

CURRENT APPROACHES TO DATA MODELING

Stanislava Šimonová, Jan Lešák, Martin Kalhous, Martin Vávra
Ústav systémového inženýrství a informatiky, Fakulta ekonomicko-správní, Univerzita
Pardubice

Abstract: *The information systems are most often realised by means of database software. The database systems make increased demands on users which have to know the data model principles. There is a question which form of the modelling approach and which software modelling tool should be selected by a user / manager - whether a standard data modelling approach or a object approach. This contribution summarizes some results of the University of Pardubice projects FG462024 and PR461006, which the students FES were involved in.*

Key words: data modelling, database system, abstract programming, standard data modelling, object approach

1 Correct data, correct program

The information system is a necessary base of any company or any institution. Most users automatically suppose that the information system contains correct data – without thinking about the actual meaning of this term, i.e. what criteria the requirement for correctness should meet. And it is just the fact that the meaning of the term correct data is not understood that results in an incorrect approach while the information system is being designed. The correct data is therefore such data that has been prepared in accordance with the principles of the data model concerned, mostly in accordance with the relational data model principles [4] [5]. Such data shows the minimum redundancy and is consistent.

Similarly is possible to characterise the term correct program. It should be pointed out that the incorrect program does not mean that its results are mistakes. Even an incorrect program may currently provide correct results but it means a danger in the future. The incorrect program is not transparent, uses unusual solutions that are not precisely documented. The danger of such program occurs with any requirement for a change; then the non-transparent relations and unusual solutions will cause mistakes without any doubt. Special unusual program solutions are also purposeless in essence as the company reality is in fact usual and consists of typical situations – products are made and sold, documents are issued, goods are released from the warehouse, etc. The correct program is based on an analysed data model, uses usual transparent solutions, makes use of patterns [3] and there is always a perfect documentation accompanying it.

2 The data modelling process

The data modelling process is a process leading from the first proposal of the future information system to implementation in a certain program environment. The basic modelling principle is that a system model is created before the system itself is formed [1]. Different methods and procedures are defined for data modelling. The current trends bring greater demands on the modelling techniques, which may be referred to as the abstract programming. The analytical modelling is really a kind of programming although on an abstract level in a pseudo-code (the pseudo-code being firstly the structured text, and secondly own UML diagrams).

There grew up basic model theory principles in the development of databases. First principle group supports a creation of certain pure models which decreases a possibility of anomaly origination in the data sets. There are defined special data integrity limitation and optimization. Second principle group is focused on universality for similar systems. Another principle group increases immunity against supposed changes. Database model system differs data modeling (belongs to regular procedures), object programming (UML is a trend) and object-relational access [2].

Our aim was both to find out user-friendly diagram notation of data modelling (with stress to relational database system) and to test software tools suitable for educational usage.

3 Diagrams for database modelling

The graphic way of express, mostly represented in form of diagrams, is used for better understanding and work with basic objects and their relationships. The well-arranged orientation given reality is graphic entry. The most frequent diagram is ER diagram (entity relationship diagram), it always determines the basic constructs, i.e. entities and their mutual relationships. An entity is a distinguishable object of the real world (within the company aspect being analysed), this is capable of independent existence and may be unambiguously distinguished from the other entities. The ER diagram is used in different variants and notations.

Current modelling approaches are enriched with an object viewpoint. For the purposes of this paper we will simplify the object viewpoint to the statement that the object viewpoint is based on objects and patterns. The object is a certain relatively independent component with its own functionality. The pattern means a well-tried procedure that is to be applied again, it is a typical re-use principle if any patterns exist, i.e. well tried procedures defined by someone (on the level of an analysis or proposal or implementation), then our solution is well comprehensible for the others (because they also know patterns). The unified modelling language (UML) is most often used for analytical modelling. This language does not relate to any specific methodology or a life cycle; may also be used for an analysis and proposal of the information system that is to be realised in the relational database system.

There is a whole series of forms of graphic entry and every one of them has their positives and negatives. Every notation has variously subtle philosophy, some is building on the basis of other, some be focusing only on terms in terms of diagrams. The important is to choose a uniform way of the graphic explanation and use our selected notation. We have focused to integrity rules, it means cardinality and partiality.

Chen notation

Let's focus on the Chen notation first, which is the oldest way of graphic entry. This diagram style is quite space exacting, so we usually don't use this notations to construct databases, but it is mostly used in study materials. The intelligibility is an advantage of this notation. The basic scheme of Chen notation is on picture.

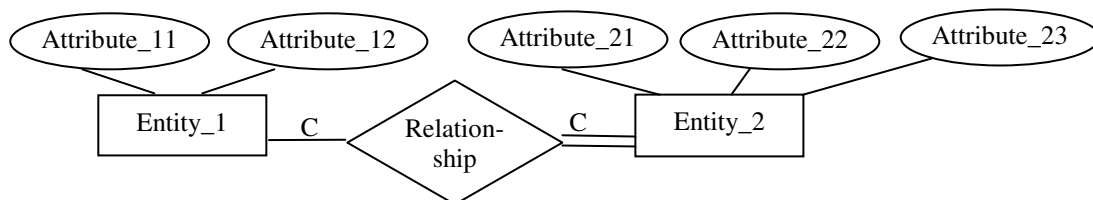


Fig. 1: Chen notation

Oblongs represent entities, single entity attribute is in ellipsis and entities relationships are presented as diamond. There are integrity limits called cardinality, which is used for every relationship type, presented as C in scheme. Here they may come three cases of relationship, 1:1, 1:N, M:N.

The problem of this basic diagram is to distinguish the partiality of relationship; it means obligatory (total) or optional (partial) relationship participation. The solution can be that the line is doubled if the participation is total.

MIN MAX notation

So-called MIN MAX notation is used to distinguish an obligatory and optional membership. This notation is used both in standard approach and in object-oriented models. We want to refer about MIN-MAX notation which uses an ER diagrams and engage just a graphic explanation of integrity limits. As referred, the basic Chen scheme omits partiality, we solved the problem by adding this

notation and it will already compensate cardinality distinguishing obligatory and optional membership. So it is not a new form of graphic explanation, but it is just extending of known model. The MIN explains a minimal count entities occurrence and has 0 value (optional membership), 1 (obligatory membership). The MAX explains maximum count entities occurrence and has value 1 (max one occurrence), N or M (more occurrence). Simple example on drawing 2.

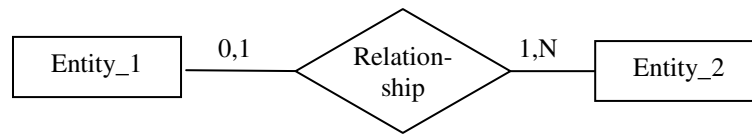


Fig. 2: Min Max notation

This form of graphics explanation is testified for his simplicity. In the standard approach the level of the standard model is not burdened with technology. It means that the analyst does not think about the technology (in our case a relational database system is dealt with) and about its possibilities and limitations. The manager-analyst has the main task which may be named – with a little exaggeration – as “to find out and not to forget anything”. Therefore the important thing is to find and identify everything in company aspects for the future information system (to describe in text and show by means of diagrams). Other following steps are then carried out by IT specialists (designers, programmers), who transform the analytical model into a relational data model. Anomalies in created proposed relations are subsequently removed by specialists through the normalisation process.

UML notation

As next important notation, which we’d like to dedicated is an UML notation - most extended and most elaborated one. It deals with unified programming language, which unite standards notation and object notations of many more. By uniting these object-oriented notation UML allows the basic for standard in a domain of object oriented analysis and proposals made upon the specialist experiences. UML is trying to use graphic syntax from different sources, because this notation is already experienced and there is no reason to change it. UML models can dependent on implementation platform or independence. One of the importances of UML is its independence on methodologies. Maybe it is reason for wide extend as an implementation language. The whole programming language is made upon a three elements: objects, relations and diagrams.

The great priority of UML language and UML diagrams is existence of extendable standard, the support of all evolutionary application cycle, or another system and great support of all different applications. But there is necessary preparation to manage UML. There are different programs to build schemes, which may a little change or combine a single marks. It means that some notations are quite same, but they can be different in some details. The important thing is to choose one notation and use it in all constructions.

In the object approach, the requirements for the user-analyst are considerably higher. He must be informed in great details about rules for creation of selected diagrams within UML, therefore he must know their syntax. He must also take into account the technology used and first of all he must proceed with the knowledge that in fact already now, in the stage of analysis) he creates an information system. In other words, the user-analyst participates in programming. The analytical modelling may be characterised by saying that it is abstract programming in a pseudo-code (the pseudo-code being firstly the structured text, and secondly own UML diagrams).

4 Software tools

There are more software tools for data modelling. Our aim was to test various tools suitable for the users’ first steps in data modelling as well as for educational usage. It means we have focused to test of trial demo versions available on internet.

SmartDraw [10]:

- Description: The SmartDraw Suite Edition helps you draw professional flowcharts, org charts, Gantt charts, mechanical and software diagrams, floor plans, and calendars - any kind of business drawing, chart or E-R diagram-in just a few minutes. The SmartDraw Suite Edition includes more than 63,000 ready - made graphics and templates, wizards for creating your drawings, automatic chart builders, and so much more. SmartDraw provides immediate benefits to both individual users and entire organizations.
- License: Trial version. Restrictions of trial version - program is full functional only for 7 days.
- Our evaluation: program seems to be very simple for use, but the restriction of trial version makes this program unusable for creating school documents, so we cannot recommend.

Case/4/0 [7]:

- Description: Case/4/0 is a tool for software analysis and design, data modeling and data warehousing. Case/4/0 supports modeling the functions, data and behavior of an application system through the use of function structures, information flows, state diagrams, ER models, data structures, data elements, relational models, module structures and type structures. The tool automatically generates code in C/C++, COBOL, PL/1, Java and Visual Basic from the model, as well as database definitions for relational databases. Already existing applications can easily be read into case/4/0 via reverse engineering. With its central repository, the tool ensures the consistency of all of the results at all times – code and model are always in sync.
- The most important features: Graphic tools for system analysis, system design and enterprise data modeling; Screen design and interface prototyping for the graphic design of user interfaces and the definition of application scenarios with the dynamic animation of information flows; Code generation for processing; Reverse engineering of databases and source code for SQL and COBOL; Integrated version management of individual results and consistent configuration of results; Generation of user-specific reports and individual evaluations for MS Word and HTML; Public COM interface for the integration with other tools and user-specific code generation via code scripting; Import/export interface for exchanging development results in distributed projects; Definition of state-dependent access rights to results for users and/or groups.
- License: Demo version.
- Restrictions of demo version: The following commands from the main menu's "System" menu are not supported in the demo version - System/Backup, System/Restore, System/Import and System/Export. Additionally, neither multi-user operation nor the creation of versions or configurations is possible in the demo version. The total number of diagrams and other elements which can be created in the demo version is also limited. Whereas the full version of case/4/0 does not limit the number of elements which can be created (aside from limitations resulting from a lack of system resources), the demo version does set some limits, which vary according to element type. If you reach such a limit, case/4/0 will show you a message box.
- Our evaluation: This program has no time limitations, but number of diagrams and other elements which can be created is 11. This program seems to be usable for creating school documents. Creating of diagrams is quite easy.

CaseStudio [8]:

- Description: This software is for all developers, database designers and for all who want to create and maintain their database structures. Key features include Entity relationship diagrams, support for various databases, generation of SQL (DDL) scripts, reverse engineering, generation of detailed HTML and RTF documentation, data Flow Diagrams, export into XML format etc.
- ER diagrams: Using the graphically very well arranged Entity Relationship Diagrams you can create and maintain your database structures. In your models you will have all the entities, attributes, domains, primary keys, foreign keys, constraints, relationships, indexes, descriptions, notes and other physical and logical data, laid-out in a transparent order.

- License: Shareware. Restrictions of shareware - not possible to save model with more than 6 entities.
- Our evaluation: In this program you can try all features. Classes, attributes and relations can be named according to users need. Czech language is not supported in names of entities and relations. Creating of diagrams is more difficult than in case/4/0.

ArgoUML [6]:

- Description: ArgoUML is a UML diagramming application written in Java and released under the open source BSD License. By virtue of being a Java application, it is available on any platform supported by Java. Software Development Magazine gives annual awards to popular software development tools in various categories. In 2003 ArgoUML was a finalist in the "Design and Analysis Tools" category. It received a runner-up award, beating many commercial tools. ArgoUML includes support for all standard UML 1.4 diagrams and is available in ten languages.
- Weaknesses: Models sometimes cannot be re-opened. Incrementally save-as-copy. Import/Export to Java. No self-calls in sequence diagrams. Must select a class to create a sequence diagram.
- Our evaluation: Program seems to be suitable for needs of education. Classes, attributes and relations can be named according to users need and the number of them is not limited. Program is quite tabular. The installation package has less than 7MB so it is easy downloadable. In problematic situation user can also download the Argo UML manual on the same pages as main program (<http://argouml.tigris.org>). The main disadvantage of this program is that it requires JAVA.

Visual Paradigm [9]:

- Description: Visual Paradigm for the Unified Modeling Language (VP-UML) is a UML CASE tool. The tool is designed for a wide range of users, including Software Engineers, System Analysts, Business Analysts, System Architects alike, who are interested in building large scale software systems reliably through the use of the Object-Oriented approach. VP-UML supports the latest standards of Java and UML notations and provides the industry's full round-trip code generation and code reverse engineering support for Java. The transitions from analysis to design and then to implementation are seamlessly integrated within the CASE tool, thus significantly reducing efforts in all stages of the software development life cycle.
- License: All Visual Paradigm Academic Partner Licenses are granted for academic, non-commercial purposes only. Licenses under this arrangement must certify that they will use Visual Paradigm products exclusively for providing educational services, and do not intend to use them for the benefit of third parties. This license for Visual Paradigm software is designed to guarantee freedom to use the software for academic purposes. Visual Paradigm grants the recipient a royalty-free, non-exclusive, and non-transferable license to use the Software furnished hereunder, upon the terms and conditions set out below. This license applies to the recipient only if is a member of a non-commercial and academic institution, e.g., a university. The license expires as soon as recipient is no longer a member of this institution.
- Standard edition: Visual Paradigm for UML Standard Edition is a toolset that combines professional UML modeling, code reverse engineering and excellent interoperability with other applications.
- Our evaluation: Program seems to be suitable for education. Only several advanced functions require installing JAVA. All classes, attributes or relations can be named according to users need. The number of classes and attributes is not limited. Program is clear and tabular and

control is relatively easy. On official pages of Visual Paradigm International (www.visual-paradigm.com) is also the User's Guide which can be helpful in a problematic situation.

MS Visio [11]:

- Description: Microsoft Visio 2003 is diagramming program that can help you create business and technical diagrams that document and organize complex ideas, processes, and systems. Diagrams created in Visio 2003 enable you to visualize and communicate information clearly, concisely, and effectively in ways that text and numbers cannot. Visio 2003 also automates data visualization by synchronizing directly with data sources to provide up-to-date diagrams, and it can be customized to meet the needs of your organization.
- Our evaluation: Program seems to be suitable as introduction to objective modeling. The number of classes and attributes is not limited. Classes, attributes and relations can be named according to users need. But creating of Use Case and Sequence diagrams is problematic.

5 Conclusion

The analysis cannot be an issue dealt with by a specialised project department, but company users / managers must participate in creation of the analysis, of the data modelling. This is logical as only they know their firm well, only they know the purposes for which the information system should be used; i.e. only they may specify the requirements precisely. Within the analysis, users may use two essential principles – standard data modelling or object modelling. The standard approach to the data modelling makes relatively lower demands on the analyst in the meaning that the analyst concentrates on a description of company aspects and does not consider the context of technology. The object approach to the data modelling means higher demands on the analyst in the meaning that he must have a larger volume of knowledge – the syntax of diagrams used, knowledge of the technology used. The selection of a suitable approach is based on the volume of knowledge the users are willing to invest and on the scope in which they want to ensure the quality of the information environment in the company.

References:

- [1] DATE, C. J. An Introduction to Database Systems, Addison-Wesley Publishing Company, 2004. ISBN 0-321-19784-4.
- [2] DORI, D. Object-Process Methodology. Berlin: Springer, 2002. ISBN 3-540-65471-2
- [3] FOWLER, M. Patterns of Enterprise Application Architecture, Addison-Wesley Publishing Company, 2004. ISBN 0-321-12742-0
- [4] MALAGA, R. Information Systems Technology, Prentice Hall, 2005. ISBN 0-13-049750-9.
- [5] PANUŠ, J., ŠIMONOVÁ, S. Managements new role in designing the information system of the company. In The Proceedings of the 6th Biennial International Economic Symposium SIMPEC 2006. Brasov: Infomarket Publishing House, 2006. ISBN (13) 978-973-8204-82-9, p. 301- 306.
- [6] Software case tool ArgoUML. URL: <<http://argouml.tigris.org>>.
- [7] Software case tool Case/4/0. URL: <<http://www.microtool.de>>.
- [8] Software case tool CaseStudio. URL: <<http://www.casestudio.com>>.
- [9] Software case tool Visual Paradigm. URL: <<http://www.visual-paradigm.com/product/vpuml/>>.
- [10] Software drawing tool SmartDraw. URL: <<http://www.smartdraw.com>>.
- [11] Software tool MS Visio. URL: <<http://www.microsoft.com/office/visio/prodinfo/overview.mspx/>>.

Contact:

Ing. Stanislava Šimonová, Ph.D.
Jan Lešák, Martin Kalhous, Martin Vávra – the students
Institute of System Engineering and Informatics
Faculty of Economics and Administration

University of Pardubice, Studentská 84,
53210 Pardubice
tel.: 466036009
e-mail: Stanislava.Simonova@upce.cz