Elektrotehniški vestnik 68(2–3): 109–114, 2001
Electrotechnical Review, Ljubljana, Slovenija

Enterprise modelling with UML

Aljaž Zrnec, Marko Bajec, Marjan Krisper

University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, 1001 Ljubljana, Slovenia
E-pošta: {aljaz.zrnec, marko.bajec, marjan.krisper}@fri.uni-lj.si

Abstract. Making effective project selection decisions in an enterprise requires a clear idea of where the enterprise is in the current state, what its vision for the future is, and how to make a transition to its desired future state possible. A strategic plan is a document that encompasses this information and is produced as an output of corporate strategic planning. In this paper we examine business modelling as an integral part of strategic planning, where models of a current and future enterprise are developed and refined. The question that we address in this regard is, whether the unified modelling language (UML) notation, the today's defacto standard in engineering modelling, can serve as an appropriate language for enterprise modelling, in particular for corporate strategic planning. In the paper we discuss types of models that are usually used in strategic planning to describe the enterprise from different perspectives and suggest some extensions to the UML notation to make it more suitable for this purpose.

Key words: strategic planning, unified modelling language (UML), notation, standard, business modelling

Poslovno modeliranje z jezikom UML

Povzetek. Sprejemanje učinkovitih odločitev o izbiri projektov v velikih podjetjih zahteva jasno predstavo o tem, v kakšnem stanju se podjetje nahaja v sedanjem trenutku, kakšna je vizija prihodnosti podjetja in kako izvesti prehod iz sedanjega v želeno prihodnje stanje. Strateški plan je dokument, ki nastane kot rezultat strateškega planiranja. Vsebuje informacije o sedanjem stanju podjetja, njegovi viziji in načrt, kako priti do želenega stanja v prihodnosti. V tem članku smo se omejili na poslovno modeliranje, ki je del strateškega planiranja. Zanimalo nas je predvsem, ali lahko notacija jezika UML, ki je že močno uveljavljena, služi kot primeren jezik za izdelavo strateških načrtov velikih podjetij. V članku smo se omejili na poslovno modeliranje in raziskali zmožnosti jezika UML za modeliranje različnih pogledov na sistem. Kjer je bilo mogoče, smo uporabili obstoječe diagrame jezika UML, ki najbolj ustrezajo potrebam po jedrnati predstavitvi določenega pogleda na poslovanje. Kadar se nam je zdela notacija pomanjkljiva, smo predlagali razširitve UML notacije.

Ključne besede: strateško planiranje, UML, notacija, standard, poslovno modeliranje

1 Introduction

The purpose of every business is to survive and make profit. These are the goals toward which its energy and enterprise are directed. Running business today is much more competitive than ever and constantly requires business people to acquire and adapt to new business logic. To remain competitive, companies must assess the quality of their products and services and consider the world around them. They must examine their products and services to find out if improvements are possible [3].

Business people must evaluate information systems in their companies. They have to keep asking: Do our information systems effectively support the way of working? Are the systems flexible enough to adapt business changes easily and rapidly? Is our information system adequate? Does it support information use as an important strategic resource? If companies want to stay competitive, they have to rethink the ways in which they do business or even the types of business they do. This undoubtedly reveals the need for a coherent and efficient business strategy.

What enterprises usually do in this regard is strategic planning, in which they develop the current and future enterprise models and set the strategy for transition from the current to a desirable state. Corporate strategic planning is a difficult and delicate process as it involves business people from the highest positions in the organisational hierarchy and produces outputs that serve as a plan for selection of development projects. In this paper we focus on activities that correspond to enterprise modelling, trying to examine the language that we use to present the enterprise from different perspectives. Like in conceptual modelling of an information system, in enterprise modelling a vast number of diagramming techniques and methods can be applied. Our experiences in corporate strategic planning for companies from dif-

Received 18 December 2000
Accepted 5 Februar 2001

ferent areas, such as telecommunications and health service confirmed in this respect the adequacy of the information engineering methodology and its diagramming techniques, such as function decomposition, dataflow diagrams, entity-relationship diagrams, etc. Today, however, UML is becoming the number one in engineering modelling, and we ask ourselves whether it has potentials to be used as a language for business modelling.

In the paper we gradually describe techniques used to model an enterprise from different perspectives. In our selection of sub-models, or better said views on the enterprise, we follow our own methodology for strategic planning which we developed as an integral part of a unified methodology for information system development (UMISD) [1]. For each view we suggest a suitable graphical representation based on the UML notation.

2 Business modelling

Modelling is a simplified representation of the complex reality that enables us to eliminate irrelevant details and focus on one or more important aspects at a time. In business modelling we focus on an enterprise rather than on its information system and develop the model that shows the enterprise from various points of view.

While ideal business model consists of one diagram that includes all the important aspects of a business, in reality this is usually not possible. Instead, we use several different views that capture specific information about one aspect of the business. Each view is called a model and includes a graphical representation and description of its elements. According to UMIST, the business model consists of the following sub-models:

- Organisation structure model,
- Organisation function model,
- Organisation dataflow model,
- Organisation business process model, and
- Organisation data model.

The meta-model exposes many complex relationships among concepts used as model elements in enterprise modelling. A brief description of the meta-model semantics is given in the next paragraph.

The business unit describes an organisation, or department that has a goal of performing business activities. Business units can be arranged in a hierarchy that reflects the organisation structure. It is a central entity of the model and is in relationship with several other concepts:

- Location: location accommodates the information about the business unit location. Note that the business unit can be distributed across many locations.
- Vision, Mission, Goal: vision and mission are significant elements that tell why a certain business and its processes and tasks exist. The vision describes the purpose of a business while a mission documents the necessary achievements to fulfil the vision. Mission is further decomposed into one or many goals that represent desirable states of a business (states that a business wishes to achieve, e.g. its position in the market).
- Organisation role: an organisational role represents one or more human resources available within an organisation. Each organisation role is therefore connected with a person, representing a human actor that participates in business processes. As shown, a human actor may have several different roles within an organisation.
- Resource: resource is an abstract element. It represents an entity that participates in the tasks that constitute or are related to business processes. Resource can be a physical resource, information resource or organisational unit.

Important relationships exist also among elements, such as entity, function, organisational role and business unit. Each of those elements is associated with others through many-to-many relationships. For instance, a function uses several entities when being executed, and an entity serves as an information resource in several functions.

In strategic planning, several other concepts are in an important position in the model. In this paper, however, we only focus on elements that are used in business modelling.

3 UML diagrams for business modelling

UML is intended for use in the process of the software analysis and design. UML's notation encompasses diagrams which are very flexible and can be used not only in software engineering. With additional mechanisms provided in UML notation, we can define new building blocks, which can be used in diagrams to extend their capabilities of representing specific aspects of the modelled system.

3.1 Organisation structure model

The organisation structure model is composed of an organisation diagram and description of organisation units that form the organisation system. The structure of the organisation diagram is hierarchic, meaning that one organisation unit can be further decomposed into several sub-organisation units.

We think that class diagrams are convenient UML diagrams for modelling the organisation structure. Class

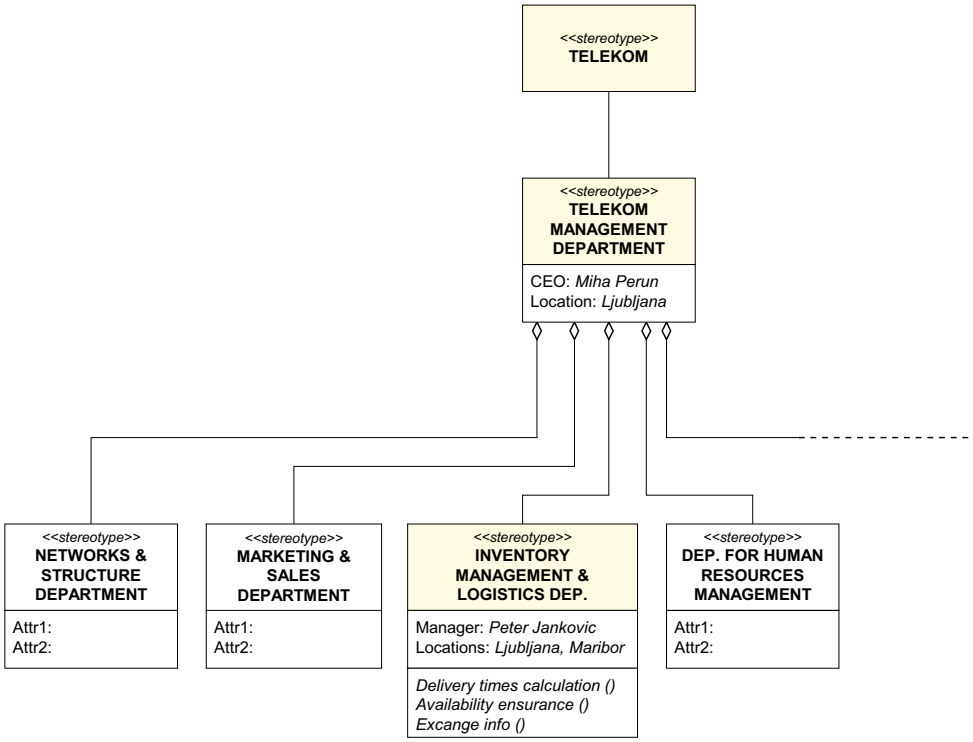


Figure 1. Organisational structure - an example

diagrams in UML are used for a graphical presentation of the static design view of the system. They show a set of classes, interfaces, and relationships. Every class is rendered as a rectangle, usually including its name, attributes, and operations. The notation of class diagrams is too strong for our needs, so we rather define a new stereotyped class for the organisation unit.

The organisation unit is in many-to-many relationships with the location, role and function. This means that the relationships between these building blocks are complex. For example, a particular organisation unit can be distributed across several different locations, while each location can host several organisation units. Similarly is true for the roles and functions.

We can use an attribute that lists all the locations where a certain organisation unit resides. Additional attributes are needed to list the roles important for organisation units. Typically we name the manager, but if we find another role significant for our needs, we name it, too. We can name the same role in several organisation units. Functions covered by an organisation unit can be also listed. We define them in the field for operations.

The organisation unit hierarchy is shown using aggregation and sometimes composition. Generally we use aggregation, which specifies a whole-part relationship between the aggregate (the whole) and a component (the part) [Grady at all, 1999]. Sometimes, if we want to explicitly present a solid structure between the whole and the parts, we use composition. Composition is a form of aggregation with strong ownership and coincident lifetime of the parts by the whole [2]. This kind of relationship is used when we want to show that a specific organisation unit can not be moved among superior units.

Instead of class diagrams we can also use packages. A package is a special mechanism for grouping things in UML. Package diagrams would be useful on the level of strategic planning, but they are not as much expressive as class diagrams.

Every organisation unit must also be described. The best way for describing is using a dictionary. Some of you may wander why we don't use a special symbol for adding notes to every organisation unit. The answer is that every diagram has to be simple and understandable. So we rather use a dictionary. An example of the organisation diagram is shown in Figure 1.

3.2 Organisation function model

The organisation function model shows functions of an organisation system. It is composed of a function diagram and description of functions. A function is defined as a specific and continuous activity defined with the nature and scope of work. It can be decomposed into smaller units-subfunctions which can be further decomposed into activities [1]. We usually use decomposition for presentation of organisation system functions. Functions are usu-

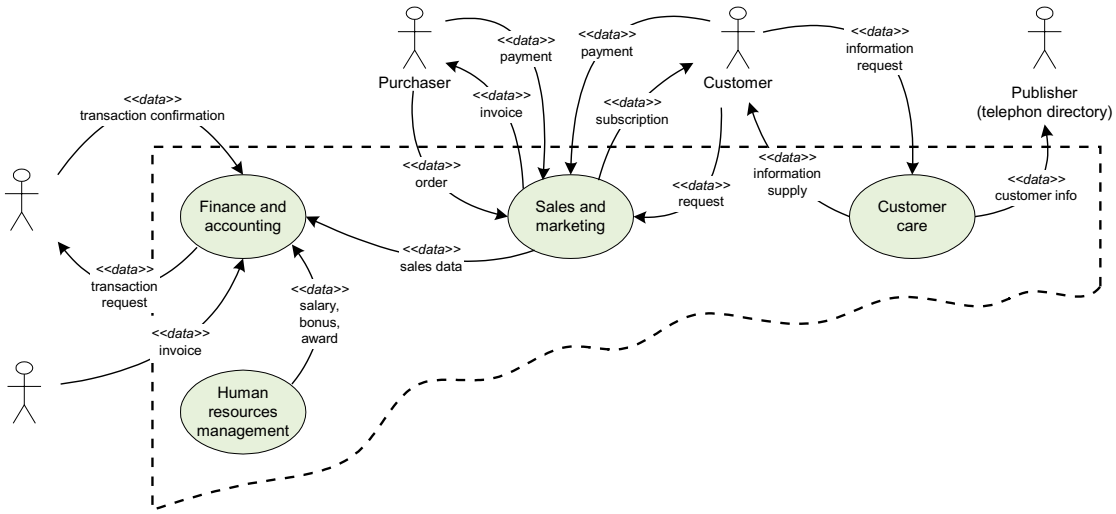


Figure 2. Dataflow model - an example

ally named with verbs or gerunds.

Similarly as for organisation diagrams we think that class diagrams are the most convenient diagrams for modelling function decomposition. We define several stereotyped classes: global function model, functional area and function. Each of these stereotypes describes functions on different level of decomposition. The global function model generally shows the name of the company, because we can't usually name the general function of the system with one verb, the functional area describes a group of connected functions, and the last stereotype - function describes the elementary function of the system.

The function is in a many-to-many relationship with the location and organisational role. A specific function can be distributed across several locations and one location can host several different functions. We describe locations of a specific function with an attribute that lists them all. Similarly we can list important roles appearing in one function, or we can list roles for each location separately. Usually we list the role - supervisor of the function, but if there is another important role, we also list it.

Functions can be further decomposed into several activities. We can use stereotyped operations for defining these activities in the stereotype function.

Functional hierarchy is generally shown as an aggregation or composition. We suggest that a composition is used only if a specific ownership has to be emphasized. In other words, if we want to show that specific function cannot be moved among superior functions. The function diagram has a similar form as the organisation diagram.

All descriptions of the organisation function model are found in a dictionary, where we describe building blocks of the function diagram. In this way we assure simplicity and comprehensibility of function diagrams.

3.3 Organisation dataflow model

Every system exchanges data with the surrounding environment and between functions and data stored inside the system. These exchanges are called dataflows. They are typically described with the organisation dataflow model that is composed of a dataflow diagram and description of its building blocks.

Dataflow models are developed step-by-step starting with the context diagram that represents the business as the root process. Its purpose is to set system boundaries. The root process is then decomposed into sub-processes that typically correspond to functions that belong to the first level of the function model (function areas). In the next step, each function is decomposed further, showing additional details of the dataflows.

For graphical representation of dataflow models we suggest to use the use case diagrams for both, the context level and its sub-models. Functions can be described as use cases and dataflows with stereotyped associations. For this purpose we introduce three additional stereotypes: data, document and physical resource, each of which describes a specific type of exchanging data. External entities can be described as actors. An example of a part of the dataflow diagram is shown in Figure 2.

3.4 Organisation business process model

Business process is a set of connected activities, performed in a business system that has direct or indirect influence on extra value while attaining a common objective of a business system. The results of the business process analysis, performed in strategic planning, are represented with an organisation business process model that is composed of a graphical representation and description of most important business processes.

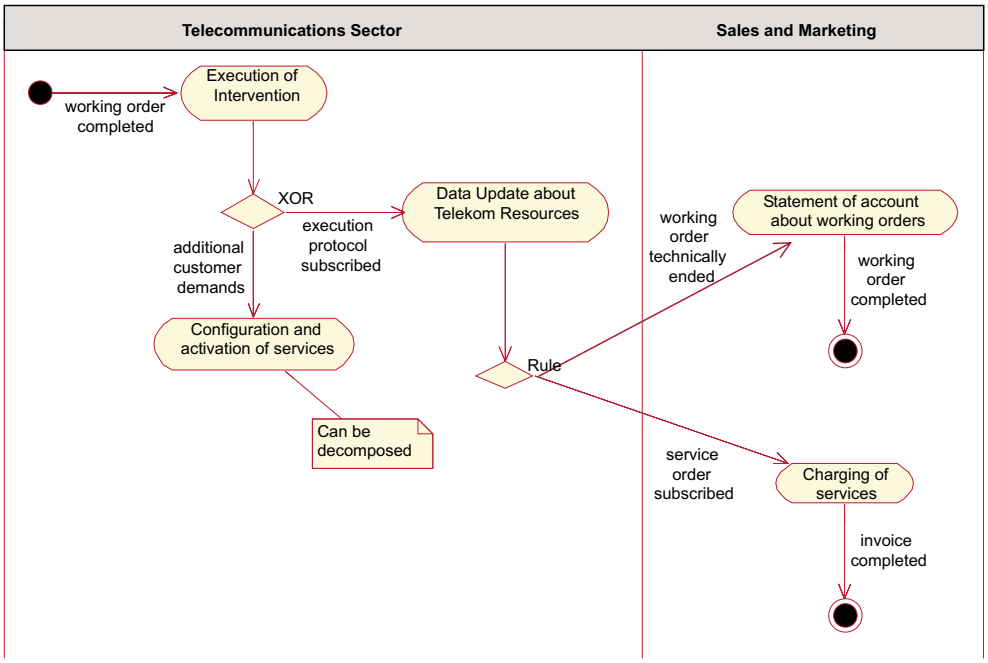


Figure 3. Business process model - an example

How can we model business process with UML? Characteristics of the business process could be presented with use case diagrams [5]. In this case, every step in the process would correspond to one use case and the actor that performs that step. Dependencies in the use case diagram would define the expected flow of business events. The flow of events could be formalized with rules for transitions. Because of the business flow dynamics and complexity of transition rules, use case diagrams are not a convenient technique to represent business processes. Instead we suggest using activity diagrams.

The activity diagram is a specialization of the state diagram intended to model the business process. The notation of the activity diagram is clear and understandable. That is a very important feature because business processes are modelled by people from the business world not computer specialists. The notation is oriented towards the representation of a business instead of technical needs.

The core elements of an activity diagram are: activity or sub-process, dependency between activities, decision activity, synchronization and forking bar, swimlanes markers between roles, states at start and end of a process. An activity is a discrete step in the context of the activity diagram and it may be connected with other activities by a set of dependencies. In case of a too complex activity, we may decompose it with another activity diagram. Dependencies between activities model the events that occur at the end or before the activity. The decision activity and synchronization bar make possible to define an exact sequence of activities.

Decisions are modelled by conditions and represented as a diamond. They specify alternate paths based on some Boolean expressions (AND, OR and XOR). In business process modelling we might encounter workflows that are concurrent. In UML we use synchronization and forking bars to specify joining and forking of these parallel flows of control. Synchronization and forking bars are rendered as a thick horizontal or vertical line.

Swimlanes are used to specify the roles participating in a business process. The role is a subject that takes part in executing the activity or it is responsible for it. This subject can be a person, organisation unit, working place or functional area. If we are interested in functional areas that take place in a particular business process, then the swimlanes will represent functional areas. But if we are more interested in people participating in the process, then the swimlanes will represent that people. We can also use the combination of both previous examples.

When we have drawn all important business processes we also have to describe them. Descriptions of all important business processes are to be found in the dictionary.

For each business process, we make out its own activity diagram to assure clarity and comprehension. An example of the activity diagram which models a business process is represented in Figure 3.

3.5 Organisation data model

Organisation data model defines entities in an organisation system on the highest level of abstraction. In

strategic planning we do not try to identify attributes, because we have to stick to a clear presentation of the data model. The organisation data model is composed of entity-relationship diagram (ER diagram) and description of entities.

Strategic planning only requires identification of main entity types. Because of this requirement, every entity in the class diagram can be presented with a symbol of the class, without definitions of attributes and operations. Every class in the class diagram is marked as persistent (a standard tagged value), which means that its objects are saved in the database.

We show relations between entities with associations, aggregation and generalisation. Relationships have several properties but in the strategic planning phase we only use cardinality.

Package diagrams are also a very convenient technique for data modelling, especially for purposes of strategic planning. We usually produce several ER models, for each functional area separately. Then we have to put them together into a complete ER diagram. To avoid complexity of the integrated diagram, we can hide the implementation of each partial ER diagram in its own package. Now we draw an integrated data diagram with packages and dependency relationship among them. With the dependency relationship we show dependency among packages. For example, if package A needs package B to operate correctly, then we say that package A depends on package B. We have to mention, that there can exist cyclic dependencies among packages.

Two packages interchange data over common entities. Common entities are entities found in both communicating packages and we describe them with classes. In strategic planning we do not draw each of these classes separately. We replace them with a symbol for an interface amongst two packages instead. A package can have more than one interface. In the end we have to describe each entity. We do that in the dictionary.

4 Conclusions and outlook

Growing popularity, expressiveness and extensibility mechanisms of the UML modelling language, force us to consider its potentials in all modelling areas, thus even in business modelling. In the paper we focused on business modelling as an integral part of corporate strategic planning and examined possibilities of the language to model different aspects of an enterprise. We discussed several sub-models and suggested graphical representations for specific views of the system. Where possible, we used the existing UML diagrams that best fit the needs for a concise representation of a specific aspect of the business, however where the notation was found insufficient, we proposed extensions to the UML notation.

Since business models produced as an output of corporate strategic planning primarily serve as a means of communication with the enterprise top management, the selection of graphical representations has to be made carefully, considering their expressiveness, conciseness, and understandability. Our intention in this respect is to gain this information from real cases where executives will have to choose techniques that they found the most useful and appropriate.

5 References

[1] M. Krisper, *The Unified Methodology for Information System Development, Working document, version 3.0*, Government Centre for Informatics, Ljubljana, 1999.

[2] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modelling Language - User Guide*, Addison Wesley Longman, Inc., 1999.

[3] H. Eriksson, M. Penker, *Business Modelling with UML - Business Patterns at Work*, John Wiley & Sons, Inc., chapter 1, pp. 1-11, 1999.

[4] F. Fowler, *UML Distilled: Applying the Standard Object Modelling Language*, Addison Wesley Longman, Inc., 1998.

[5] C. Marshall, *Enterprise Modelling with UML - Designing Successful Software through Business Analysis*, Addison Wesley Longman, Inc., chapter 3, pp. 63-73, 2000.

Aljaž Zrnec received the B.Sc. degree in computer science in 1999 from the Faculty of Computer and Information Science, University of Ljubljana. In 1999 he became a research assistant. His primary research interest includes relational databases, distance learning and strategic planning.

Marko Bajec received the B.Sc. and M.Sc. degrees in computer engineering from the University of Ljubljana in 1996 and 1998 respectively. In 1996 he became an assistant at the Faculty of Computer and Information Science, University of Ljubljana. His primary research interest includes information systems development and information systems reengineering. He is a member of the Slovenian Society of Informatics.

Marjan Krisper received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of Ljubljana in 1971 and 1977, respectively. In 1990 he received a Ph.D. degree in computer science from the University of Belgrade. In 1972 he became a research assistant at the University of Ljubljana and in 1982 an assistant professor. Since then he has been teaching courses on information systems and information system development. His main research interests are information systems, information systems development and renovation, and strategic planning.