

Business Modeling with UML

Hans-Erik Eriksson and Magnus Penker, Open Training

Hans-Erik Ericsson is found and Chairman of Open Training Magnus Penker is CEO of Open Training. **In order to keep up and be competitive, all companies and enterprises must assess the quality of their products and the efficiency of their services. In doing so, they must consider what is happening in the world around them, and they must also take an introspective look at their products or services. Is their internal operation working smoothly? Can they improve their product or service? Is their production running as efficiently as possible? Can they expand their product or service portfolios to reach new markets and customers?**

In addition to products and services, today's businesses must also evaluate their information systems. Do the information systems effectively support their way of working? Do the systems adapt easily to change? Is information used as an important strategic resource in the business? Is the information adequate and correct? In many of today's businesses, information systems no longer merely support the business. Increasingly, they are becoming an integral part of it. All businesses make some use of information technology, and it is important that their systems are really built to support the businesses of which they are an integrated part. The business is what ultimately defines the requirements on the information systems, and creating software without a proper understanding of the context in which that software is to operate is a dangerous adventure.

In order to get such an understanding, it is essential to make a model of the business. A *model* is a simplified view of a complex reality. It is a means to creating abstraction, allowing you to eliminate irrelevant details and focus on one or more important aspects at a time. Effective models also facilitate discussions among different stakeholders in the business, allowing them to agree on the key fundamentals and to work towards common goals. Finally, a business model can be the basis for other models, such as models for different information systems that support the business. Modeling (e.g., with UML) has been accepted and established as a means of analyzing and designing software. In order to create the best software, the businesses in which the software systems operate must also be modeled, understood, and sometimes improved. The business model is the center for conducting business or improving how the business is operated. The evolving models also help the developers structure and focus their thinking. Working with the models increases their understanding of the business and, hopefully, also their awareness of new opportunities for improving business. Many development processes that use UML advocate that the system development should start with *use case modeling* to define the functional requirements on the system. A *use case* describes a specific usage of the system by one or more

actors. An actor is a role that a user or another system has. The objective of use case modeling is to identify and describe all the use cases that the actors require from the system. The use case descriptions then are used to analyze and design a robust system architecture that realizes the use cases (this is what is referred to as "use case driven" development). But how do you know that all of the use cases, or even the correct use cases that best support the business in which the system operates, are identified? To answer such questions you need to model and understand the system's surroundings. Modeling a business's surroundings involves answering such questions as:

- How do the different actors interact?
- What activities are part of their work?
- What are the ultimate goals of their work?
- What other people, systems, or resources are involved that do not show up as actors to this specific system?
- What rules govern their activities and structures?
- Are there ways actors could perform more efficiently?

The answers to these questions come from tackling the entire business and looking beyond the functions of the information system currently being built (and using techniques other than use case modeling). The ultimate objective of all software systems is to give correct and extensive support to the business of which it is a part. However, when modeling the surroundings of the information system, you are no longer modeling software. Enter the world of *business modeling*.

There can be many reasons for doing business modeling:

- **To better understand the key mechanisms of an existing business.** The models can be used to train people by providing a clear picture of their role and tasks in the overall organization.
- **To act as the basis for creating suitable information systems that support the business.** The descriptions of the business are used to identify necessary information system support. The models are also used as a basis for specifying the key requirements on those systems. Ideally large parts of the business model can be mapped directly onto software objects. As more and more infrastructure software systems are bought, there is a potential for the systems that are developed to become more business driven where the developers can concentrate more on functionality that supports the business rather than solving technical incompatibilities or problems.
- **To act as the basis for improving the current business structure and operation.** The models identify changes in the current business that are necessary to implement the improved business model.
- **To show the structure of an innovated business. The model becomes the basis for the action plan.** Innovation suggests that radical change, rather than incremental changes, have been made to the business

processes.

- **To experiment with a new business concept, or to copy or study a concept used by a competitive company (e.g., benchmarking on the model level).** The developed model becomes a sketch of a possible development for the business. The model can be a new idea, inspired by modeling other businesses, or taking advantage of new technologies, such as the Internet.
- **To identify outsourcing opportunities.** Parts of the business that are not considered the "core business" are delegated to outside suppliers. The models are used as the specification for the suppliers.

Business Modeling with UML

UML has quickly been adopted as the standard modeling language for modeling software systems. This article (and the book from which it is an extract) discusses how UML also can be used for business modeling and thus demonstrate that the same modeling language can be used for the business models as for the software models.

UML was defined to model the architecture of software systems. Even though there are similarities between software and business systems, there are also some differences. Business systems have many concepts that are never intended or suitable to execute in a program, such as the people working in the business, manufacturing production equipment, and rules and goals that drive the business processes. UML was initially designed to describe aspects of a software system. Because of this, UML needed to be extended in order to more clearly identify and visualize the important concepts of processes, goals, resources, and rules of a business system. To address this issue, we have created a set of extensions based on the existing model elements of UML. These extensions are called *Eriksson-Penker Business Extensions* and provide symbols for modeling the processes, resources, rules, and goals of a business system. These extensions form a basic framework for business extensions to UML (rather than a definitive set of business extensions) to which a business architect can add stereotypes or properties suitable to his or her line of business.

The standard extension mechanisms in UML that allow you to adapt UML to accommodate new concepts are:

Stereotypes. An extension of the vocabulary of the UML, which allows you to create new building blocks from existing ones but specific to your problem [Booch, 1998]. Stereotypes may have their own visual icons that replace the icon which the existing UML element uses.

Tagged values (properties). An extension of the properties of a UML element, which allows you to create new information in that element's specification [Booch,1998].

Constraints. An extension of the semantics of a UML element, allowing you to add new rules or modify existing ones [Booch, 1998].

Although different businesses have different goals and internal structures they use similar concepts to describe their structure and operation, and it is to represent these concepts the extension mechanisms in UML have been used for this purpose. The primary concepts used when defining the business system are (a complete meta-model describing these concepts and their relationships is available in the book):

- **Resources.** The objects within the business, such as people, material, information, and products, that are used or produced in the business. The resources are arranged in structures and have relationships with each other. Resources are manipulated (used, consumed, refined, or produced) through processes. Resources can be categorized into physical, abstract, and informational (each having their own stereotype).
- **Processes.** The activities performed within the business in which the state of business resources changes. Processes describe how the work is done within the business. Processes are governed by rules.
- **Goals.** The purpose of the business, or the outcome the business as a whole is trying to achieve. Goals can be broken down into sub-goals and allocated to individual parts of the business, such as processes or objects. Goals express the desired states of resources and are achieved by processes. Goals can be expressed as one or more rules.
- **Rules.** A statement that defines or constrains some aspect of the business, and represents business knowledge. It governs how the business should be run (i.e., how the processes should execute) or how resources may be structured and related to each other. Rules can be enforced on the business from the outside by regulations or laws, or they can be defined within the business to achieve the goals of the business. Business rules are defined using the Object Constraint Language (OCL) which is a part of the UML standard.

Activity Diagram

The most important UML diagram for doing business modeling is the activity diagram, now supported in Rational Rose.

Activity diagrams are often used in software development projects to document program flow, e.g., to show the algorithm used to implement a specific operation. Activity diagrams have a wide-range of uses, in that they can show activities (sequential and in parallel), the objects consumed, used, or produced by an activity, who is responsible for an activity, and the relationships and dependencies between activities. All of this is essential in business modeling.

The *business* processes are the active part of the business.

They describe the functions of the business, and involve resources that are used, transformed, or produced. A common definition of business process is (Davenport 1993):

A process is simply a structured set of activities designed to produce a specified output for a particular customer or

market. It implies a strong emphasis on how work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action.

A process could in simple terms be described as a number of activities. Eriksson-Penker Business Extensions represent a process in an UML class diagram with the process symbol shown for Process A in [Figure 1](#). In formal UML, the symbol is a *stereotyped activity* (an action state) from an activity diagram. A process, that is an activity stereotyped to process, takes input resources from its left-hand side and indicates its output resources on its right-hand side (shown as dependencies to and from the process, according to standard UML syntax). For instance, in a telesales process, the input resources could be prospects of potential customers and the output could be actual orders. The goal of the process is either illustrated as a goal object above the process symbol, or more informally specified in the tagged value goal of the process (the tagged value is not visible in the diagram but is available in a tool). A goal of a telesales process could be a specific number of orders per week. Resource objects used or involved as part of the process are shown below the process symbol. Resources that are used or needed by the process have their dependency stereotyped to "supply" and a resource that controls the process has its dependency stereotyped to "control". Resources in the telesales process could be salespersons (supplying resource) and telesales manager (controlling resource).

Business Views

A complete business model is shown in a number of views, similar to how a software system is modeled in a number of views (Kruchten 1995). Each view is expressed in one or more diagrams. The diagrams can be of different types, dependent upon the specific structure or situation in the business that it is depicting. Diagrams capture the processes, rules, goals, and objects in the business, and their relationships and interactions with each other. The Eriksson-Penker Business Extensions use four different views of a business, and they are:

- **Business Vision View.** The overall vision of the business. This view describes a goal structure for the company, and illustrates problems that must be solved in order to reach those goals.
- **Business Process View.** The business processes that represent the activities and value created in the business. This view illustrates the interaction between the processes and resources in order to achieve the goal of each process, as well as the interaction between different processes.
- **Business Structural View.** The structures among the resources in the business, such as the organization of the business or the structure of the products created.

- **Business Behavioral View.** The individual behavior of each important resource and process in the business model and how they interact with each other.

The views are not separate models; they are different perspectives on one or more specific aspect of the business. Combined, the views create a complete model of the business. Business Rules, defined in the OCL language, can be applied in all of the views.

Business Vision View

The *Business Vision View* depicts the company's goals. It is an image of where the company is headed. This view sets up the overall strategy for the business, defines the goals of the business, and acts as a guide for modeling the other views. The ultimate result of the Business Vision View is a definition of the desired future state of the company, and how that state can be reached. The primary result is expressed in a vision statement, one or more goal/problem models, and sometimes also a conceptual model. A *vision statement* is a short text document that outlines the vision of the company some years into the future. The *goal/problem model* is a UML object diagram that breaks down the major goals of the business into sub-goals, and indicates the problems that stand in the way of achieving those goals. The *conceptual model* is a UML class diagram that defines important concepts and relationships in the business to create a common set of terminology. [Figure 2](#) shows a goal-problem diagram in which a high-level goal has been broken down into more concrete goals, goals that in turn can be allocated to business processes. With the goals are shown problems which hinder the achievement of that goal, and this typically leads to the identification of further sub-goals.

Business Process View

The Business Process View is at the center of business modeling. As previously discussed, the processes show the activities to achieve an explicit goal and their relationships with the resources participating in the process. Resources include people, material, energy, information, and technology, and can be consumed, refined, created, or used (i.e., act as a catalyst) during the process. There are relationships between a process and its resources, between different processes that interact, and there is a coupling of processes to goals. A process diagram (based on a UML activity diagram) can also show how business events are generated or received between different processes, (i.e., as a means to interact or communicate between processes).

[Figure 3](#) shows a process diagram (again based on an activity diagram with stereotypes from the business extensions) where the relationships between different processes are shown. This diagram also makes use of swimlanes, which are used to show the organizational habitat of the process (it can also be used to show who is responsible for the process).

There is a variant of the process diagram, called the *assembly line diagram*, in the Eriksson-Penker Business Extensions. As

with the process diagram, it is based to a large extent on the UML activity diagram. The assembly line diagram (introduced in the Astrakan method) has been used successfully for process modeling, particularly when the purpose of modeling is the production of information systems that support the processes. The diagram shows how the processes write or read different objects placed in a package (called the "assembly line" because it looks like an assembly line below the process diagram). The process communicates with objects in the package and requests different services, or reads or writes information from it. The assembly line package typically represents an entire information system or a subsystem in a information system. The diagram shows the requirements that the process has on the information system through the interface it has to it.

This interface is normally described through use cases in object-oriented modeling, and a set of references in an assembly line diagram typically become a use case that the information system has to provide. This is very important, because it maps the business process to use cases that describe the functional requirements on an information system, and it also identifies the proper actors to the use cases (the roles played by the resources in the process that uses the assembly lines). Assembly line diagrams provide the connection between business modeling and software system requirements modeling with use cases. A common question when modeling use cases on a software system is "How do I know I have defined the right use cases in terms of the business?" The assembly line diagram, which is fully demonstrated and elaborated upon in the book, is a good technique for this.

Business Structural View

The *Business Structural View* ([Figure 4](#)) shows the structures of the resources, the products, or the services, and the information in the business, including the traditional organization of the company (divisions, departments, sections, business units, etc.). Traditional organizational charts and descriptions, and descriptions of the products and services the company provides are the basis for the Business Structural View. The information from the Process View is also used since it shows what resources are used. Note that typically these two views are modeled in parallel, since they contribute to each other and must be consistent. The view shows the structures of resources, of information and of the business organization.

Business Behavioral View

The *Business Behavioral View* illustrates both the individual behaviors of resources and processes in the business as well as the interaction between several different resources and processes. The behavior of the resource objects is governed by the Business Process View, which shows the overall main control flow of the work performed. However, the Business Behavioral View looks into each of the involved objects in

more detail: their state, their behavior in each state, and possible state transitions. The Behavioral View also shows the interaction between different processes, such as how they are synchronized with each other. By doing so, the Behavioral View is an important tool to use when allocating the exact responsibility for different activities, and when defining the exact behavior of each resource that takes part in each process.

The Behavioral view makes use of state chart diagrams, sequence diagrams and collaboration diagrams. Process diagrams can also be used to show interaction between processes.

From Business Model to Software

A common use of business models is to use them as the basis for identifying functional and non-functional requirements on one or more information systems. As previously discussed, business modeling is the way to actually know if the use cases defined for a system are the "right" or "optimal" requirements on that system. There are also other uses of the business model, because many of the objects and relationships found in the business model will also be objects and relationships in the information system model. It is important to realize that it is not a one-to-one mapping though, a critical analysis must be made of the business model to see what is applicable for a specific information system.

[Figure 5](#) shows an overview of the relationships between the business modeling process and the information system development process.

The ultimate goal is of course to create the software system(s) that best supports and fits into the business. The business model is used in software modeling to:

- **Identify the information systems that best support the operation of the business.** The systems can be new systems, standard systems, or legacy systems.
- **Find functional requirements.** The business model is used as a basis to identify the correct set of functions or use cases that the system should supply to the business processes.
- **Find non-functional requirements.** These requirements, such as robustness, security, availability, and performance, typically span and involve the entire system. They are often generic and not attached to a specific use case.
- **Act as a basis for analysis and design of the system.** For example, information about resources in the business model can be used to identify classes in the system. However, it is not possible to directly transfer the classes in the business model to the software model.
- **Identify suitable components.** Modern software development makes use of components: autonomous packages of functionality that are not specific to a

certain system but can be used in several systems. Most of component technology has concentrated on technical components, but there has been an increasing interest in defining business components that encapsulate a specific and reusable area of business functionality. Business models are a good way to identify areas of functionality and to define the appropriate set of services.

References:

1. Eriksson, Hans-Erik and Penker, Magnus: "Business Modeling with UML: Business Patterns at work", Wiley & Sons, Fall 1999.
2. Eriksson, Hans-Erik and Penker, Magnus: "UML Toolkit", Wiley & Sons 1998.
3. Booch, Grady, Jacobson, Ivar and Rumbaugh, James: "The Unified Modeling Language Users Guide", Addison Wesley 1998.
4. Kruchten, Philippe: "The 4+1 View Model of Architecture", IEEE Software, IEEE, November 1995.
5. Davenport, Thomas: "Process Innovation: Reengineering work through Information Technology", Harvard Business School 1992.
6. Harrington, James: "Business Process Improvement", McGraw-Hill 1991
7. Marca, David and McGowan, Clement: "IDEF0/SADT: Business Process and Enterprise Modeling", Eclectic Solutions 1993.

About the Authors

Hans-Erik Eriksson (hanserik.eriksson@opentraining.com) is Founder and pfChairman of Open Training - a company specialized in advanced online learning solutions and e-Training. He has over 10 years experience in object-oriented technology.

Magnus Penker (magnus.penker@opentraining.com) is CEO of Open Training, and a former senior management consultant specializing in process modeling and object-oriented modeling. Hans-Erik and Magnus are the authors of UML Toolkit, one of the early books on UML and in October their new book, on which this article is based, Business Modeling with UML: Business Patterns at Work (Wiley), will be released. More information about the book can be found at <http://www.opentraining.com>.

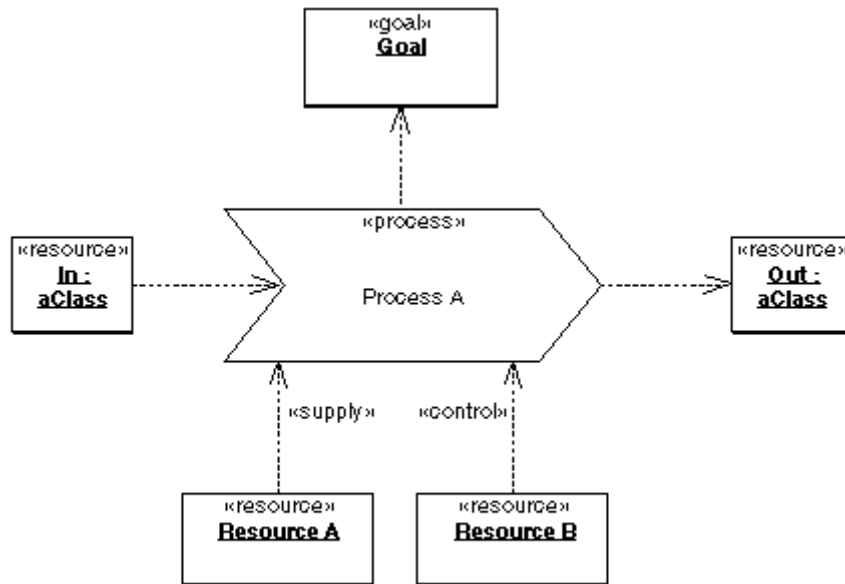


Figure 1: Activity diagram with stereotypes for a process and for objects involved in the process.

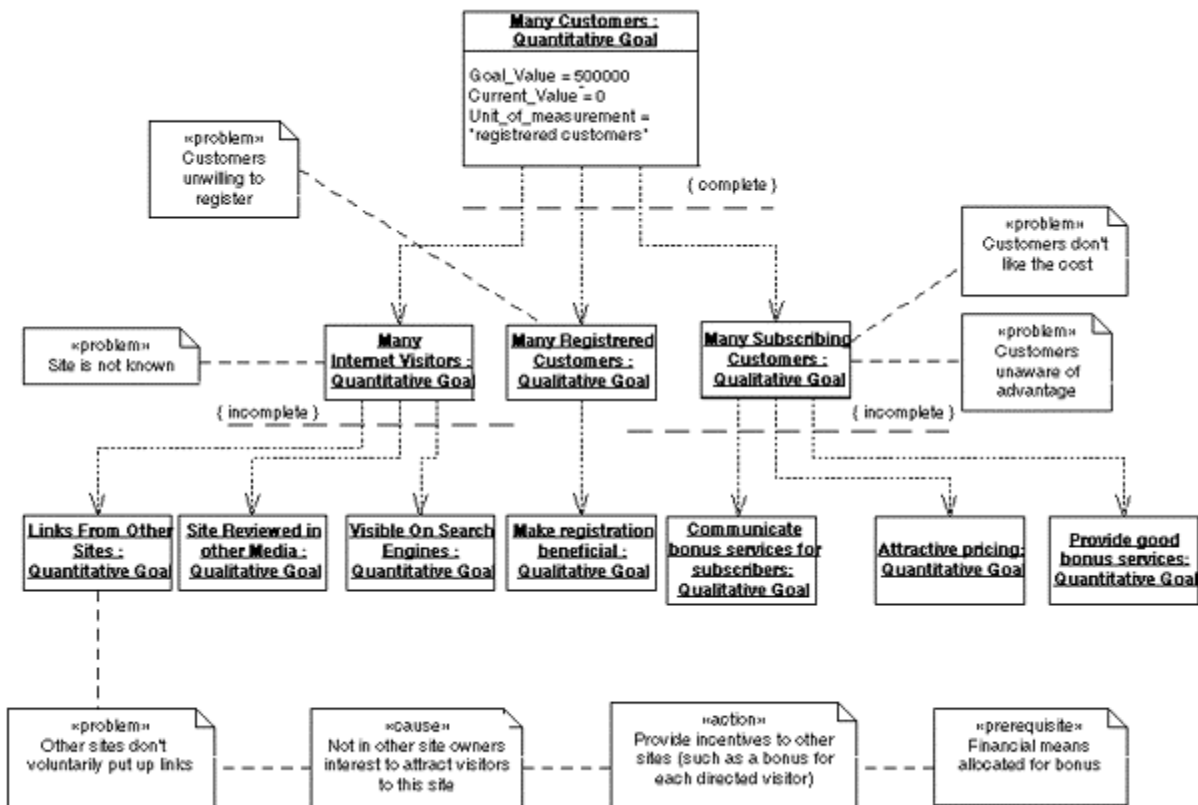


Figure 2: A goal-problem diagram based on a UML object diagram.

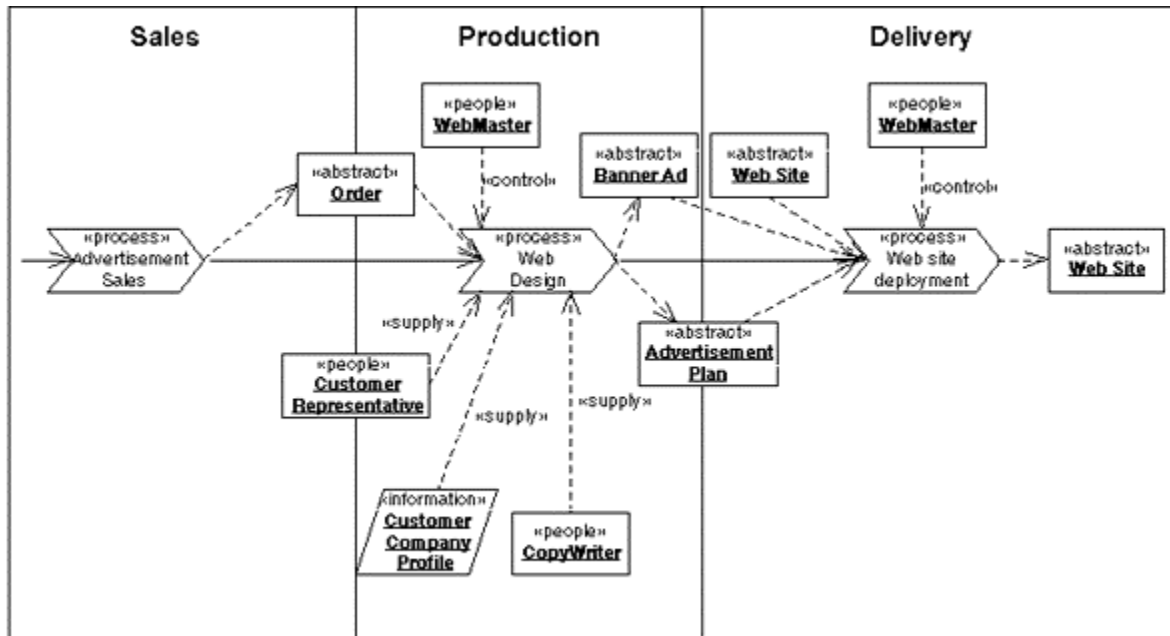


Figure 3: A process diagram based on an activity diagram with swimlanes.

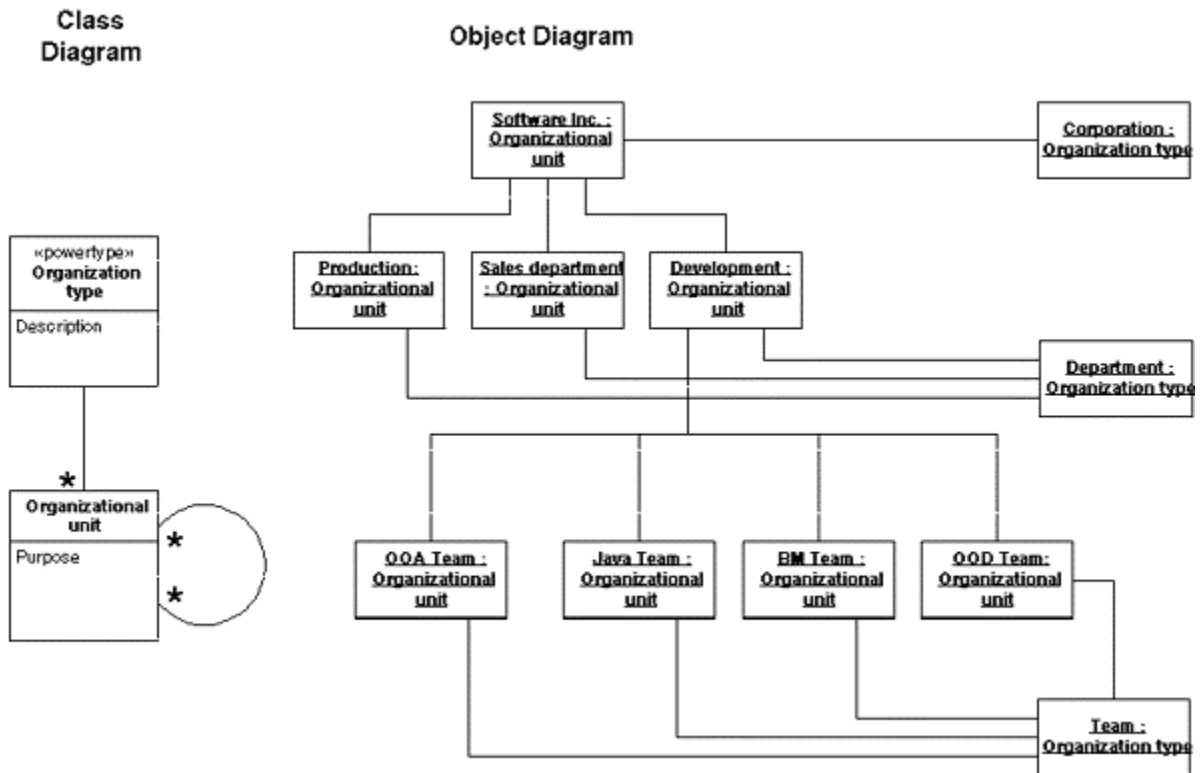


Figure 4: An organization diagram based on UML class and object diagrams. Note the use of the UML construct power type, in which objects of a class are used to represent types.

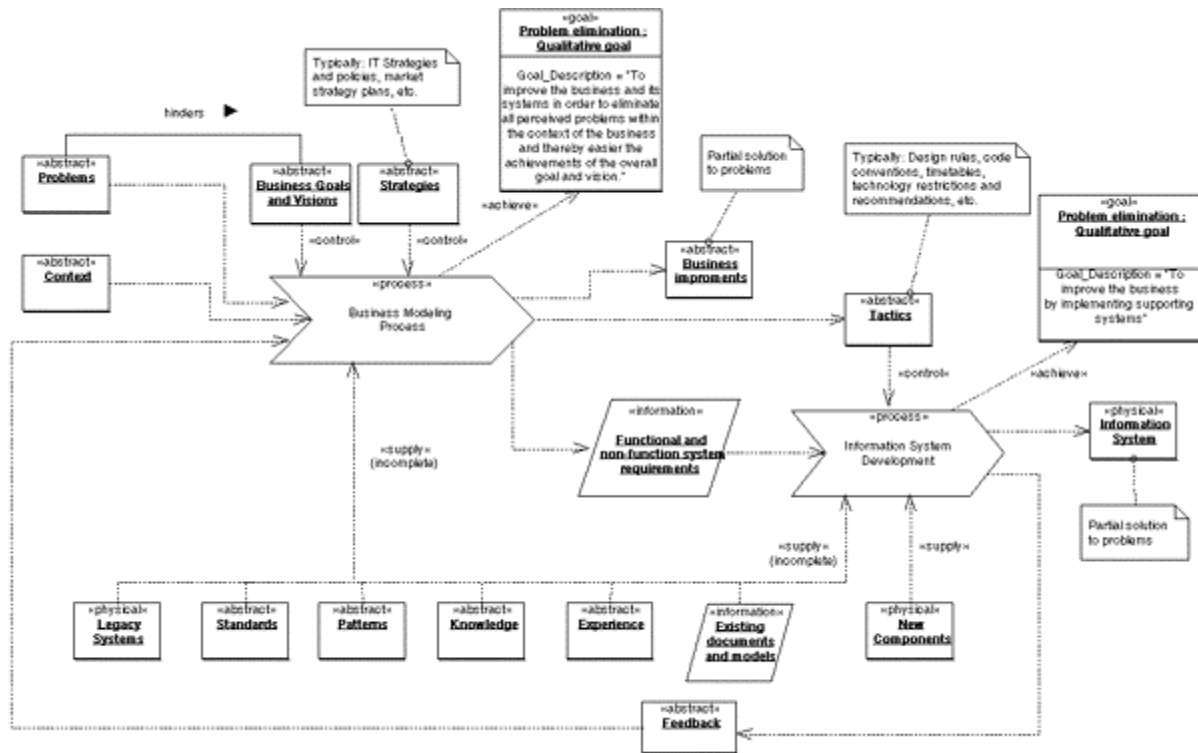


Figure 5: A process diagram showing a simplified view of the relationships between business modeling and information system development.