A methodology for creating a Conceptual Model under Uncertainty

Bogdan Walek, Jiri Bartos, Cyril Klimes

***Abstract*—**This article deals with the conceptual modeling under uncertainty. First, the division of information systems with their definition will be described, focusing on those where the construction of a conceptual model is suitable for the design of future information system database. Furthermore, the disadvantages of the traditional approach in creating a conceptual model and database design will be analyzed. A comprehensive methodology for the creation of a conceptual model based on analysis of client requirements and the selection of a suitable domain model is proposed here. This article presents the expert system used for the construction of a conceptual model and is a suitable tool for database designers to create a conceptual model.

***Keywords*—**conceptual model, conceptual modeling, database, methodology, uncertainty, information system, entity, attribute, relationship, conceptual domain model, fuzzy

I. Introduction

THE most important activity that is taking place in the early stages of creating any information system is the correct understanding of the requirements for information system. Appropriate analysis of the requirements for this system is fundamental as well. For a proper understanding of these requirements a form of visualization is the best approach. In some information systems analysis of requirements is consistent with the analysis of requirements for data structure and data storage in this system [3]. As a consequence, in these cases an analysis of requirements is in fact an analysis conducted for creation the database itself and its logical structure. In this case the appropriate tool for analysis of requirements and their visualization is conceptual model that determines the contents of future database.

For completion of context, it is stated that in the case of client-server architecture of the information system, three basic layers are distinguished:

1. presentation layer
2. application layer
3. data layer

Bogdan Walek is with the Department of Informatics and Computers, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic (e-mail:bogdan.walek@osu.cz).

Jiri Bartos is with the Department of Informatics and Computers, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic (e-mail:jiri.bartos@osu.cz).

Cyril Klimes is with the Department of Informatics and Computers, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic (e-mail:cyril.klimes@osu.cz).

Based on the complexity of the information system and the way how various layers of above stated architecture are used and handled, the information systems can be divided into following types:

1. *Complex IS* – presentation, application and data layers are logically separated and the information system is used for effective and comprehensive support of all processes of the organization.

2. *Simple IS* – application and presentation layer can penetrate each other; information system supports and usually resolves only a few simple tasks (or processes).

3. *Database IS* – all three layers are interconnected and are parts of a database; information system performs simple tasks and supports simple processes.

Moreover, in the case of the first type of the information system, the emphasis is on the correct design of application logic and implementation of crucial processes and functions. On the basis of these activities, the data model that is the foundation for future database can be relatively easily created.

In the case of the second or third type of information system, which implements generally simple processes, more emphasis is put on the design of the data layer of the system. After the design of the data layer is completed, the focus moves on creation of application and presentation layer altogether because these two layers are often very closely linked to the data layer or are even created directly in the proposed database system. Therefore, during the analysis of requirements it is appropriate to establish a conceptual model that can afterwards be transformed into a logical model and from it the physical model can be easily generated.

This distribution implies that the creation of a conceptual model as a basis for analysis of requirements is suitable for the second and third type of information systems. It may therefore seem that for the first type of information system the conceptual model is not an appropriate visualization tool. Olivé however states that for any development of an information system it is necessary to define its conceptual scheme [8]. Conceptual diagram of an information system determines its domain and functionality that the system covers. Each conceptual scheme can then be described using logical language, such as UML, which can be used for modeling the basic Use–case of the system or even its analytical model. Since the conceptual model is a part of the conceptual scheme, it is possible to use it for visualization of basic entities, attributes and relationships that can occur within the targeted system. This conceptual model can afterwards be transformed into an analytical or object model if the system.

The article focuses primarily on the information systems whose analysis of requirements is essentially an analysis conducted for the creation of database and its logical structure.

II. Problem Formulation

The traditional approach to the creation of the database consists of the following steps:

- Analysis of requirements
- Creation of a logical model of the database
- Transformation into physical database model

In this approach the main role and responsibilities for creation of logical and physical model of the resulting database on the basis of user requirements lies on the database designer. This approach assumes a lot of expert knowledge on the side of the designer and the final form of the resulting database depends largely on this knowledge, which can be disadvantageous. Another disadvantage is the fact that the designer often does not have the expert knowledge in the field of given problem domain, which can lead to an incorrect database design.

It should also be emphasized that the conceptual model will be used further in the text in the context of database and its implementation, not in the context of the entire information system. The terms conceptual model and analysis of requirements will be used for database design and its transformation into a logical model and afterwards the physical model that is implemented in a specific relational database system.

Currently, one of the main tasks during the analysis of requirements for the proposed database is to understand the requirements of the customer properly. In case of misunderstanding, the risk of a bad database design is imminent and can cause problems in data handling and other operations. It also happens that seemingly correct understanding of customer requirements is implemented by the designer in a different way because the requirements were defined rather vaguely and indefinitely, which allowed several methods of their interpretation. The customer, however, often fails to specify requirements in a way to make them perfectly understandable for designers, which is necessary for error-free database design. Therefore it is important to interpret and process the vague requirements properly.

Another problem which can arise in the process of the database design is to identify the problem domain appropriately, especially to understand and use concepts, vocabulary and dependencies that are valid for the domain correctly. An analyst and a database designer are not often experts in the domain in which they design the proposed database and information system, therefore they need to learn how to orientate in the problem domain properly. One way how to conduct this learning process is to invite an expert for the domain who is well suited in it and together with the client they are able to pass important information applicable to a given problem domain. Another way is the creation of a conceptual domain which includes entities and relationships relevant to that domain. Or even already existing suitable domain conceptual model that describes the problem domain can be used. How to create a conceptual domain model and select one model from the already prepared set of domain models will be described below.

Next, there is a need to design the resulting conceptual model based on input information and correct the problem domain. This activity is traditionally done by database designer, whose abilities and knowledge influence the quality of the resulting model. The article proposes the creation of an expert system that will handle the inputs and according to the selected domain model will use the correct knowledge base resulting in a conceptual model containing the most appropriate entities, attributes, and relationships among those entities. The expert system then may partially or completely replace a database designer or may become a tool for database designers, who will be able to automate some of their tasks.

III. Problem Solution

As was already stated in the text above, it is appropriate to propose a methodology solution for the creation or selection of a correct conceptual model based on analysis of requirements. The solution that consists of partial steps was outlined in the previous chapter. The goal of the article is to propose a comprehensive methodology that will describe the conceptual modeling process starting with the analysis of requirements and resulting into construction and visualization of the final conceptual model.

Before the methodology definition itself, this chapter will cover the basic terms which will be further used in the text.

A conceptual model is a general model of a system that does not describe its implementation or its technological specifications, but defines the content of the future IS [1], [2]. In this article, the conceptual model is used to model the database of information system.

Conceptual model works with these basic concepts [1], [2], [7]:

1. *Entity* – is an abstraction from the complexities of a domain. It represents an aspect or an object of the real world. Examples of entities: student, subject, teacher.

2. *Relationship* – identifies how two or more entities are related to one another. Relationships can be represented by verbs, linking two or more nouns (entities). Examples of relationships: teaches (between entities teacher and subject), studies (between entities student and subject).

3. *Attribute* – represents the properties of entities. Examples of attributes: name, surname, age, email, phone (attributes of entity teacher).

Conceptual domain model describes the problem domain using the concepts from conceptual model (entities, attributes and relationships) [1].

Creation of methodology started from these basic steps, which will be described in detail in next chapters:

1. Processing of the requirements along with the uncertain information
2. Identification of the problem domain and the selection of appropriate conceptual domain model or design of a new domain model

3. Creation and impletion of the knowledge base for purposes of the expert system
4. Selection of the most appropriate entities and relationships
5. Construction of the resulting conceptual model

The steps of the methodology are visualized in the following figure:

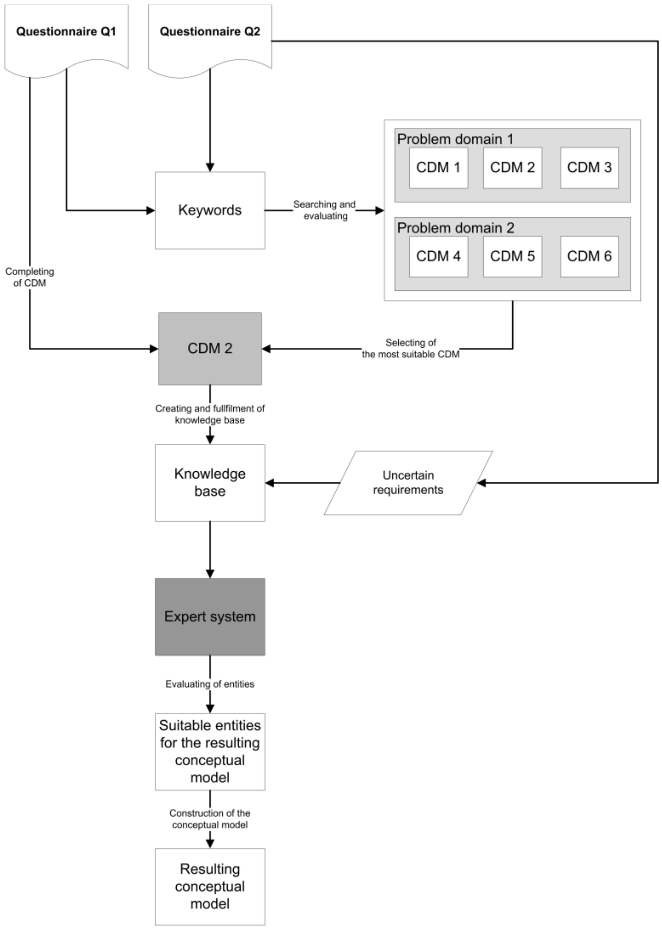

Fig. 1 Methodology for creating a conceptual model

A. Processing of the requirements along with the uncertain information

At the beginning the requirements of the customer need to be analyzed properly to create a correct conceptual model of the future database. These requirements are often specified in a fairly vague and unclear manner, there is an essential need to know the answers to the following questions:

a) What will the structure of the database be like and what data will be stored in it?
b) What is needed to be stored for the individual entities (e.g. if the entity will include a customer, what data will be stored? Only their basic contact information or even the data about residence of a customer)?
c) What are the relationships and interactions among different entities?

If communication between the customer and the database designer is going the right way, at the end of this step the designer will have a complete picture of the future database.

The step of processing the requirements for the database is conducted before identifying the problem domain in which is being operated and before the identification of conceptual domain model that will be used. The reason for this decision is that from some requirements it is possible to obtain the necessary information to determine the problem domain and select the most appropriate conceptual domain model. To obtain information about the problem domain, a list of keywords that characterize the requirements of the customer is created. Then the tool will gradually go through the conceptual domain models and search for the entities and attributes that correspond to those keywords. The result is an evaluated list of conceptual domain models that have been identified as suitable for further work. Problem domain is then automatically assigned, based on the conceptual domain model that will be selected in the next step.

Formally the algorithm that evaluates conceptual domain models on the grounds of keywords can be described as follows:

CDM - conceptual domain model
CDM_base = the set of all conceptual domain models
IMP[entity] = degree of importance of entity
IMP[attribute] = degree of importance of attribute
Rate[CDM] = the evaluation of conceptual domain model (its rate)
keywords – the list of keywords
write_rate – the method for saving the evaluation of CDM

```
Forall CDM in CDM_base
{
  Rate[CDM] = 0;
  Forall entity in CDM do
  {
    Forall(keyword in keywords)
    {
      If(entity(name) = keyword)
      {
        Rate[CDM] = Rate[CDM] +
        IMP[entity];
      }
    }
    Forall att in entity do
    {
      Forall(keyword in keywords)
      {
        If(att = search_keywords(keyword))
        {
          Rate[CDM] = Rate[CDM] +
          IMP[attribute];
        }
      }
    }
    write_rate(Rate[CDM],Rates[CDM])
  }
}
```

Formally the definition of calculation of the rate of conceptual domain model looks as follows:

E – set of all entities that are corresponding to the keyword

A – set of all attributes that are corresponding to the keywords

$IMP(e)$ - degree of importance of entity e from set E, $IMP(e) \epsilon \{0,10\}$

$IMP(a)$ - degree of importance of attribute a from set A, $IMP(a) \epsilon \{0,5\}$

The conceptual domain evaluated is then calculated using the following formula:

$$Rate[CDM] = (e_1 * IMP(e_1)) + (a_1 * IMP(a_1)) + (e_2 * IMP(e_2)) + (a_2 * IMP(a_1)) + \ldots + (e_n * IMP(e_n)) + (a_n * IMP(a_n))$$

The resulting list of evaluation for every conceptual domain model is then simple set of numbers that represents the rate of conceptual domain model:

Rates[CDM1] = 89

Rates[CDM2] = 53

Rates[CDM3] = 26

The searching of keywords can be described simply using the following picture:

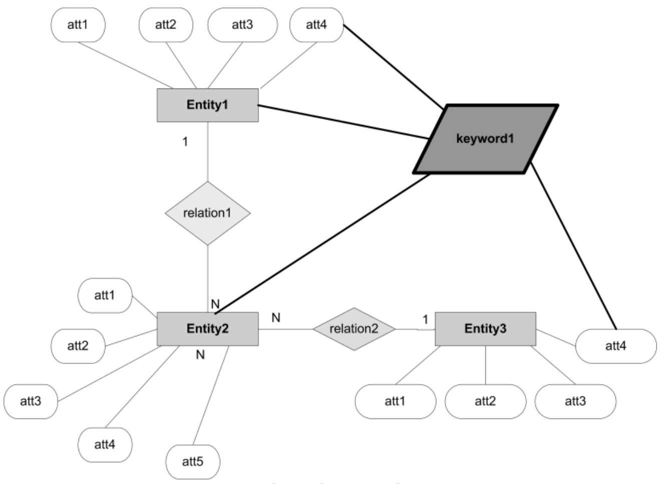

Fig. 2 Searching keywords in CDM

Various options of choosing the most suitable conceptual domain model will be described in the next step.

As mentioned, during the analysis of requirements a vague description of these requirements encounters will be very frequent. Therefore, two types of questionnaires which will contain all the requirements of the customer are created. Questionnaires are documents for representation and storage of requirements in a suitable form both for the customer and for the proposed tool that processes these questionnaires.

The first questionnaire Q1 will include identification of entities that are part of the generated conceptual model. Along with entities their attribute names are also clearly defined as well as links to other entities (if there is any relation to another entity) by this questionnaire.

The sample questionnaire Q1 for entities indentification can look like this:

TABLE I
SAMPLE QUESTIONNAIRE Q1

Entity name	Entity description	Attributes
Entity 1	Description 1	Attribute 1-1
		Attribute 1-2
		Attribute 1-3
		Attribute 1-4
		Attribute 1-5
Entity 2	Description 2	Attribute 2-1
		Attribute 2-2
		Attribute 2-3
Entity 3	Description 3	Attribute 3-1
		Attribute 3-2
		Attribute 3-3

This questionnaire can be also used for completion of the conceptual domain model that will be used for the construction of the final conceptual model. If the names of the entities in questionnaire and the entities in domain model are the same, their attributes will be processed and the correlation will be used for extension of the entities in domain model and the conceptual model.

Formally the algorithm can be described as follows:

```
Forall entity in questionnaire
{
  If(entity==search_entity_in_domain_mod
  el(entity))
  {
    Forall attributes in entity
    {
      If(attribute !=
      search_attributes_in_domain_entity(
      entity)
      {
    insert_att_to_domain_entity(entity);
      }
    }
  }
}
```

The second questionnaire consists of further requirements that do not lead to clear identification of all entities or attributes but are essential for the future database structure. This questionnaire contains the answers on questions that specify what exactly the customer wants to store in the database. On the basis of the questionnaire more entities, attributes and relationships will be later identified.

Questionnaire Q2 with additional requirements can look as following:

TABLE II
SAMPLE QUESTIONNAIRE Q2

Question	Answer
Question 1	Yes|Rather yes|Rather no|No
Question 2	Yes|Rather yes|Rather no|No

Question 3	Small|Medium|Big
Question 4	Few|Medium|Many

B. Identification of the problem domain and the selection of appropriate conceptual domain model or design of a new domain model

In the second step the most suitable conceptual domain model will be chosen.

Firstly we present the formal definition of conceptual domain model:

Conceptual domain model is a set, $D=(E,\ A, A_e, R, e_s, e_f, car(e_s, R_a), car(e_f, R_a))$, where:

E –finite set of entities

A – finite set of attributes

A_e – finite subset of the set A belonging to the element e of the set E

R – set of relationships among entities

e_s – element of the set E entering relationship R_a

e_f – element of the set E leaving relationship R_a

$car(e_s, R_a)$ – number of instances of the entity e_s that are part of relationship R_a

$car\left(e_f, R_a\right)$ number of instances of the entity e_f that are part of relationship R_a

As an example of conceptual domain model we can present this sample model with three entities:

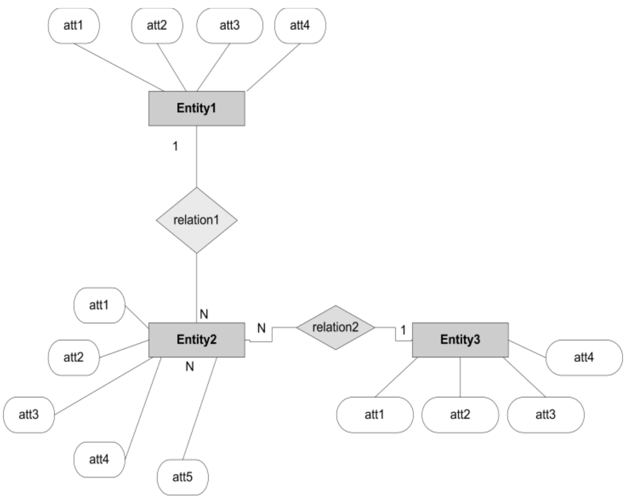

Fig. 3 Sample conceptual domain model

Database designer has three options for choosing the most suitable conceptual domain model:

1. Manual selection based on expert knowledge and subjective experiences of database designer. In this case the evaluated list of models (created in the first step of methodology) peforms assistant role in the final decision. Problem domain is selected along with conceptual domain model.

2. The selection of the “most suitable” conceptual domain model is made on the grounds of evaluated list of models that was created during the previous step. After the selection of conceptual domain model the problem domain is automatically identified – the main reason is that every domain model is a part of corresponding problem domain.

3. Alternatively new conceptual domain model can be created following a set of three rules:

- Identification of entities and attributes and their appropriate labeling
- Retrieval of the relationships between entities and their appropriate labeling
- Retrieval of cardinal numbers for individual relationships and correct assigning of these numbers to corresponding relationships

Example of creation of conceptual domain model is presented in the following picture:

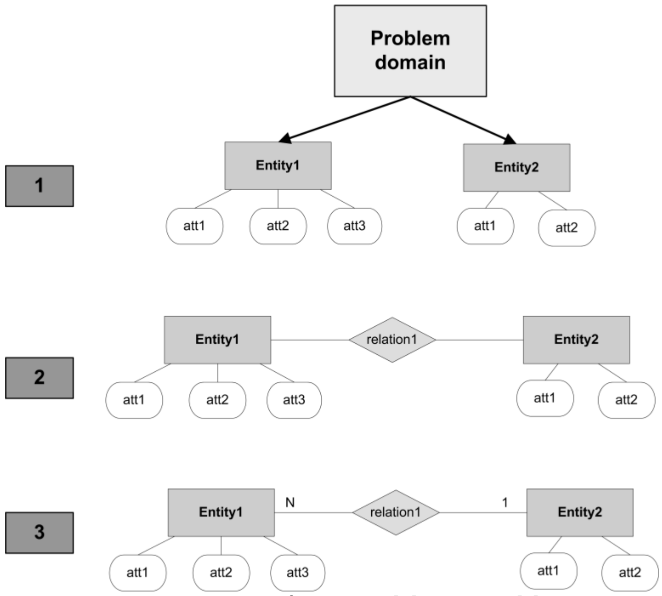

Fig. 4 Creation of conceptual domain model

C. Creation and impletion of the knowledge base for purposes of the expert system

Next step is the creation and impletion of the knowledge base of the expert system that will be used for the selection of the most suitable entities for resulting conceptual model. The knowledge base consists of set of IF-THEN rules, which contain the information about target domain and are based on vague input information that was gathered using second questionnaire and is defining the suitability of the entities for resulting model.

The knowledge base contains rules for the evaluation of the conceptual domain model entities, specific rules are applied to the input information that is obtained from the questionnaire altogether with uncertain requirements.

Formally the IF-THEN rules created on the grounds of vague requirements collected from questionnaire can be described as follows:

IF $X_1$is A_1 AND $X_2$is A_2 AND … AND $X_n$is A_n THEN Y is B

where:

$X = \{X_1, X_2, \dots, X_n\}$, where elements $X_1, X_2, \dots, X_n$ represent questions of the questionnaire Q2 and n is the number of questions of the questionnaire Q2

$A = \{A_1, A_2, \dots, A_n\}$, where elements $A_1, A_2, \dots, A_n$ represent answers of the questionnaire Q2 and n is the number of the questionnaire Q2 responses (answers)

Y – represents evaluated entity

B - represents linqual evaluation of the evaluated entity

The knowledge base for specific entity consists of set of IF-THEN rules that are defined as the Cartesian product:

$$Z = Q \; x \; A \; x \; S$$

where,

Q - finite set of questions of the questionnaire Q2

A - finite set of answers to the questionnaire Q2

S - finite set of IF-THEN rules

A part of knowledge base for creation of conceptual model for hotel information system can look as following:

```
IF (LOCATION is CITY) and (AREA is MEDIUM) and (FAMILIES are MANY) and (YOUNG PEOPLE are FEW) and (SENIORS are FEW) THEN (RATE OF AQUAPARK is BIG)
IF (LOCATION is CITY) and (AREA is MEDIUM) and (FAMILIES are MANY) and (YOUNG PEOPLE are FEW) and (SENIORS are FEW) THEN (RATE OF EXTREME SPORTS is VERY SMALL)
```

D.Selection of the most appropriate entities and relationships

All entities from the conceptual domain model are, based on uncertain information obtained from questionnaire and application of the appropriate IF-THEN rule set, evaluated using the expert system. Expert system evaluates the entities witch elements of the fuzzy set A and universe U and this evaluation can be describe as a function:

$$A: U \;\rightarrow [0,1]$$

where values $A(x) \in [0,1]$ represent the degree of membership of element $x \in U$ to fuzzy set A [4].

In this case, the values $A(x) \in [0,1]$ determine the suitability of the occurrence of the entities in the resulting conceptual model. The higher value $A(x)$, the more the occurrence of the entities in the final model is recommended.

Final decision about model suitability is in the hands of database designer who evaluates the rate of suitability for every individual entities and selects all the appropriate ones for creation of final conceptual model.

E. Construction of the resulting conceptual model

Resulting conceptual model is afterwards created on the basis of information collected in the first questionnaire and evaluated entities which are the outputs from expert system. These entities along with attributes and relationships (attributes and relationships both originate in conceptual domain model) form the final conceptual model. Entities, their attributes and their relationships can then be generated in the general form which can be stored and furthermore used for visualization. XML file is used for storing all of the information that describes the final conceptual model.

The proposed structure of the XML file is based on the following elements and attributes:

<entity> - element representing the entity

name - attribute representing the name of an entity

attributes - the number of attributes in the entity

<attribute_name> - element representing the entity attribute

<relationship> - element representing the relationship between two entities

<startentity> - name of the entity entering into relationship

<relation> - name of the relationship

<endentity> - name of the entity entering into relationship

<car_start> - cardinality of the entities entering into relationship

<car_end> - cardinality of the entities leaving from the relationship

An example of XML document representing the conceptual model of the university information system is shown below [5]:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<entity name="student" attributes="3">
  <name>John</name>
  <surname>Smith</surname>
  <age>18</age>
</entity>
<entity name="subject" attributes="2">
  <name>Biology</name>
  <content>anatomy</content>
</entity>
<relationship>
  <startentity>student</startentity>
  <relation>studies</relation>
  <endentity>subject</endentity>
  <car_start>1</car_start>
  <car_end>N</car_end>
</relationship>
```

For this example a simple way of representing entities, attributes, relationships and in conceptual model can be found. Conceptual model in this format can be visualized and relatively easily transformed into another XML format that can be imported into any of the tools that are being used for data modeling. One of these tools is Toad Data Modeler [6], which allows creating a logical model and then a physical model of the future database a conceptual model. Due to this fact the real application of the methodology and the expert system that proposes the resulting conceptual model can be ensured.

IV. CONCLUSION

This article analyzed different types of information systems and the suitability of usage of conceptual model for visualization of requirements for database of information system and for future creation of such database. Next, the disadvantages of traditional approach were defined. The general methodology for creation of conceptual model based on basic analysis of customer requirements and selection of appropriate conceptual domain model was created. Moreover, the method of representation of clearly defined and vaguely described inputs for the expert system was proposed. The expert system proposes suitable entities for resulting conceptual model. The resulting conceptual model uses XML form and therefore can be easily visualized and furthermore used by Toad Data Modeler tool.

ACKNOWLEDGMENT

Presented topic is also a part of the internal grant SGS10/PřF/2012, called Fuzzy modeling tools for analysis and design of information systems, at the Department of Infomatics and Computers, University of Ostrava.

REFERENCES

[1] V. Řepa, *Analýza a návrh informačních systémů.* EKOPRESS, Praha 1999, pp. 146-159.

[2] C. Batini, S. Ceri, S. Navathe, *Conceptual database design: an entity-relationship approach*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1992, ch. 1,2,3,4.

[3] S. Lauesen, *Software Requirements: Styles and Techniques.* Addison-Wesley Professional, Glasgow, 2002, ch. 1,2,3.

[4] C. Klimeš, “Model of adaptation under indeterminacy”, In Kybernetika, Vol.47 (2011) No.3, Prague 2011, pp. 355 – 368.

[5] B. Walek, C. Klimes, "Fuzzy tool for conceptual modeling under uncertainty" in Fourth International Conference on Machine Vision (ICMV 2011): Machine Vision, Image Processing, and Pattern Analysis, edited by Zhu Zeng, Yuting Li, Proceedings of SPIE Vol. 8349, SPIE Bellingham, 2012, pp. 83492Z-83492Z-5.

[6] Toad Data Modeler, http://modeling.inside.quest.com/index.jspa, 2012.

[7] P. Atzeni, V. De Antonellis, *Relational Database Theory*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1993, ch. 1.

[8] A. Olivé, *Conceptual modeling on information systems.* Springer-Verlag Berlin Heidelberg, New York 2007, ch. 1.