Comparing tools for enterprise modeling

L. Madarász M. Timko

Dept. Of Cybernetics and AI
University of Technology Košice
Letná 9/B, 042 00 Košice
madarasz@tuke.sk, timko@tuke.sk

Abstract

Paper briefly compares three modern tools used for enterprise modeling. Two of them – Aris Toolset and System Architect were evaluation copies only. Basic reference architectures of modeling frameworks are also summarized and described.

1. Introduction

Enterprise architecture consists of techniques as *knowledge management, business process reengineering, data warehousing* and many others. In compliance with control of complex systems results a need to model such system.

Enterprise modeling is a generic term, which covers the set of activities, methods and tools related to developing models for various aspects of an enterprise [6.]. At present time, when information technologies are in the first place in a chance to achieve strategic goals, there is a strong pressure for enterprises to create and maintain models of whole businesses.

This means to have common perception of all parts of enterprise. It does not matter if organizational structure, data model, principles of operational procedures, regulation texts or codes of application programs are taken into account, the view from any aspect or understanding of any person for the same thing must be the same. This is the aim of enterprise modeling.

Tools for enterprise modeling are used for creating such models. They also can help with implementation of information systems, help with maintaining complexity of enterprise parts and they can keep know-how.

There are many of such tools spread in the world. Each of them is implementation of enterprise architecture and uses techniques of those architectures. Reference architectures are intellectual paradigms, which facilitate analysis and accurate discussion and specification of a given area of discourse [2.], [5.], [6.].

2. Enterprise Modeling

In recent years, new tools and technologies used to understand modern enterprises have risen including process-oriented modeling, objected-oriented modeling, knowledge management and data warehousing. The purpose of enterprise architecture is to integrate different types of technologies into consistent and usable describe of the enterprise as well as to provide logic structure appropriate for classification and organization of the enterprise.

Enterprise integration means integration of both information and material flows of the enterprise and also provides efficient information exchange within the enterprise with the help of information technology (IT).

The aim of modeling and optimization of enterprise processes is to take the enterprise or organization from earlier defined strategic goals and visions to materialize in developed and described processes, which developed ideal data structures, functions and organizations adjusts to function ability, real possibilities of IT and application software used.

There are two basic trends in modeling. The first is Enterprise Ontology [2.], [6.] – formal describe (provides standard shared terminology and semantics) of units and their properties. The second one is computer or graphical models used to define, analyze or visualize "things" to be modeled. All the three tools are from second group of principles.

Visual Modeler is not implementation of enterprise architecture. But it can implement any architecture.

3. Reference Architectures

In development of enterprise modeling and enterprise integration it is meaningful to consider, that a lot of reengineering projects is actually similar. Thus they could be get-together, standardized and reused instead of to be developed from scratch all over again.

There are many enterprise architectures in the world. Some of them are more popular in Europe, such as CIMOSA or PERA, and other are used mainly in USA (GERAM). In next sections two of them are briefly introduced.

3.1. ARIS

Aris means Architecture for integrated information systems [10]. It is structured into four **views** and three **modeling levels**. The four views are:

- The **function view**, which is used to define the function model as a hierarchy of functions.
- The **data view** is used to define semantic data models (in terms of entity-relationship diagrams).
- The **organization view**, which is used to define the enterprise structure summarized by an organization chart, the network topology and the physical network implementation.
- The **control view**, which federates the architecture and is related to the three other views.
- **Output** view, which covers all input and output material and non-material flows, including money flow.

ARIS, instead of focusing on computer-integrated manufacturing systems, deals with traditional business-oriented issues of enterprises. The focus is on software engineering and organizational aspects of integrated enterprise system design. ARIS is open architecture in the sense of formalism used within the various views and levels of the architecture. The architecture is supported by ARIS-Toolset tool. Our department has a copy of the product and also has joined an interconnection for collaboration with IBCS-Group (creator and provider of tool).

3.2. Zachman architecture

John A. Zachman stated that the framework as it applies to Enterprises is simply a logical structure for classifying and organizing the descriptive representations of an enterprise that are significant to the management of the enterprise. It was derived from analogous structures that are found in the older disciplines that classify and organize the design artifacts created over the process of designing and producing complex physical products.

A Framework divides an enterprise into manageable chunks. It also helps to prevent the isolation of a single problem area from the other areas its change or elimination might affect. Framework overall is represented by a table. Framework is composed of 6 perspectives and from 6 focuses (e.g. 6 rows and 6 columns).

Each cell in the architecture framework represents the intersection of a particular focus and a perspective. Each focus (the question what, how, where, who, when, and why) is depicted in a column and each perspective (point of view) in a row. The perspective determines the kind of information that will be recorded in a row and/or cell. Without a proper perspective, information can never become knowledge. The perspectives define the point of view or the level of abstraction for the information contained in the cell.

3.3. Visual Modeler

Visual Modeler is a graphical object-modeling tool that is integrated with Microsoft Visual Studio 6.0. Visual Modeler is a subset of the Rational Rose tool, both developed by Rational Software Corp. Visual Modeler allows fulfill the promise of object-oriented programming by quickly and easily creating applications that are maintainable, have a long lifetime, and are comprised of components that can be reused in other applications.

A model of a system describes a view of the system on a higher level of abstraction than the source code. There is no need to have to write the skeleton source code. User can simply generate all class modules, data members, and method specifications. Then, the implementation phase means to refine the generated source code until we have an executable system. Also if we have an undocumented system, we can simply use the reverse-engineering feature

to create a model of that system, and continue the development work in the new model.

Because it is easier for other people to understand the system's design from our diagram we have added this tool to our comparing.

· development team have a common diagram notation, which means that everyone understands the diagrams without further explanation.

· can concentrate on what the system is supposed to do, and not how things are going to be implemented.

· it is a quick job to change the diagrams and specifications when the requirements change, compared to the time it takes to rewrite the corresponding code.

· can try various design approaches and communicate our ideas among the development team members.

The model is used when we discuss the structure of the system with others, or when we want new project members to quickly understand the system, as well as during further development of the system to get a quick overview of how things are related in the system. The three-tiered architectural approach supports the object paradigm for the large client/server community. The application development model, often referred to as the Microsoft Three-Tiered Model, supports the creation of large, complex client/server applications.

These are the main features of Visual Modeler:

- Class diagrams - the design of the system you are about to develop in terms of a model, using a high level of abstraction. The diagram notation provided by Visual Modeler is a subset of the modeling constructs defined by the Unified Modeling Language (UML).
- Code generation - the ability to automatically generate Visual Basic and Visual C++ code from the design model you have created with Visual Modeler.
- Reverse engineering - the ability automatically create or update the model with changes made to the Visual Basic code.
- Round-trip engineering - the combination of modeling, code generation, coding, and reverse engineering.

In Visual Modeler a system is modeled from three different views, each one with its own purpose:

- Logical view - describes the logical structure of the system – the classes and their relationships.
- Component view - describes the physical structure of the system – how the system is divided into .exe files and DLLs.
- Deployment view - shows the system's nodes, the connections in between, and the allocation of processes to nodes.

Each package, class, method, property and relationship in the model is completely described by a specification.

4. Creating of models

4.1. ARIS

The focus is in identification of processes and the goal is to achieve process control of organization (to define process and documentation flow) thus the initial event should be continued by a sequence of processes leading to final event. The depth of hierarchization of a model must be relevant to the level of describe competence of meta-model of ARIS. Methodic and models used:

1. Organizational structure represents hierarchical structure of organization. Organization unit is step of hierarchy of organization, which is independently responsible for some activity.

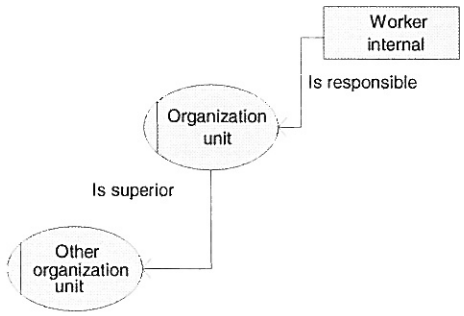

Fig. 1 Organizational structure

2. E-R model (Entity Relationship model)

This model was used to describe the objects and the relationships between them. It describes relations between entities (abstract describe of entities, attributes and relations of them). Entity "*Subclass*" is superior entity to "*First entity*" and the "*Second entity*" and there is a

relationship between them. The second entity is described by *Attribute 1* and *Attribute 2* (Fig. 2).

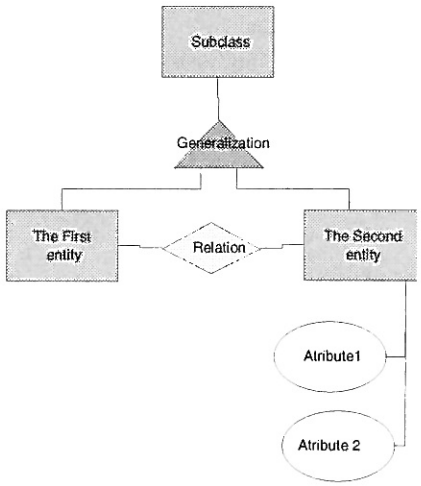

Fig. 2 E-R model

3. EPC Model defines and describes a sequence of processes from initial event to final event. Processes figured by EPC model can delete unused functions and independent events to obtain customer adapted processes. To every function in EPC model can be in ARIS Toolset made text describe (Fig. 3).

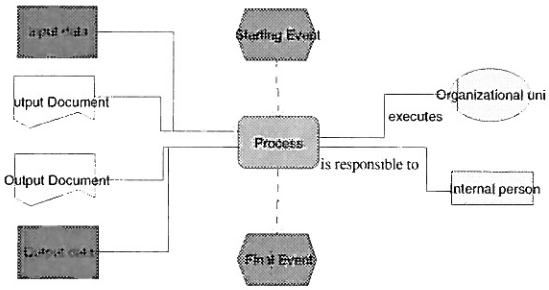

Fig. 3 EPC model

With modeling in ARIS we mean using of elements to describe enterprise processes and phases defined in ARIS. Modeling is a creative process and thus there are no rules. Observation of some axioms can on the other hand help to understand models created by others persons.

Concept of ARIS was created on meta-level independent on applications. On the level of meta is defined only general term *Object type*. Information model is developed on the meta level. On the base of symbol it can be assigned to every model corresponding level of abstraction. Normally there are created models on the level of type in the process of definition of enterprise process. Models than are not so sensitive to change of instances.

4.2. System Architect 2001

Many entities and descriptions (state and goals of the enterprise, constraints, ...) represents System Architect in the text form as a "*Description*" of defined part. It is an attribute of every item to which whatever describe can be assigned in the base of relevancy and efficiency.

Only some models are represented by graphic diagrams. For example:

1. Data flow

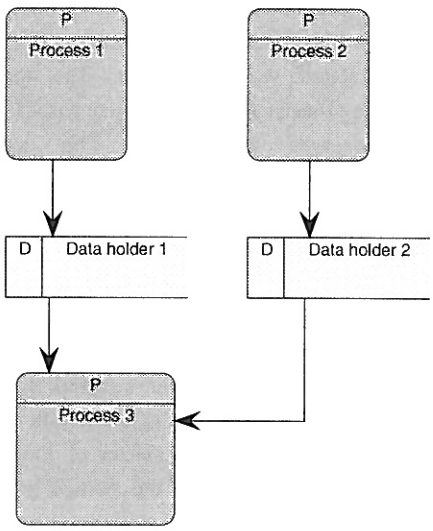

Fig. 4 Data flow diagram

2. E-R Diagram has the same scope, only a graphic representation is different.
3. Organizational view – almost the same as those of ARIS.
4. Process chart – it is the same as EPC model in ARIS (it shows process flow in the enterprise), but the graphic representation differ.

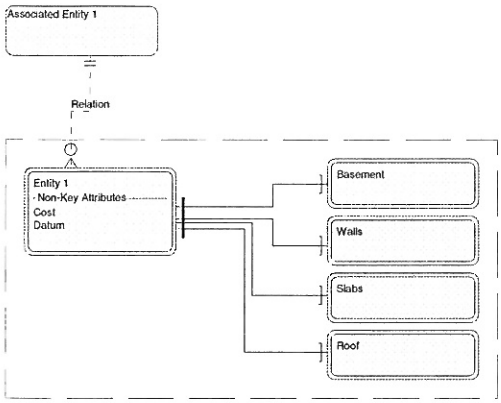

Fig. 5 E-R Model in System Architect

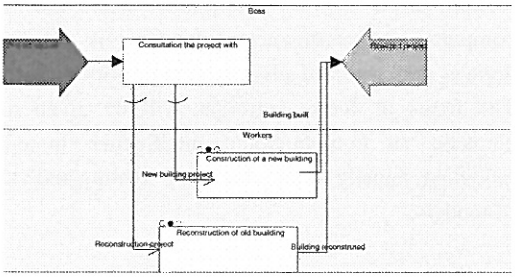

Fig. 6 Process chart

System Architect 2001 offers wide resources of enterprise modeling, object-oriented modeling, development and improvement of enterprise components, relational data modeling, structure analyses and design. These resources are supported by multi-user environment based on shared, user-adaptable database in real time.

It consists of rich collection of components providing describe, improvement of modeling and creating of complex enterprise systems. All informations are saved in multi-user repository named *Encyclopedia*.

Support of modeling of enterprise processes includes wide covering of CSC catalyst structures, IDEF methodologies and huge amount of links to simulation tools of other producers. Modeling of objects and elements is possible thanks to complete support of UML rotation, with ability of direct and reverse engineering in many programmic languages. Modeling of business data includes *Entity Rotation Models* from many sections, separate physic models, generic of schemes and back data acquisition.

System Architect offers drawing tools for graphic composition of models representing enterprise system, applications or databases, which should be modeled. Through dialogs is possible to edit all definitions in Encyclopedia. Properties representing all types of definitions can be customized thanks to wide scale of metamodels. This tool offers package of matrix editors by which it is possible to assign informations about model before the diagram is created. This concept provides to fixate a wider view of problem before user plugs into detail analyses.

Every object (graphic or non-graphic) has in System Architect at least one property – *Description*. Some graphic objects (entities, tables) has many properties that are alternating (creating system for Oracle is different to one in MS Access).

System Architect has built support of numerous common used of software products. For example:

1. Microsoft Visual Basic – it is used for creating of user scripts,
2. Microsoft Word – it is possible to use it for creating of reports of analyses results,
3. HTML generator,
4. Screen Painter – simple tool for creating graphic user interface in windows-style.

4.3. Visual Modeler

A model consists of diagrams and specifications, which provide a means of visualizing and manipulating the model's elements and their model properties. Since diagrams are used to illustrate multiple views of a model, icons representing a model element can appear in none, one, or several of a model's diagrams. The application therefore enables user to control which element, relationship, and property icons appear on each diagram, using facilities provided by its application window.

The application's method recommends the use of static and dynamic views of a logical model and a physical model to capture the in-process products of object-oriented analysis and design. Using the notation, the application enables user to create and refine these views within an overall model representing problem domain and software system.

The three-tiered architectural approach supports the object paradigm for the large client/server community. The three-tiered model addresses the historic lack of concern

for basing an application's construction upon a layered software architecture that isolates the user interface and database from the domain of the problem being modeled. Visual modeler is a powerful tool for creating large, team based applications. Support of UML makes this tool independent on platforms, consistent and integrated tool for creating complex applications.

5. Evaluating of models

As every tool has a different rules and formalisms, it is not simple to evaluate models each other. A major problem concerns ability to assess the quality of models and their complementarity. There are no defined system or methodology for evaluating or comapring methods. Only some aspects can be considered and compared.

ARIS Toolset is an abstract modeling tool. Almost every modeled part and their relations can be graphically depicted. The basic part of *ARIS Toolset* is *ARIS Designer* – modeling module. Models are accessible through reports to other modules (for example *Simulation* – simulation module, *WebLink* – module for connecting with http protocol, *Lotus Connectivity*). Large disadvantage is package of reports written in own script language. This approach decreases possibility of changes. On the other hand, transport of models is very simple. We can take ARIS as very abstract tool, we can hardly automatically implement model. Big advantage is specification of view in the process of modeling (organization, data, function, control) and graphic definition of relations.

System Architect 2001 unlike ARIS Toolset describes many items in text form. This approach vanish simplicity, but it provides accurately describe of system. Sometimes offering looks very blind and it is hard to choose the right model although it is important. It makes possible to define relationships only between particular types. System Architect works on lower level of abstraction. It makes possible simpler automatic implementation of system. It can generate source code in many implementations. It uses Visual Basic (improved flexibility) and the tool has own Visual Basic editor and compilator.

System architect is a subset of Rational Rose tool. It is strong tool for modeling, creating and engineering of application. Using of UML makes from this tool integrated and complete tool to software development. But it is not tool for enterprise modeling oneself. It can be used to create parts of information system, maintain it and can help in the process of development and analyses.

6. Conclusion

Paper briefly compared three architectures used in the process of enterprise modeling as well as implemented software. At present time our attention is mostly concerned to reach optimization of enterprise activities in area of decision processes, reducing the time of preparing and final decisions itself, speeding up realizations and increasing the quality of decisions in conditions of medium-sized serial enterprises with the use of computational intelligence. A lot of work still has to be done as only a few of surveys were performed in this area. The focus of future activities will be given to apply elements of computational intelligence in enterprise modeling, namely in the process of creating and validating of models.

References

[1.] Enterprise Modeling – Aligning Business and Information Technology, Popkin Software's home page, www.popkin.com

[2.] M.S. Fox - M. Gruninger: *Enterprise Modelling.* In: AI Magazine, AAAI Press, Fall 1998, pp. 109-121.

[3.] A.W. Scheer: *ARIS – from enterprise processes to application systems.* Springer-Verlag, Berlin – Heidelberk, 1998, ISBN 80-238-4719-8, 187pp.

[4.] System Architect 2001 Tutorial, Popkin Software's home page, www.popkin.com

[5.] M. Timko: *The Using of Computational Intelligence in Enterprise Modeling.* (In Slovak) Dissertation thesis , Dept. of Cybernetics & AI, TU of Kosice, 2000, Slovak Republic.

[6.] Francois B. Vernadat: Enterprise Modelling and Integration : principles and applications. Chapman & Hall, London, 1996, ISBN 0-412-60550-3, 509 pp.