

Chapitre d'actes

1999

Published version

Public access

This is the published version of the publication, made available in accordance with the publisher's policy.

Active Business Objects (ABOs): A novel paradigm for building (and using!) Business Information System

Konstantas, Dimitri; Morin, Jean-Henry; Barziv, Orly; Koumpis, Adamantios; Kobel, Christian

How to cite

KONSTANTAS, Dimitri et al. Active Business Objects (ABOs): A novel paradigm for building (and using!) Business Information System. In: Proceedings of the First International Workshop on Enterprise Management and Resource Planning Systems: Methods, Tools and Architectures - EMRPS'99. Eder, J. & Maiden, N. & Missikoff, M. (Ed.). Venice (Italy). Rome : IASI-CNR, 1999. p. 139–147.

This publication URL: <https://archive-ouverte.unige.ch/unige:72577>

© This document is protected by copyright. Please refer to copyright holder(s) for terms of use.

Last update in Archive ouverte UNIGE on 15.03.2023 00:17

Active Business Objects (ABOs): A novel paradigm for building (and using!) Business Information System¹

Dimitri Konstantas
Jean-Henry Morin
Orly Barziv
Adamantios Koumpis
Christian Kobel

Abstract

The ABO project aims in the design and development an agent platform where mobile agents represent business processes. These agents encapsulate the policies, business practices and models of different business activities (like order management, payroll, warehouse and stock control). The aim of ABO is to provide a new paradigm of Business Information System design and use.

1 Introduction – background of the idea

The objective of the ABO project is to support enterprises capitalise on the benefits of the information society, by enabling the adoption of a new paradigm of Business Information Systems (BIS). Instead of reactively responding to needs for information and data exchange amongst intra- and inter-enterprise structures (e.g. networks of contracting companies, company departments, etc.), ABO will provide an active infrastructure based on agent technology that will realise an open business environment integrating components from various specialised IT vendor solutions.

Specifically, in ABO we aim to develop and validate a novel approach in building and integrating BIS from software components based on a scalable modeling approach that supports the exchange of business data and information encapsulated in agents (i.e. Business objects). These agents include “methods” that allow them to act and interact in an independent way according to the needs and the operational requirements of the activity in which they participate.

Based on experiences coming both from research activities of the OSG group at the University of Geneva [5][6][7][8][10], and Unisoft's exposure to the real market needs and potential for technology utilisation, ABO focuses on providing essential technology and infrastructure for the development, deployment, operation and evolution of business systems, in which policies,

1. This project is supported by the Swiss Federal Government with the SPP-ICS project INVENT (project No. 5003-057751) and by the Commission of the European Union and the Swiss Federal government with the 5th Framework project Dilemma (For the EU project No.IST-10092 - for Switzerland OFES project No. 99.0259)

business practices and models are embedded in agents representing intra- and inter-enterprise processes and activities.

At the technical level, the goal of ABO is to develop an agent-based support framework upon which different business operation models and environments can be implemented and integrated, like for example, workflow and information flow, process-activity models and ABC/ABM (Activity Base Costing/Activity Based Management). The basic building block of ABO is the Mobile Agent: a self contained reusable active entity encapsulating data and code and implementing specific services of the BIS application.

Mobile agents [1][3][9] (on which we base our Active Business Objects concept) is now a recognised programming method for distributed working environments (Intranet/Extranet/Internet). By allowing for the migration of computation between hosts and platforms, networked services can be made more flexible and cost effective. In ABO we design and implement a framework augmenting existing agent deployment platforms with tools for implementing, monitoring and managing business processes (i.e. processes for work- and information-flow) for an extended enterprise environment. This is an environment made up of departments or companies distributed over the network. The support framework enables companies / departments to join and leave the business flow, offer services for a work- or information-flow process, and have these services accounted for their resource utilisation.

The main idea is to design and develop an agent platform where mobile agents represent business processes. These agents encapsulate the policies, business practices and models of different business activities like for example order management, payroll, warehouse and stock control. By combining different agents offering diverse services we will be able to compose complex applications for monitoring and managing in real time an extended enterprise. According to this scenario, integration of value-added component, in the form of agents, from different SMEs will be seamlessly achieved.

To give a solid example of the ABO vision one can compare it to the wide spread and evolution of PCs. Today a PC is composed from different hardware components developed by different companies. Each company excels in the development of a specific hardware component, like for example a CPU, a motherboard, memory, special purpose cards etc. As a result a wide range of solutions is available on the market and users can easily compose PCs to fit their specific needs and preferences. At the same time SMEs can bring to the market highly specialised hardware products addressing niche markets and more harmonised life cycles.

Today business systems, like SAP R3, provide integrated solutions for companies. Any new extensions/adaptations of the ISs must be either implemented by the IS designer or supported by (narrow scope) access to IS functionality by means, for example, of ODBC or native drivers. In this way there is little space for SMEs to provide highly customised value added components reflecting the particularities of the specific end user when these do not conform to the base IS. ABO's vision is to provide a technology where SMEs will be able to provide self contained software business components in the form of agents that will be seamlessly integrated to create an IS tailored to the needs of the specific end-user, as this is today possible for PC hardware.

The ABO technology is compatible with existing IS solutions and can be introduced in a company in an incremental way. ABO software gateways can be easily designed and implemented allowing the communication of existing ISs with the ABO platform. This will be done by designing proxy agents that will represent the existing IS in the ABO platform and allow intercommunication. As a result the existing ISs can be slowly replaced or augmented with the integration of new specialised composite ABOs constructed by different SME business software vendors.

2 Design Issues and Goals

The design issues under investigation for ABO are divided into two general areas. The first deals with the necessary extensions to the mobile agent infrastructure while the second is concerned with looking at how agents (active objects) can help structure the existing process backbone of business IS.

2.1 Developing a Mobile Agent Infrastructure

To make agent technology a more general support infrastructure for the virtual enterprise, the following design issues are investigated:

- **Programming models:** An agent sent out onto the net within a virtual process must be able to react to a number of events, ranging from network failures, broken pointer links to information and security attacks, to time-outs for the delivery of their content and spawning and controlling “children” agents for handling tasks in parallel. It is not yet clear what might be the most convenient model of programming agents to handle such events.
- **Backbone Integration:** A real electronic commerce backbone needs to marry the benefits of agent technology with standard middleware technology. Integration of the ABO agent platform with a middleware architecture such as CORBA or DCOM is an important goal of ABO [8].
- **Resource control:** Agents must be made accountable for the resources that they consume at a location during a process. To do this, the identity of the enterprise that created the agent and which should be charged for a given resource consumption must be easily established. Efficient mechanisms for achieving this will be designed, based on existing standards and technologies.

2.2 Agents and Process Backbone

The second issue that needs to be addressed for the ABO platform design is how the agents can be exploited to alternatively structure the business flow process and reservation protocols. For instance, agents can be used to remotely configure resources at a site; this is useful for improving the monitoring aspect of work- or information-flow processes.

In particular, agent support will be implemented fulfilling the following requirements:

- **Security:** agents allow computations to be moved. From a security viewpoint, this can be used to move process computations to safe areas to avoid certain attacks [6][11].
- **Off-line operation:** to reduce the cost of using the network, it is important to identify those services in a work-flow process that can be down-loaded to a client site in the form of agents.
- **Reliability:** bad network connections can also be overcome by computation mobility.

The integrated outcome of the ABO project is a coherent software framework on top of which software components for networked enterprises can be developed and seamlessly integrated and deployed in small, medium and large enterprises.

3 From Cobol to ABO, or How ABO will save the world!

ABO may form the basis for a significant turn in the technologies used by suppliers of business software systems all over Europe. In this respect, a repositioning of the business software industry is enabled, by allowing investments in *technology development*, that may provide SME IT companies with a higher volume of returns on investments, than those expected in the cases of *applications and systems development*. Moreover, ABO will be a major technological step for the business software sector by addressing central issues which are directly related to the imminent need for repositioning of investments carried out by the business software industry towards *novel technology development*.

In the project, we address issues related to *component-based software engineering*, as we adopt a component-based approach, focusing on component integration and the concept of evolutionary re-configuration. Active Business Objects as developed within the project may transform the organisational and functional levels of an enterprise, thus enabling the smooth transition to more efficient and cost-effective ways of structuring work both at the intra- and at the inter-enterprise level. Active Business Objects coming from different sources (i.e. ERP systems of different enterprises) may thus form the “cornerstones” of new, complex systems and services, thus taking advantage of the “system families” concepts and stimulating both real-life practice improvement and the take-up of the associated software technologies.

During the last 20 years Information System (IS) design and implementation has evolved from the monolithic flat programming (e.g. Cobol based), to modular programming (e.g. C), to object-oriented programming (e.g. C++) and finally today in agent based programming (e.g. Java). Each evolutionary step improved the ability of component reuse in the design and implementation of applications and introduced a new paradigm for application and component inter-operation.

With monolithic programming the reuse of components for the implementation of applications was near zero and the application interoperation was performed with raw unstructured data.

Modular programming allowed the design of applications based on well defined module interfaces which were able to be reused by different applications and introduced the notion of interfaces (API and RPC) and client-server model for the interoperation of applications.

Object-oriented programming increased component reusability introducing the notion of objects encapsulating data and behaviour, and proving application interoperability via the exchange of messages. Object libraries provided the required mechanisms for the implementation of applications.

A further evolutionary step was the introduction of middleware (e.g. CORBA) that allowed the composition of applications based on the notion of services. Complex applications were designed by decomposition in required services which were to be found on the network.

The basic characteristic of the above application design and implementation models is that they are based on the client-server model where the client sends data to the server who processes them and returns them to the client.

During the last few years a new model for the design and implementation of applications has appeared: the agent based application programming. Agents allow application interoperability with the exchange of code, that is agents, that encapsulate a specific behaviour. With agent programming instead of sending the data to the location of the behaviour, we import the behaviour at the location of the data.

Programming model	Component Interoperation mechanism	Application Composition Paradigm
Flat (Cobol)	Raw data	-
Modular programming (C)	Structured data	API
Object Orient programming (C++)	Message passing	Behaviour
Middleware (CORBA)	Message passing	Service
Agent programming (Java)	Code passing	Behaviour
ABO	Agent Encapsulated information	Service (active object)

Figure 1 Evolution of programming paradigms

However without realizing it, the agent-programming model has taken us one step back in the design and implementation of applications, when compared with the client-server evolution model approaches. That is, whereas middleware was an evolutionary step from OO programming allowing us to implement applications in terms of *high level services* instead of behaviour, agent based programming has brought us back to application design based on behaviour.

The first contribution of ABO is providing a framework that will allow application design and implementation based on the notion of *services' composition*, facilitating component integration not in terms of API and network middleware standardization, but with open-ended mobile agents. The ABO framework platform that will enable customized tailored composition of

IS for large- medium- small companies according to their niche business operations. ABO based applications will interoperate using active objects (agents) which encapsulate services implementing policies and practices of applications.

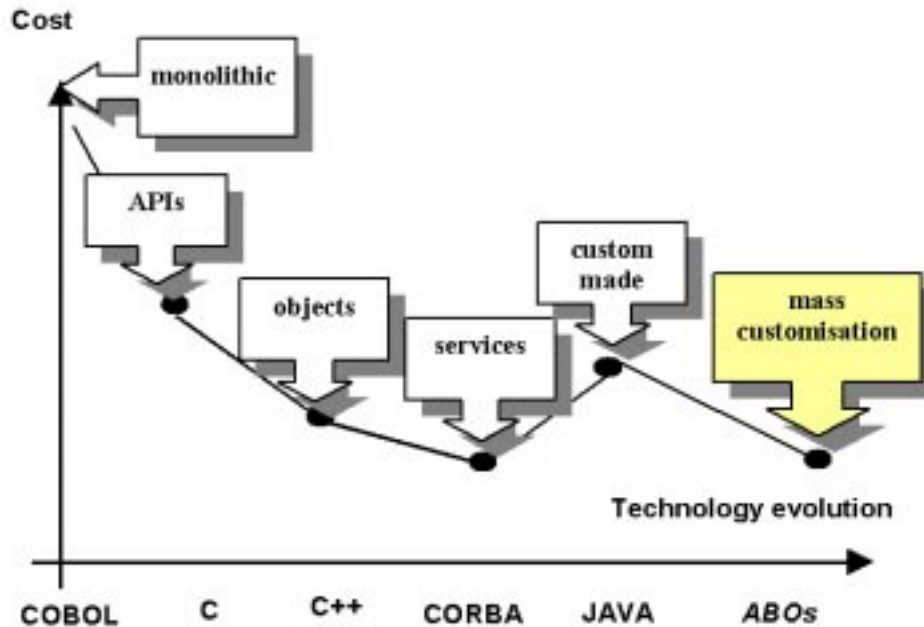


Figure 2 Evolution over time of the software system production cost for different programming paradigms

Based on this platform the ABO project will design and implement verifiable concept proving innovative business applications introducing agent technology for the management and control of enterprise processes and activities. ABO will promote the employment of agents in business information systems by means of using a consistent and integrated approach that enables policies, business practices and models to be embedded in intra- and inter-enterprise processes and activities.

The benefits offered by the ABO approach in the design and implementation of business information systems are many. First of all the cost of design and implementation of BIS will be drastically reduced. The BIS designer will be able to seamlessly reuse existing interoperable applications and components and directly integrate services available on the ABO platform. A second benefit will come from the increased modularity of the developed BIS, which will be easily adaptable and customisable according to the specificity of the end user's needs. This increased modularity will also allow the evolutionary adaptation of the BIS to cover new and evolving business models.

Close liaison will be maintained with European and international standardisation bodies to ensure the end-languages and/or protocols are generically useful and not unique one-off creations.

4 The ABO Target Applications

In order to test and verify the ABO ideas and concepts two customisations of the ABO environment will provide an example of the *application layer*. The ABO framework will be developed in order to provide a common abstraction to highly differentiated end-user environments and namely a set of collaborating companies and a public administration.

4.1 ABO based ABC/ABM

The ABC/M model [2][4] relies on a producer-consumer relationship between three concepts that are necessary and sufficient to describe and model a Business environment: processes, activities and resources. We provide below a definition for each:

- **Process:** a process is a notion freed from any organizational or structural dimension. It captures a functional dimension, transverse to the organization, oriented towards a production objective. A process describes a functional objective of the business and is composed of activities and/or sub-processes together with the necessary resources involved in their fulfillment.
- **Activity:** an activity identifies an action which can be characterized by a verb and an object upon which the action applies. Activities are necessary in (i.e., consumed) fulfilling processes. Moreover, activities consume resources necessary to their fulfillment.
- **Resource:** a resource identifies in a very broad sense any production factor whose consumption occurs within activities (i.e., which are necessary for fulfilling activities).

We have decomposed the ABC/ABM ABOs in two distinct categories: Referential ABOs and Operational ABOs. The first category serving the purpose of capturing the business in terms of a model. The second category, the operational ABOs to capture the actual life (instances) of the business, through business objects. A central issue for ABC/ABC in the ABO framework is to be able at all time to measure everything and hence report activity through management instrument panels and allow action on the business to be taken both at the referential level and at the operational level. Although both categories will be modelled as agents, only the operational ABO agents will have migration capabilities. Referential ABO agents do not need mobility as they are bound to a given abstraction describing a business reason for which we consider them as static agents.

From there on, it becomes possible to measure and monitor these agents in a way similar to placing probes within them. As a result, managers and decision-makers can be given tools allowing them to build management instrument panels showing in real time the indicators for which they have expressed interest. These indicators can reflect information from either the referential level (i.e., Referential ABOs) or from the operational level (i.e., Operational ABOs). Furthermore, it also becomes possible to browse and navigate through the ABOs to audit the business trying to identify poor business patterns and malfunctions and thus take corresponding actions. Finally, goal oriented simulations and scenario evaluation become possible.

The ABO based ABC/ABM system will be installed and used by a Swiss public administration for the organization and management of personnel and resources. This user is already us-

ing ABC/ABM methodology and systems. The introduction of the ABO technology will allow them to improve the resource allocation and obtain a detail and real time view of the organization.

4.2 ABO based Supply Chain BIS

The industrial testbed application is directed to inter-enterprise usage (networks of suppliers and providers, clusters of contracting companies, Plug-In Enterprises, etc.) that will be applied to three Greek industrial partners that have established links between each other as they are affiliated. One reason that we have purposefully chosen to have user partners having this type of relationship is that in case the user partners had only the usual link of a client relationship, they would not be motivated to take the risk of opening their data/info infrastructure to other parts.

The users will be enabled within the project to exchange business data and information mainly administrative operations and in control of each company's supply chain related issues, such as inventory, order processing, financial procedures, etc. by forming seamless Information Supply Chains [12].

Using the ABO as an essential technology and infrastructure, the applications based on the ABO backbone will enable different application scenarios to be realised. In the case of a scheduling and management application, the particular company employees will provide the application with parameter data on each order which arrives from a client, as well as any necessary technical specification (this is of utmost importance the first time an item is ordered). The system responds with a first-pass scheduling, which the user may change. Finally, the application concludes the optimised production scheduling.

Though the IT infrastructure of the industrial companies of the ABO project, enables cost-efficient and highly competitive "under-order" operation by each of them, they all have not previously exploited the advent of agent-based technologies for business communication with clients. Transactions with clients still take place using paper, or face-to-face meetings, or via phone calls or facsimile messages.

More than a static business data repository, the pilot application will empower the companies clients to dynamically monitor workflow or production procedures related to their orders.

This eliminates the need to know who to call or the waiting for information to come via a facsimile message. As a consequence the discontinuities that often frustrate a company's customers and which can compromise business deals are greatly reduced.

Last but not least, as clients can continuously view the state of their orders, they are thereby better able to (re)organise their own internal business activities, thus contributing to harmonisation of a bigger part of the supply chain.

5 Why invest in ABOs

ABO brings a fresh approach in the way agent based applications and business system infrastructures are built and treated, which may be utilised for facilitating the proactive planning – from the side of the technology end users - of evolvable IT structures that help for a paradigm

shift in the way SME IT companies are proceeding to make-or-buy decisions in regard to investments they make for either new technology or new applications developments. Until now, resources have mainly been assigned to the latter, while any resources invested in the former – i.e. for new technology development – have been considered as mid-to-short run tactics for capitalising in the mid-to-long run in a specific market (segment).

Having in mind the convergence of information processing, communications and networking technologies and infrastructures, which is enabling applications to make vaster use of network processing infrastructures (as show in the figure) the ABO project poses the focus on technologies and infrastructures **common** to several applications, while those specific to one application only would be addressed in the context of employing ABO developments as the platform upon which innovative applications will be built, e.g. for E-Commerce, public administrations, etc.

While ABO is moving on the aim of increasing network (Intranet / Extranet / Internet) resources, it supports a beneficial change in the costs related to the development and usage of agent-based applications. As shown in the figure, Java-based applications that relied on the facilities of plain agent platforms have increased the cost parameters, as the approach was similar to the one met in older generations of custom-made systems.

Of course, this was not necessarily visible and identifiable by the end user, as the higher degree of network utilisation significantly improved the “big picture”. In any case, this was a barrier that would in different ways prevent the rapid development of services that had to be designed, implemented and executed for distributed (Intranet / Extranet / Internet) non-homogeneous environments that should support multi-platform inter-operation.

In this respect, what the ABO technology does help is to target applications and architectures for mass customisable and dependable systems.

References

- [1] J. White, *"Mobile Agents"*, General Magic White Paper, General Magic, Inc. 1996.
- [2] Johnson, H. Thomas, and Robert S. Kaplan. *"Relevance Lost: The Rise And Fall Of Management Accounting"*, Boston: Harvard Business School Press, 1987.
- [3] General Magic, *"Agent Technology: Odyssey"*, General Magic, Inc., <http://www.generalmagic.com/technology/odyssey.html>
- [4] H.T. Johnson, *"Organisational Design Versus Strategic Information Procedures for Managing Corporate Overhead Cost: Weyerhaeuser Company 1972-1986"* in *Accounting and Management: Field Study Perspectives*, Bruns W.J. and Kaplan R.S. (Eds), 1987, Boston, Mass.: HBS Press, pp.17-48
- [5] Jean-Henry Morin, Christian Kobel and Dimitri Konstantas, *"Active Business Objects (ABO): When Agents meet ABC/ABM based Management"* in *Trusted Objects*, ed. D.Tsichritzis, CUI, 1999
- [6] Ciarán Bryce *"Police Agents for Enforcing Internet Security"* in *Trusted Objects*, ed. D.Tsichritzis, CUI, 1999.
- [7] Dimitri Konstantas, *"Interoperation of Object-Oriented Applications,"* *Object-Oriented Software Composition*, O. Nierstrasz and D. Tsichritzis (Ed.), Prentice Hall, 1995, pp. 69-95.

- [8] Dimitri Konstantas, *"Migration of Legacy Applications to a CORBA Platform: a Case Study,"* Proceedings of the IFIP/IEEE International Conference on Distributed Platforms - ICDP'96, Dresden, Germany, February 27 - March 1, 1996
- [9] Jan Vitek, *"New Paradigms for Distributed Programming,"* European Research Seminar in Advanced Distributed Systems, ERSADS'97, Zinal, Switzerland, March 1997.
- [10] Jean-Henry Morin, *"HyperNews, a Hypermedia Electronic-Newspaper Environment based on Agents,"* Proceedings of HICSS-31, Hawaii International Conference on System Sciences, Kona, Hawaii, January 6-9, 1998., pp. 58-67.
- [11] Jan Vitek and Giuseppe Castagna, *"Towards a Calculus of Secure Mobile Computations,"* IEEE Workshop on Internet Programming Languages, Chicago, Illinois, May 1998.
- [12] Walsh P., Koumpis, A., *Introducing the concept of Information Supply Chains: The Buddy Project,* Journal of Logistics Information Management, Vol. 11, No. 2, 1998.