the supertype. The lower limit one means that each entity of the supertype is connected at least to one entity of the subtype.

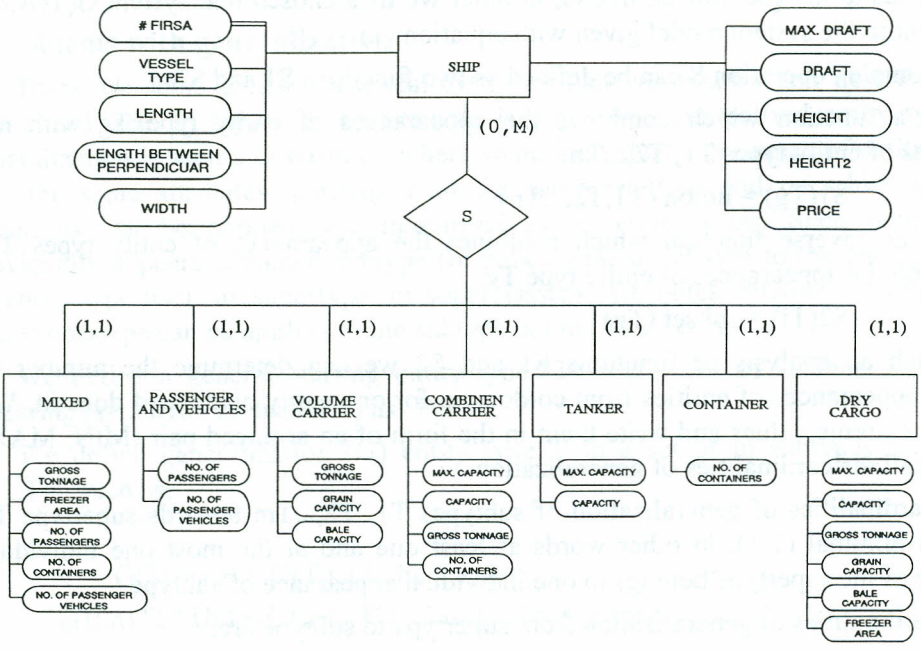

Figure 1. Generalization tree of entity type SHIP

Subtypes and supertype of the generalization tree can be connected to other types of entities. In this work, we will demonstrate the relationship of one subtype T3 to the entity type O1 through the connection type V1 (see Figure 2).

Generalization in practical use is undesirable because of the problem associated with it and planners of information systems avoid it. The development of a software aided by the systems of third or fourth generation (INGRESS, SUPRA and others) which are based on models: hierarchical, network, relational or postrelational type, do not provide a back-up for the generalization concept.

Besides this problem, other problems exist for a generalization tree with numerous subtypes:

Problem 1. A great number of subtypes

Problem 2. An unknown number of subtypes

Problem 3. Unknown attributes of known and unknown subtypes.

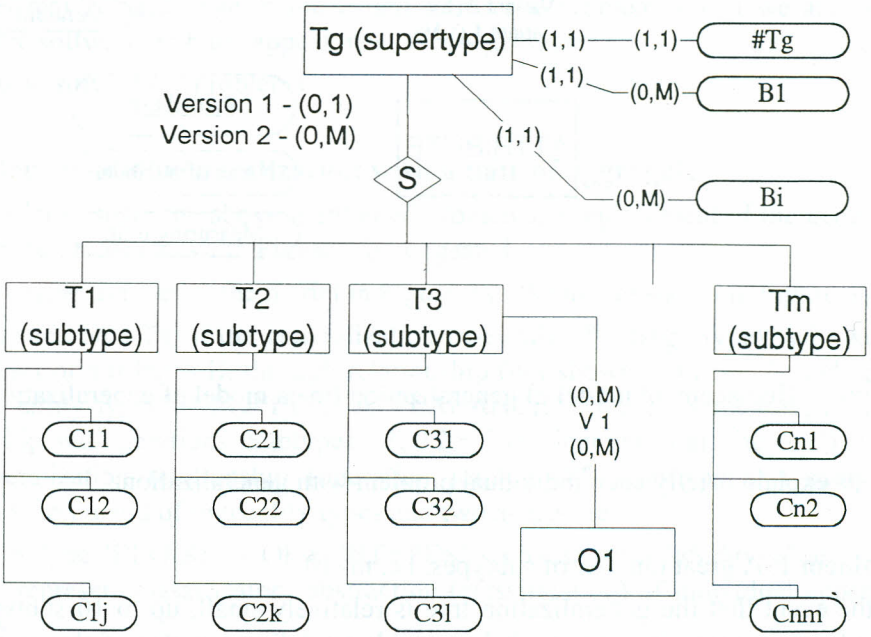

Figure 2. General generalization tree

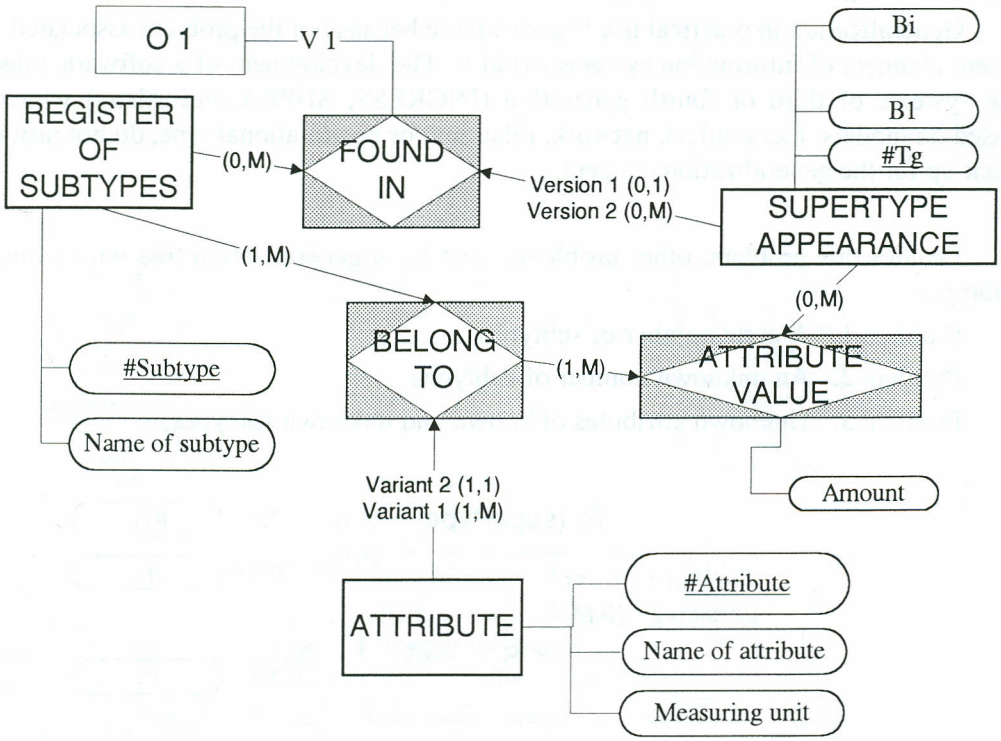

Figure 3. Replacement model of generalization (meta model of generalization)

Let us explain briefly each individual problem with generalization.

Problem 1. A great number of subtypes, i.e. m>10.

In the event that the generalization tree is relatively small, up to 10 subtypes with number of appearances in subtype, it is possible to define transfer of the same into a number of relations. According to this, it must be transferred, more or less successfully, into concepts which the chosen system of 4GL (Alagic, 1984) hold.

An extreme case of generalization would be a large number of subtypes, e.g. over 1,000 or more with the minimal number of appearances for each subtype. This would not be justified to transfer and realize according to the previous rules. Then it is not correct to transfer such model into a number of almost empty relations. The problem is then what to do.

Problem 2. An unknown number of subtypes, i.e. m is the element from domain [2,k] where k is a natural number greater than 2.

The model can be dynamic in such a way that subtypes over a period of time can be introduced. This requires a continuous operation in the field of knowledge with IS structure. Such IS operations continue indefinitely. The aim of every good program is its maximum independence from changes taking place in the real system. A problem arises when a new subtype appears and we are obliged to change the software.

Problem 3. Unknown attributes of known and unknown subtypes, i.e. elements of sets C1, C2,...Cm are changeable.

The attributes of subtypes (particularly new) can be unknown. Introduction of attributes requires a change in the IS structure. The problem is that we are obliged to change the software with the appearance of new attributes.

How to solve these problems?

3. Replacement of generalization with a pair of aggregates

In order to solve the above-mentioned problems, a replacement of the generalization tree with the model shown in Figure 3 is suggested.

We define transfer of the model in Figure 2 to the model shown in Figure 3.

The supertype Tg of the generalization tree called "Tg(supertype)" with key called #Tg and set of attributes B, through relationship (not shown in Figure 2) and operations becomes entity type "SUPERTYPE APPEARANCE" with the same key, attributes B, relationship and operations. Subtypes T1, T2,....Tm, their set of attributes C1, C2,....Cm and generalization relationship (operation joining "S" supertype and subtypes) are replaced with the aid of two entity types and two aggregates.

Entity type "REGISTER OF SUBTYPES" with attributes "#Subtype" and "Name of subtype" represent classificatory abstraction (classification) of individual appearance of subtypes.

Entity type "ATTRIBUTE" with attributes: "#Attribute", "Name of attribute" and "Measuring unit" represent classificatory abstraction of individual appearance of attributes of all subtypes.

The relationship type (aggregate) "BELONG TO" between these two entity types, represents the merging of attribute to subtype. At least one attribute must belong to one subtype i.e. sets C1, C2,...Cm should not be empty (otherwise its existence is not justified and the integrity of the entity is not satisfactorily met) but many attributes can belong to it (cardinality of relationship with "REGISTER OF SUBTYPES" is 1:M).

One attribute must belong at least to one subtype (otherwise it is senseless and impossible) but may belong at the most :

- variant 1. - to many subtypes (cardinality of relationship for "ATTRIBUTE" is 1:M) or
- variant 2. - to one subtype (cardinality of relationship for "ATTRIBUTE" is 1:1).

We can ask why two variants of generalization transfer exist (it is not similar to the previously defined versions). The reason for this is that it is not known if in the generalization tree semantically the same attribute belongs to two or more subtypes. Only that attribute which belongs to all subtypes becomes a supertype attribute. It is to be expected that there are some, as attributes exist which are common to all subtypes and are shown in the supertype. Thus, the poorer variant 2 is not general and a common solution is suggested so that the cardinality of aggregates of BELONG TO are (1,M):(1,M). Such model accepts both variants, and it is not necessary to prove.

From the structural concept viewpoint, the generalization tree contains also a concept of generalization relationship "S" i.e. S1 and S2, the copying functions of individual appearance of the supertype Tg into subtypes T1, T2,..Tm and vice versa. It relationship individual appearance subtype and supertype, i.e. adds to the of the individual appearances of individual supertype as well as attributes of supertype B and attributes of subtypes C1, C2,...Cm.

Relationship type "S" is built into the model (see Figure 3.) as relationship type "ATTRIBUTE VALUE" between the aggregate "BELONG TO" and entity type "SUPERTYPE APPEARANCE". This relationship type (aggregate) "ATTRIBUTE VALUE" has itself the attribute "Amount". Generalization "S" has two versions of cardinality from supertype to subtype. Aggregates of "ATTRIBUTE VALUE" with cardinalities (0,M) from "SUPERTYPE APPEARANCE" to "BELONG TO", i.e. one individual appearance of supertype has none or has many values of various attributes of various subtypes answers to version 2. Version 1. is a subvariant of version 2. and it is linked to it, but instead of (0,M) on the present defined model, limitations of (0,1) cannot be explicitly written in as they do not represent version 1. That would mean that one appearance of a supertype can have at most one attribute in all subtypes together. If we introduce relationship type "FOUND IN" between type of object "Supertype

APPEARANCE" and "REGISTER OF SUBTYPES" then the cardinalities versions from Figure 2 are built in as versions in cardinality of "APPEARANCE OF Supertype to "REGISTER OF SUBTYPES". Cardinalities of inverse copying from subtype to supertype are always (1,1) as in Figure 2 and have no version.

Emphasizing transfer of generalization concept in the model in Figure 3.

GENERALIZATION	HAS BECOME
entity type Tg (supertype, superclass)	entity type APPEARANCE OF SUPERTYPE
subtypes T1,T2,...Tm	individual appearances of entity type REGISTER OF SUBTYPES
attributes of all subtypes C1, C2,...Cm	individual appearances of entity type ATTRIBUTE
relationship of attribute to subtype	aggregation "BELONG TO"
generalization relationship of "S"	aggregation "ATTRIBUTE VALUE"
cardinality "S"	cardinalities type relationship "FOUND IN"
subtypes attribute values	values of attribute Amount

The similarity of the structural concept and limitation of two different models has thus been established. It remains to be shown that the relationship wherein the subtypes and the operations over the subtypes participate coincide with the relationship in model in Figure 3.

If subtype T3 (see Figure 2) participates in relationship type Vl with entity type 01, then that signifies that the appearances of subtype participate in V1. For this linkage, the fact that the subtype has certain attributes and that this individual appearance has the value of the same attributes is unimportant. The same applies for O1. Only the key attributes are important. Thus, in relationship type V1, knowledge about key individual appearances of T3 and O1 participates. The solution in Figure 3 shows that each entity type O1 and the appropriate V1 relationship still exists in the substitute model. The relationship V1 is fixed in that place in the model where individual appearances of the subtype are. This place in the model is an aggregation "FOUND IN". The cardinalities relationship type V1 are important for transfer of V1 to the relational model. The following combinations are possible: 1. (0,M):(0,M), 2. (0,M):(0,1) and 3. (0,1):(0,M) observing from O1. In the case where (0,M):(0,M) V1 is an aggregation. In key V1, beside key O1 two levels of abstraction (entity type key and entity type name) are built. Both levels of abstraction are present in relationship type (aggregation)

"FOUND IN" and that aggregation we link to V1. If this exceptional modeling, key relationship creation, is eliminated and we define that in V1 (in the case of M in the upper limit) only the key of individual appearances can enter as a key, then it can be shown that V1 is not linked to subtype T3 but to supertype Tg. In that case, in Figure 3 we link V1 with "APPEARANCE OF SUPERTYPE" but not with "FOUND IN". In both cases cardinalities V1 are identical in Figures 2 and 3.

Let us consider the case of cardinality type V1 (0,M):(0,1). That means that to O1 belongs at least none and at the most many entity appearances and to one appearance of entity subtype belongs at most one appearance from O1. While transferring V1 and O1 to the relational model, it is sufficient to introduce the external key relation O1 in the relation "FOUND IN". In the case of cardinality V1 type (0,1):(0,M) it is sufficient to introduce the external key "#Tg,#Subtype" in the relation O1 originating from entity type O1.

If any subtype T1, T2, ...Tm is linked with random entity types, then these relationships are connected to the aggregation "FOUND IN". It is thus shown that the model in Figure 3 has been a good substitute for the generalization tree even in the case of its subtypes being linked to various entity types.

There remains to be shown how the operations above the generalization tree have been transferred to the appropriate operation above the model in Figure 3. What is with the operations of subtypes T1, T2,...Tm and supertype Tg. Let us observe the basic operations of adding, erasing and changing the supertypes and subtypes. Introduction in the generalization tree is carried out through the supertype Tg. First of all, the key to appearance in supertype Tg and all its attributes B are defined, then subtype Ti is chosen (for i it goes from 1 to m) and its attributes Ci are entered (the key to appearance for Tg and all Ti is the same). If the cardinality relationship is (0,M) - version 2, then after insertion in some Ti, the following subtype Tj is chosen (for "j" it goes from 1 to m) and attributes similar to Ti are inserted and it continues in such a manner until all attributes of appearances classified in several subtypes from sets of T1 to Tm have been defined. In the subtype Ti we cannot add appearance if it has not been added in the supertype Tg.

According to the model in Figure 3 appearance with its key and attributes can be added in "SUPERTYPE APPEARANCE". After that it is inserted in the aggregation "FOUND IN" if that appearance has attributes in any subtype. A subtype can have one or more attributes, which is written in relationship type "BELONG TO" and with each appearance of that relationship type, the appearance of relationship type "ATTRIBUTE VALUES" is created. It is possible, according to the model in Figure 3, to add new subtypes and their attributes but the same is not possible in Figure 2. We can understand this as additional possibilities and suitabilities from the viewpoint of insertion (but not

erroneous erase). If we proclaim the concept "REGISTER OF SUBTYPES", "BELONG TO" and "ATTRIBUTES definitive for the user (insertion is not possible from the application and their content is determined by the designer) then the operation of insertion of appearance is equal in two different models.

The logic of the operation of transfer is similar to the insertion operation except that existing attribute values are suggested and it is up to the user to choose whether to keep them or change them.

The erase appearance operation in supertype Tg according to Figure 2 causes erase of appearance in all subtypes T1, T2...Tm (subtypes act towards supertype as weak entity types). According to Figure 3, due to referential integrity in aggregation "FOUND IN" and "ATTRIBUTE VALUES" appearance is erased as soon as we erase appearance in "REGISTER OF SUPERTYPES". The same applies for V1 in both models.

By this it is thus shown that the basic operations above the generalization tree have corresponding operations in the model in Figure 3.

We can conclude that the model in Figure 3 is a suitable replacement of the model from Figure 2 from the viewpoint of structure concept, link concept with the environment and concept operation. That model has no conceptual deficiencies. All information contained in the generalization tree exists and access to it is simple. Transfer of that model in a relational scheme database is given in Figure 4.

The former mentioned problems have disappeared. Problem 1. A great number of subtypes is unessential as it represents a problem of a great number of appearances of entity type "REGISTER OF SUBTYPES. Problem 2. An unknown number of subtypes is unessential as it represents performance of operation (adding, changing, erasing) for entity type REGISTER OF SUBTYPES. Problem 3. Unknown attributes of known and unknown subtypes have disappeared and have been reduced to the operation on entity type ATTRIBUTE.

REGISTER OF SUBTYPES (#Subtype, Name of subtype)
SUPERTYPE APPEARANCE (#Tg, B1, B2,...Bm)
ATTRIBUTE (#Attribute, Name of attribute, Measurement unit)
FOUND IN (#Tg, #Subtype)
BELONG TO (#Subtype, #Attribute)
ATTRIBUTE VALUE (#Subtype, #Attribute, #Tg, Values)

Figure 4. Relational scheme of data base

The basic idea of this model is to show the subtypes attributes as entity type. The problems generated from that are solved by adding new concepts. In order to achieve knowledge contained in generalization, several different methods for supplementing knowledge exist. The model suggested claims to be minimal.

4. Conclusion

By this, we have demonstrated that a generalization tree can be carried out from a classificatory and aggregative abstraction, in other words, it can be shown without loss of knowledge. The term generalization signifies association of partial similarities while classificatory and aggregative are extremes with complete similarities or without similarities.

Although the models are equivalent, they show themselves to be different while being transferred to the data base scheme. The user, according to model 3, has the possibility of adding attributes and subtypes. The reason for this is that that model is a general one and we can understand it as a generalization metamodel. A generalization metamodel permits expansion of the generalization scheme.

The disadvantage of the solution offered is that the user himself can erase subtypes and attributes and destroy the integrity of the data. This deficiency can be avoided if operation erase is forbidden.

This paper has not included an analysis of subtypes of subtype i.e. multiple generalization.

The generalization metamodel suggested could be used by planners of CASE tools and 4GL for development of data base on Entity-Relationship methods.

The paper demonstrates that one coordinate of modeling space has not been established but carried out. For a generalization abstraction it is held that it is basic and orthogonal in the modeling space, the same as classification and aggregation. It has been shown that it can be carried out with complex referential ratio from the other two abstractions. Classification and aggregation are two extreme variants (coordinate) of a modeling system and generalization is a defined function in the given coordinates.

During research, the author was faced with several questions such as:

Does any other basic abstraction exist which cannot be carried out from basic abstraction?

Do any more abstractions which have been carried out exist?

Could aggregation be observed as a metamodel and what kind?

What use could be made of it?

How would such a model look?

How to prove that two such different models are the same?

Can a general method be found which will copy the initial to a final model without loss of information?

References

[1] (Alagić 1984) S. Alagić , "Relacijske baze podataka", Svjetlost, Sarajevo,1984.

[2] (Brodie 1986) Michael L. Brodie (1986), "On the Development of Data Models", Springer-Verlag, NJ, USA, 1986, (Brodie 1986).

[3] (Borgida 1986) A. Borgida, ... (1986), "Generalization / Specialization as a Basis for Software Specification", Springer-Verlag, NJ, USA, 1986, (Brodie 1986).

[4] (Chen 1976) Chen, P.P. (1976), The entity-relationship model - Toward a unified view of data, ACM Transactions of Database Systems. Vol.1., No.1.

[5] (Chaing and Bargeron 1980) T.C. Chaing and R.F. Bargeron, "A Database Management System With an E-R Conceptual Model", (Chen 1980).

[6] (Chen 1980) P.P. Chen (Editor), "Entity-Relationship Approach to Systems Analysis and Design", North-Holland, Amsterdam, the Netherlands, 1980.

[7] (Chen 1983) P.P. Chen (Editor), "Entity-Relationship Approach to Software Engineering, Anaheim, North-Holland, Amsterdam, the Netherlands, 1983.

[8] (Codd 1970) E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, Vol.13, No.6,. 377-387, June 1970.

[9] (Codd 1979) E.F. Codd, "Extending the Database Relational Model to Capture More Meaning", ACM Transaction on Database Systems, Vol 4, No. 4, 397-434, December 1979.

[10] (Date 1986) C.J. Date, "An Introduction to Database Systems" (volume I fourth edition), Addision-Wesley Publishing Company, Inc., 1986.

[11] (Davis 1983) C.G. Davis, S. Jajodia, P. A-B. Ng. and R.T. Yeh (Editors), "Entity-Relationship Approach to Software Engineering", North-Holland, Amsterdam, the Netherlands, 1983.

[12] (Giorgio 1987) B. Giorgio and E. Antonio, "Extending the Entity - Relationship Approach for Dynamic Modeling Purposes", (Spaccapietra 1987).

[13] (Hawryszkiewycz 1988) I. T. Hawryszkiewycz, "Introduction to Systems Analysis and Design", Prentice - Hall, New York, 1988.

[14] (Hice 1978) G.F.Hice, W.S.Turner, L.F.Cachwell, "System Development Methodology", North-Holland, Amsterdam, 1978.

[15] (Mylopoulos 1986) J. Mylopoulos, H. J. Levesque (1986), "An Overview of Knowledge Representation", Springer-Verlag, NJ, USA, 1986, (Brodie 1986).

[16] (Pirotte 1977) A. Pirotte, "The Entity-Association Model: An Information- Oriented Data Base Model", International Computing Symposium, North-Holland Publishing Company, Amsterdam,1977.

[17] (Spaccapietra 1987) S. Spaccapietra (Editor), "Entity - Relationship Approach : Ten Years of Experience in Information Modeling", North - Holland, Amsterdam, the Netherlands, 1987.

[18] (Strahonja 1992) Vjeran Strahonja i dr., "Projektiranje informacijskih sustava - metodološki priručnik", ZID i INAINFO, Zagreb, 1992.

[19] (Tsichritzis 1982) D.C. Tsichritzis , F.H. Lochovsky ,"Data Models", Prentice Hall, 1982.

Received: 1995-09-01

Pavlić M. Problemi u primjeni generalizacijske apstrakcije

Sažetak

Prema metodologiji projektiranja informacijskih sustava temeljni koncept za opisivanje složenih sustava je princip apstrakcije. Ovaj rad analizira generalizacijsku apstrakciju koja je često prirodan opis sustava. Definirana je jedna problematična varijanta generalizacijskog stabla u kome vrijedi: postoji velik broj podtipova, broj podtipova je nepoznat i nepoznati su atributi poznatih i nepoznatih podtipova. Rad opisuje probleme i daje jedno rješenje zamjenom bilo koje generalizacijske apstrakcije modelom sačinjenim pomoću niza klasifikacijskih i agregacijskih apstrakcija.

Rad pokazuje da se jedna koordinata prostora modeliranja, za koju se drži da je bazna i ortogonalna u prostoru modeliranja, da izvesti složenim relacijskim odnosom iz ostale dvije, odnosno klasifikacija i agregacija su dvije krajnje varijante (koordinate) modeliranija sustava a generalizacija je definirana funkcija u zadanim koordinatama.