



Zakaria Maamar and Jeff Sutherland

Toward Intelligent Business Objects

Focusing on techniques to enhance BOs that exhibit goal-oriented behavior.

With the increasing development of information and communication technologies, users are becoming more and more demanding on companies, requiring new types of specialized tools and advanced services. To cope with these growing and complex requirements, system designers and developers need new, advanced approaches to support their work. Such approaches range from design methods to programming languages. As an emerging technology, Business Objects (BOs) seem to be a good candidate and a promising approach [3]. BOs should be able to provide an insight into what aspects of a business should be delegated, how these aspects may evolve, and what will be the effect of specific changes.

Currently, BOs seem to meet the requirements of the organizational field. However, BOs could be “leveraged” in order to be applied to other complex fields, such as cooperative manufacturing systems. Leveraging BOs with techniques borrowed from advanced research domains, for instance Distributed Artificial Intelligence (DAI), is an avenue

needing investigation. Therefore, the purpose of this “Technical Opinion” is to outline how BOs could be enhanced with different DAI techniques. Using these techniques, we aim at having BOs that exhibit goal-oriented behavior, instead of just responding to external stimuli.

ACCORDING TO THE OMG’S Business Object Management Special Interest Group, a BO is “the representation of a thing active in the business domain, including at least its business name and definition, attributes, behavior, relationships, and constraints.” Through BOs, managers and users can understand each other by using familiar concepts and creating a common model for interactions. Moreover, the Business Object Management Architecture (BOMA) defines a BO through the concepts of purpose, process, resource, and organization [1]. A purpose defines why a business exists. A process defines how this purpose is reached through a series of dependent activities. Each process requires resources for its fulfillment.

Finally, resources are managed by organizations.

Thanks to the network technology, distributed systems can exchange information and services, and hence can collaborate to achieve common operations. To enable such systems to perform coherently and efficiently, DAI techniques (such as planning and negotiation) can be used. DAI aims at studying different issues related to the distribution and coordination of knowledge and actions in environments involving multiple entities, called agents [4]. Agents can be integrated into frameworks, called Multi-Agent Systems (MASs), that can be spread across networks. Communities of agents (designed at a macro level) are much more powerful than any individual agent (designed at a micro level).

When multiple BOs have to evolve in the same environment, it is important to define their behaviors in order to regulate their interactions and avoid conflicts, for instance on resources. However, behaviors of BOs may emerge as a result of their interac-

tions with each other and/or with the external environment. Therefore, the following questions need to be answered: How can collaborative BOs be designed? What types of information could these BOs exchange? What types of communication middlewares could support these exchanges? And, how could these BOs be associated with common goals?

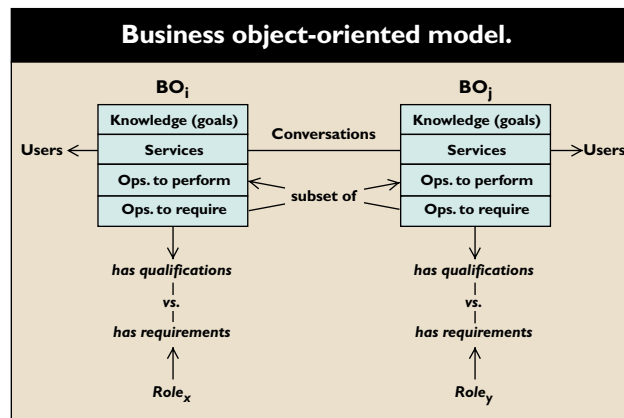
Techniques borrowed from DAI and applied to BOs might provide answers to these questions. For instance, collaboration between BOs could rely on different coordination mechanisms such as resource scheduling, redundancy avoidance, and others.

A BO is an encapsulation of data and methods. Methods specify the behavior of this BO

when it performs the services assigned by its designer. This assumes the designer is aware of all the situations in which the BO will be involved. As a result of its static behavior, this BO does not have the ability to adapt its operations according to its current states and the characteristics of its environment. Interesting are the situations where a BO would be able to decide with whom to collaborate, what services to offer, what services to request, and what visible “parts” in terms of behaviors (from private to public, and vice versa) to exhibit.

BOs could be considered as reactive agents. BOs respond to the messages they receive from other BOs or from the external environment. More interesting are the situations requiring the adaptability of BO behavior. Such BOs

could be considered as cognitive agents. For instance, once the service of a BO is requested, this BO designs the plans that carry out this service. A plan, viewed as a BOMA process, could be obtained from a goal decomposition approach. Furthermore, plan design could take into account different aspects. For example, who has requested the service?



What are the operations that carry out the service? What happens if an operation fails? And, what kind of resources do these operations require? The goal decomposition approach consists of identifying a service with a global goal, viewed as a BOMA purpose, and decomposing this goal into a set of subgoals, until each subgoal is associated with a plan of tasks. A plan is instantiated when a triggering event—that satisfies this plan’s conditions—occurs. Furthermore, each task produces outputs that will be used as inputs by other tasks. Therefore, tasks as well as plans have to be executed in an ordered way.

When a BO collaborates with other BOs, it initially has to identify and locate these BOs. Instead of knowing all the BOs’ character-

istics, a specialized BO, called a Broker, could be used. The Broker could let BOs discover each other based on the services they provide and look for. BOs advertise their capabilities at the Broker level. When a BO looks for appropriate BOs, because these BOs manage the required resources, this BO sends a request to the Broker. According to the needs of this BO, the Broker identifies the specific BOs. Once these BOs are identified, the BO interacts with them either directly or through the Broker.

In this situation, we assumed BOs collaborate without taking into account their current states, for instance, their workloads. More appropriate would be if these BOs consider their states

before committing themselves to other BOs. Therefore, BOs have to coordinate their activities with other BOs before reaching agreements. In DAI, the Contract-Net, a well-known coordination strategy that relies on the manager-contractor pattern, is a well-known coordination strategy [4]. For example, a BO, acting as a manager, advertises a contract containing its needs in terms of BOs’ services. BOs receive and evaluate the advertisement and those with appropriate resources and workloads, and capabilities reply to the BO with bids indicating their ability to perform the advertised contract. Finally, this BO evaluates the bids it has received and awards the contract to the BO, called a contractor, that returned the winning bid.

BOs could interact with other

BOs, whether being the same type or not. However, to interact efficiently, BOs have to understand and communicate with each other. This raises the need for a Business-Object Communication Language (BOCL). A BOCL could be viewed as Knowledge and Query Manipulation Knowledge for agents. Furthermore, the BOCL could be a subset of the Component Definition Language (CDL) of the Business-Object Component Architecture (BOCA). Using the BOCL, BOs interact to accomplish individual as well as social goals. In fact, the possible interactions that could take place between BOs represent conversations. A conversation describes the chronology of messages passed between BOs.

The figure illustrates how an intelligent BO could be modelled. Such a BO is a multilayer architecture. The first layer contains the BO's knowledge in terms of goals to reach. The second layer contains the services that are offered to users and carried out by this BO. Each service is associated with a goal of the BO's knowledge. The third layer corresponds to the operations used to perform user services. Such operations could be "packaged" into workflows. Finally, the fourth layer contains a reference to the operations belonging to other BOs. These operations are required to fulfill user services. In an organization, an intelligent BO could fill one or several roles. In fact, this BO has the qualifica-

tions that meet the requirements of these roles. ■

ZAKARIA MAAMAR (maamarz@exite.com) is an associate professor at the College of Information Systems, Zayad University, Dubai, U.A.E.

JEFF SUTHERLAND (jeff.sutherland@computer.org) is a chief technology officer at Virtual Medical Systems, Cambridge, MA.

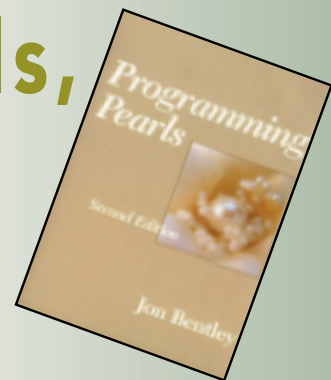
REFERENCES

1. Business Object Management Group; www.sesh.com/Guide/BusinessObjects.html.
2. Davis, R. and Smith, R.G. Negotiation as a metaphor for distributed problem solving. *AI 20* (1983), 63–109.
3. Eeles, P. and Sims, O. *Building Business Objects*. Wiley, 1998.
4. Jennings, N. Sycara, K., and Wooldridge, M. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems 1*, 1 (1998), 7–38.

© 2000 ACM 0002-0782/00/1000 \$5.00

Programming Pearls, Second Edition

by: Jon L. Bentley



Order #: 702003 Paperback 2000 256 pp.

ISBN: 0-201-65788-0

Members: \$22.45 Non Members: \$ 24.95

To order online: <http://www.acm.org/catalog/>

Call Toll Free:

1.800.342.6626 (USA & Canada)

1.212.626.0500 (outside US)

Fax: 1.212.944.1318



Association for Computing Machinery
The First Society in Computing