Business Objects

Introduction

Business engineering projects have been driving the development of business applications for the past 20 years. Originally, this was known as computer-aided software engineering (CASE). In the meantime, however, the CASE market has become more segmented. The key to useability today is not the CASE tool but the merging of models, methodologies and applications combined with CASE tools. As part of an ongoing effort to apply leading technologies to new development, SAP has begun work in the area of object technology.

Business design is the challenge for most companies because of the complexity of their organizations. Add the implementation of a new software system to the equation, and the task becomes even more difficult because highly integrated applications tend to run completely through a business. Without having a very clear understanding of how these applications execute a process, it's harder to adapt the software to the business. However, with "pre-fabricated models" of the applications, companies can reduce their risks while at the same time take advantage of a benchmark for business process redesign.

Business process engineering is not limited to describing processes, however. In a business context, a model or business process diagram illustrates not only the tasks and organizational structure of a company but also how things are done. A company's information model can also include the description of other aspects of a company as well, including data, function, organization, information flow and communication flow. Or it can be used to emphasize a company's individual requirements, creating a model of a company's ideal "to-be" or goal situation based on ist present "as-is" situation.

A large part of the history of business modeling is the history of software engineering and the trials and tribulations of describing data-hosted systems that were, for a large part, the bread and butter of commercial data processing. Especially in the late 70s and early 80s, data modeling became a primary task for data processing departments. "Enterprise modeling," as it was later called, was an attempt to create a complete blueprint of data and process structures within a company, primarily with the aim of creating applications. But often due to the lack of sophisticated modeling techniques, creating a database blueprint from scratch was a disaster. In fact, many of these projects really failed to deliver any added value to reengineering or even to application development, a lesson IT management bitterly learned.

The evolution of process modeling was the result of a trend toward increasing the productivity of software development projects and the general trend toward quality, total quality management, BPR and lean management movements, where management philosophy changed from managing all parts of a particular business transaction to actually optimizing the transaction itself. A fundamental shift took place. This shift could only have happened after companies had a good understanding of what data they were dealing with. These two elements tend to fit together. As soon as one has processes, which implies also a functional description of the business, it automatically becomes the task to describe the organization and simulate and consider model changes to the organization that could affect that particular process.

Continuous engineering, as it's being called today, is the second-generation of BPR. It involves not only revamping or changing an organization or reengineering an organization but doing it over a particular life-cycle or timeframe. This is a very important issue for enterprise applications. A lot of the BPR dollars spent until now were used to implement tools and systems. Without a continuous engineering concept in place, many companies ended up having to throw away a lot of the process design information that they had collected. SAP's vision is to aid today's continuous engineering efforts with a long-lasting platform to manage that change.

As the next generation development in client/server computing, object, or component-based, technology plays a role in continous business engineering. Applications built upon object technology take advantage of a flexible, multi-tier design spread across a heterogeneous computing platform. In a business environment, companies are exploiting distributed information technology based on objects or components not only to improve service or quickly adapt to changing markets, but also to "reuse" the functionality across both computing platforms and applications and to see what data they are dealing with in normal business terms.

Because client/server environments are heterogeneous, objects must be able to co-exist in a loosely coupled, multi-platform and multi-standard environment. High-level middleware such as object request brokers is particularly critical for the use of objects and communication between them. Emerging object integration technologies such as CORBA for distributed object infrastructures and Object Linking and Embedding (OLE), developed by

Microsoft, for encapsulating and reusing objects play a role in the development of SAP's object-oriented client/server system. Such technologies offer a means of interfacing processes, functions and business objects.

OLE is a mechanism that allows applications written by different companies in different programming languages to interoperate more effectively. Users of OLE applications create and manage compound documents containing data, or objects, in many different formats. Objects can include sound clips, spreadsheets, text, bitmaps, or anything that can be displayed or controlled by an application. The services of the different applications are integrated so that the users feel as if a single application is being used.

Business objects are the topic of discussion among the Object Management Group (OMG), IBM, Microsoft , Texas Instruments and others in the industry. Although standards have been developed, until now, however, no one has come up with business objects that can be taken into a company and used. In SAP's R/3 Release 3.0, 180 business objects are implemented in R/3 and are related to each other in object models. SAP is delivering commercial business objects -- which are business-oriented abstractions of standard software solutions -- that relate to a running application.

This is possible because of SAP's development in the areas of open technology and object technology. Release 3.0's Business Engineering Workbench (BEW) includes a complete reference model of the business functionality of R/3 provided in media format and with viewing technology.

SAP Business Object Strategy

Object technology by itself does little to solve application development problems. In the context of SAP, object technology has to be applied productively with other tools and technologies needed for development and implementation. An important trend today is how objects - especially business objects - are contained, brokered, managed and exchanged in a business.

A business object in the context of R/3 refers to everyday business elements like customer orders. SAP's client/server application, R/3, is based on a three-tier architecture. This architectural design uses intercommunicating software objects. Objects in the upper tier, often known as the application tier, use and reuse the business objects in the middle level that define business rules and protect database integrity.

Business objects also relate to the Business Engineering Workbench, the suite of technology and tools that make up SAP's implementation, modeling and design solutions. The goal of the Business Engineering Workbench is to bring together and integrate existing SAP modeling and customizing tools.

Companies often talk about changing their organizations as new infrastructures for business computing become available. This is often referred to as business process reengineering, process design, change management or continuous engineering. SAP's BEW addresses this range of business engineering activities with an integrated toolset that accesses the R/3 Repository and offers different views of the descriptive application "meta" data.

The BEW includes:

- intelligent graphical browsers
- navigation from one view of the system to another
- access to reference models
- reports and specialized information about models
- execution linkage with the application
- linkage between the models and customizing

The BEW is a tool for understanding how business objects can be used to design a new organization over time. It is not, however, an upper CASE tool, which would generate new programs and support application development. The BEW is somewhere in the middle, offering business process design analysis and support.

SAP's business objects have two main characteristics: they are linked to other objects in the system, and they are used to perform business transactions. In SAP's R/3, objects provide the clear interface between applications and workflow management. An attribute describes the features of objects. They can be things like name, date of creation or change, status, etc. Attributes provide the link to data fields and foreign keys.

Methods are operations that transform objects into a new object. The definition entails: an event parameter, parameter (export, import) and exceptions. Methods provide the link to transactions, function modules, routines and reports.

SAP's business objects, such as customer orders, project structures or production orders, don't just store data. They also control and coordinate the execution and flow of subprocesses, invoke additional processes - like orders, manufacture, warehouse movements, etc. - and collect and coordinate feedback messages. "Business Workflow" embraces such an application-driven control system, enhancing it with application-independent process objects and services.

In particular, SAP objects include events. Events show that something has happened to the object. With events, the business object becomes the central part of the control-flow or workflow used to interface with other event-oriented applications, providing information flow from the application to workflow management.

Industry Trends

In 1989, eight companies founded an industry consortium dedicated to the development of object-oriented software. Founding members included 3Com Corporation, American Airlines, Canon, Data General, Hewlett-Packard, Philips Telecommunications, Sun Microsystems and Unisys. Today, the Object Management Group (OMG) has more than 500 international members, including SAP AG and many of the world's top software developers. In fact, most of the major vendors and users of computer systems today use or plan to use object-oriented technology.

The idea behind OMG is to foster the growth and development of object technology. The group has established industry guidelines that make it possible to develop applications for all major hardware platforms and operating systems. According to the OMG definition, object technology is software development that models the real world with representational 'objects.' Made up of software components and methods, objects are encapsulated to hide their complexity from users. The overriding benefit of an object-oriented IT approach is the ability to increase functionality by adding new objects to or extending existing ones in the system.

Object in Business Engineering Workbench

The Repository, which is now inside R/3, is the bridge between business engineering (the Business Engineering Workbench) and application development (the ABAP/4 Development Workbench). Process models, function models, data models, business objects, object models and related data and connections are all stored in the Repository. On the technical side, the Repository also stores information like data definitions, screens and program objects used for developing or extending R/3.

The SAP business object structure differs from other views of business objects. One difference is that between object attributes and normal data fields. Object attributes show features of business objects that can be used outside the SAP system, e.g. in OLE applications or third-party workflow systems. The Repository's API (application programming interface) allows other software vendors to hook up to the Repository, take information out and downstream it into their own toolset or application.