

Business Object Design and Implementation IV: From Business Objects to Complex Adaptive Systems

OOPSLA'98 Workshop Report

Jeff Sutherland
IDX Systems Corporation
jeff.sutherland@computer.org

Abstract

The NCITS H7 Object Information Management Committee and the Object Management Group Business Object Domain Task Force (BODTF) jointly sponsor the Workshop on Business Object Design and Implementation for the purpose of improving business object design and implementation through use of patterns, interoperable components, object-oriented workflow, and Internet strategies for distributed computing.

Of particular interest are heterogeneous distributed workflow systems for operation of enterprises over Intranets and Extranets. These systems need to provide business object solutions for mobile agents, process engines, and systems that exhibit emergent behavior. Interactive, autonomous, business object components may be intelligent agents roaming the net to perform complex tasks. They will exhibit complex behaviors, catastrophic events, and chaotic interactions and have been studied under the umbrella of "complex adaptive systems (*cas*) for use in predictive economic simulations, the building of artificial life, computer models that can independently adapt and evolve, and "avatars" that can personally represent the creator in Internet transactions.

NCITS H7 Object Information Management

H7 has recently become part of T3 - [Open Distributed Processing](#) (ODP).¹ Technical Committee T3 is responsible for the development of the National Standard Reference Model on Open Distributed Processing (RM-ODP) and represents the U.S. position at the International Standards Organization (ISO). RM-ODP models features and functions required to achieve a transparent distributed

processing platform and will be used to coordinate future standards development in the area of distributed processing. T3 Project Editor, Joaquin Miller (miller@shl.com) solicits support and assistance for development of the RM-ODP enterprise viewpoint standard.

OMG Business Object Domain Task Force (BODTF)

The Object Management Group's central mission is to establish an architecture and set of specifications based on commercially available object technology to enable distributed integrated applications. Primary goals are the *reusability*, *portability* and *interoperability* of object-based software components in distributed heterogeneous environments. To this end, the OMG adopts interface and protocol specifications that define an Object Management Architecture (OMA) that supports applications based on distributed interoperating objects.

Goals of the Workshop

1. Enhance the pattern literature on the specification, design, and implementation of interoperable, plug and play, distributed Business Object components. Specify how patterns can be reused and how they can be useful to end users.
2. Clarify the design and implementation of object-oriented systems, particularly systems in which workflow patterns and the REA accounting model are basic building blocks. These systems must deal with the impedance mismatch between application transactions (units of work) and Transaction Monitor (TP) transactions in large systems.

3. Contribute to emerging architectures for Intranet/Internet/Extranet applications, particularly those applications that integrate business objects, web servers, object and relational databases, and new approaches to client delivery of content. Strategies for implementation of real time, event driven systems with generative process planning, automated routing, and dynamically changing roles of resources are of interest.
4. Pursue issues developed in previous workshops on heterogeneous distributed workflow systems. Specify business object solutions to mobile agents, learning systems, process engines, and systems that exhibit emergent behavior. Cross-fertilize business object design concepts with experience from the field of complex adaptive systems. Address hierarchies of workflows, peer to peer workflows, workflow between organizations, separation of workflow from application business logic, generic workflow execution patterns that are part of every business process object, and workflow concepts that can be seamlessly interwoven into all applications.
5. Provide explicit experience reports on business object systems developed and in production, with emphasis on implementation of domain object models that are subject to dynamic change.

Many of the outstanding issues in business object design and implementation for enterprise systems involve federated, autonomous business object frameworks on the Internet acting collectively to solve business problems in an automated fashion. Manola describes the future construction of these distributed object systems on the Web.²

Workshop Presentations

Workshop position papers are available online.³ Many of the 1998 papers and those from previous years are revised and extended in the OOPSLA'96,'97,'98 Business Object Workshop Proceedings published by Springer.⁴

Session 1: Business Objects

[From Business Objects toward Adaptive Agents](#) by Pam Rostal, Compuware Professional Services

Considering units of work as autonomous goal-driven agents emphasizes the connection between business objects and complex adaptive systems. Assigning goals to work objects, in addition to the usual attributes, behaviors and rules, allows them to act as adaptive agents. Applying user-modifiable conditions to their actions keeps the agents goal-driven behavior within limits dictated by volatile consumer markets, changing corporate direction, new legislative rulings, etc.

[The Business Component Approach](#) by Peter Herzum and Oliver Sims, SSA

The business component is a single unifying concept that supports system definition and requirements and continues through deployment and customization to subsequent system evolution. It does this with minimal transformations across the development life-cycle, and is supported by appropriate processes, architectures and tools. If we are to address the multi-dimensional systems development problem for mission-critical, large-scale, multi-user, distributed business systems, we need a component concept that addresses the entire development life cycle.

[Business Object Component Architectures: A Target Application Area for Complex Adaptive Systems Research](#) by Jeff Sutherland, IDX Systems Corporation

Concepts in Complex Adaptive Systems (*cas*) research are relevant to the development of enterprise business object component systems. Many mathematical and computing models have been developed for *cas* in recent years⁵ and much of this work can be applied to Business Object Component Architectures now emerging as the mechanism of choice for building large distributed object systems.

Session 2: Workflow

[Structuring Specification of Business Systems with UML \(with an Emphasis on Workflow Management Systems\)](#) by Pavel Hruby, Navision Software

Unified Modeling Language (UML) symbols have defined semantics, which means that the visual workflow description can be used as a software specification. This supports UML specification of workflow management systems, tracability of business processes to object-oriented software design and structuring the project repository with UML deliverables.

[Building Workflow Business Objects](#) by Marc-Thomas Schmidt, IBM Software Group

The OMG is in the final stages of defining the Workflow Management Facility, an important building block for the emerging OMG Business Component Architecture. It provides a framework for workflow business components which enable realization of business processes in a business component environment. Workflow components take care of the overall process logic of a business process, enable monitoring of workflow execution, and support flexible combination of reusable business components into workflow applications.

Session 3: Modelling and Framework Issues

[Organization in a Chaotic World](#) by Chris Marshall, SES Software Inc, Marietta GA

Concepts described at OOPSLA 97 (Business Object Management Architecture) have been extended and refined using ideas drawn from complex adaptive systems. The paper proposes that traditional hierarchical enterprises that rely on command and control based on central authority will be replaced by networks of organizations that coordinate their work through contracts.

[Adaptive Framework for the REA Accounting Model](#) by Hiroaki Nakamura and Ralph E. Johnson, University of Illinois at Urbana-Champaign

Business applications must be able to adapt to a changing business environment, to access large amount of data, and to be efficient. A framework for business applications must also be reusable and provide uniform access to information. We have developed a framework based on the REA accounting model that meets these requirements. It uses the active object-model and composite query technique for adaptability, the Interpreter pattern and the integrated query system for

information uniformity, and incremental computation for performance. Each idea in our framework can be applied to many other business-object implementations.

[Business Procedures are not Represented Adequately in Business Applications and Frameworks!](#) by Hans Albrecht Schmid, FB Informatik and Fernando Simonazzi, LIFIA, UNLP, La Plata, Argentina

Recently, the business community discovered the importance of business process engineering, and the object-oriented community has put much emphasis on the structuring of business applications. However, there is a gap between the two efforts. Our claim is that state of the art business application structures and frameworks do not represent business procedures adequately. Business procedures are not modeled naturally, i.e. in the same way as they are described by the business community.

[A Business Object Framework Architecture](#) by Hans Albrecht Schmid, FB Informatik, Matthias Riebisch, Deutsche Post AG, Torsten Heverhagen, Informatik, University Essen, Harald Liessmann, Wirtschaftsinformatik, University

We advocate a 5-layer architecture for business object frameworks with the layers: presentation, business process, business entity, data access, and data storage, instead of the more common 3-layer architecture, and give reasons why the 5 layers are required.

["Ride The Mainstream!" with MACK Business Objects and Escape the Divine Programmer Syndrome](#) by Christopher Spottiswoode, Metaset, South Africa

MACK is the proposed standard "Metaset Architecture for Common Knowledge", where Metaset™ is a conformant program currently being developed towards full groupware. Good groupware is both the product and the facilitator of due commonality and standards. Full groupware includes both developer and user facilities, combined as a market infrastructure. It mediates groups such as supplier organizations, market segments, and communities of the two together, both within and between the groups.

Session 4: The User Experience

[Business Object Transitioning](#) by Lenny Estrin

This paper describes the experience and history of introducing Business Objects into large organization and illustrates the transition process. The road of establishing the Business Objects as a best practice in a large insurance company, which has a tremendous investment in a legacy code and which has had problem adopting a new technologies, has been a challenging one.

[Working with Business Objects: A Case Study](#) by W. Hordijk, S. Molterer, B. Paech, Ch. Salzmann, Institut für Informatik, Technische Universität München

We report on the first steps of a case study on the usage and adaptability of business objects (BOs). This toy-world scenario is the first stage of a study for a German automobile company to evaluate business objects especially from the reuse perspective. Therefore we transfer a well known example - a time-planner - from UML to CDL, realize it on the basis of IBMs San Francisco framework and take a critical look at the resulting benefits of BOs against ordinary object-oriented techniques.

[A Dynamic Business Object Architecture for Supporting Strategic Management Planning](#) by Kitty Hung, Tony Simons, and Srba Cvetkovic, The University of Sheffield

One of the current problems in the software industry is the communication gap between the business end users and the software developers. This gap tends to limit the software developers knowledge in the business in terms of requirements, strategies and operations. As a result, the software delivered cannot meet expectations and end users find it difficult to justify the return on investment from software. This position paper presents a Dynamic Business Object Architecture (DBOA) with an aim to fill this gap.

Conclusions

Complex Adaptive Systems

It is useful to apply *cas* concepts to enterprise business object component systems. For example, Holland creates a synthesis of seven relevant *cas* concepts.⁶ Odell has directly related *cas* to building distributed object systems.⁷ The following core concepts can help organize our discussion of business object systems.

Aggregation (property) - there are two important modes of aggregation in *cas* systems. Aggregation is a basic mechanism in object modeling and forms identity, a fundamental object concept. Forming components out of objects and enterprise systems from components is higher level aggregation. More important are emergent properties such as intelligence that evolve out of dumb subsystems. This is the basic concept in Minsky's "Science of Mind" or Hofstadter's analysis of an ant colony. Meta-agents (an enterprise) are formed of aggregates of agents (enterprise systems) and exhibit emergent behaviors (revenue, profitability, cash flow, the indices of value creation).^{8,9}

Tagging (mechanism) - this mechanism facilitates the forming of aggregates, from HTML pages to the mechanisms in CORBA or DCOM that allow interobject communication. They facilitate selective mating, i.e. firewalls block certain tagged elements to protect the enterprise. Thus they preserve boundaries between aggregates. They allow us to componentize object models and enable filtering, specialization, and cooperation. They are the mechanism behind the development of hierarchical aggregates that exhibit emergent behaviours like an operating system.

Nonlinearity (property) - nonlinear systems exhibit catastrophic and chaotic behaviors. Traffic flow on the Internet is nonlinear, leading to predictions of the collapse of the network. Brownouts, system loadings, scalability effects are often nonlinear. The arrival, proliferation, and destruction of viruses on the Internet is a nonlinear phenomenon that can be modeled like predator/prey interactions in biological systems.

Flows (property) - workflows are examples of flows in action. Message routing is a flow. Tags condition flows which often exhibit nonlinear behaviors and emergent behaviors. Flows typically have a multiplier effect. Money injected into the economy has an effect out of proportion to the amount, similar to email or other message flows on a network. The recycling effect of flows enables the rainforest, as well as an enterprise computer ecosystem. Individual pieces evolve, die, are replaced or reused, constantly changing the characteristics of the enterprise. Living software is software that is constantly changing due to flows, as rivers change their course. Dead software is

eventually detritus that is expelled from the enterprise organism.

Diversity (property) - persistence of an individual agent depends on the ecosystem of agents that surround it, whether the agent is an ant in the rainforest or a business object in an accounting system. The evolution of these agents as software changes causes convergence of system architectures. It is the basis of emergent patterns that are so important in software development, patterns that reappear again and again in widely disparate environments. It is difficult to evolve a single agent to make it more useful in an isolated context. Usefulness in business object systems arises from interactions between diverse agents, as in human societies.

Internal models (mechanism) - the utility of complex systems is enhanced if the system can learn from experience and adapt its behavior. The ability of the system to develop and act on internal models that simplify the external world is basic to this phenomenon. It allows the system to infer the results of actions before they are taken, and to choose actions that have productive results. The prospects for longevity of software systems depend on this capability, just as in living systems.

Building blocks (mechanism) - reuse is dependent on building blocks used over and over again. It is the basis of Moore's law in hardware production. It could be the basis of dramatic improvements in software productivity. Building blocks are the basis for generation of internal models and are essential to the construction of adaptive enterprise systems.

Since these seven cas building blocks have proved useful in describing cas systems in general, they could be the basis for a taxonomy of business object systems.

Business Object Components and Workflow

A core issue for business object development is lack of consensus on the definition of a business component. Herzum presented a comprehensive picture of a business component as a software implementation of a concept which itself is a **aggregate** of software artifacts (including distributed components) that provides a unifying concept through the development cycle and

across architectural tiers.¹⁰ It is worthy of note that developing an enterprise framework for support of a Business Object Component Architecture (BOCA) costs \$10-15M for a large organization due to lack of standards for **aggregation, flows, internal models, and building blocks**. Lack of standards prevent a market from developing standard components and tools.

Hubry provided a comprehensive overview of software artifacts essential to the business component life cycle with special emphasis on workflow and noted that UML did not support all artifacts. The inability to precisely define **aggregates, flows, and building blocks** impedes development.

Rostal's work on applying **flow** concepts to business objects implied that adaptive agents must function like distributed workflow components where each business object is its own workflow engine.

The current state of the OMG workflow standard was reviewed by Schmidt. The workshop noted that the OMG jFlow standard¹¹ is a major step forward in supporting non-centralized workflow models, but it did not support the concept of distributed business components as individual workflow engines, essential to emerging distributed workflow systems discuss by Paul in last years workshop.¹²

Modeling and Framework Issues

There is an emerging concensus apparent in the papers by Herzum and Schmid, and reinforced by Francis¹³ at last year's workshop that at least four architectural tiers are required for a BOCA— a presentation layer, a workspace layer supporting a business transaction, an enterprise layer supporting distributed business object components, and a resource layer supporting transactions, persistence, and other services.

Marshall's paper illustrated the necessity of moving beyond hierarchical execution structures. Without network based component interactions driving enterprise systems, the degree of adaptability required to keep pace with changes in business processes will not be achieved.

Once again, the REA accounting model was used as an example of a pattern that has wide

applicability in enterprise systems. The need for adaptive object modeling was demonstrated by Nakamura.

Problems with the impedance mismatch between real-world business processes and BOCA frameworks were raised by Schmid. Once again, Spottiswoode argued that his MACK approach to business objects provides the answer to such problems. Hung proposed a Dynamic Business Object Architecture as a mechanism to improve the communication gap between software developers and the business management.

User Experiences

While the patterns movement has had a significant impact on software development in recent years, Estrin noted the lack of standard patterns for BOCA development. This, combined with the lack of standard frameworks and components makes the introduction of the best approach to enterprise systems tremendously challenging in large organizations.

Hordijk noted that the lack of standard aggregation patterns for components in the CORBA environment resulted in excessive overhead in the use of CORBA services. The lack of a CORBA component model and an OMG BOCA that allows components to be used in an enterprise framework remains a severe impediment to building enterprise systems.

Open Questions

Once again some questions were answered and more were raised by the Workshop. A few unanswered questions provide a backdrop for next years workshop.

- How does an agent handle workflow/rules?
- How can workflow adapt to changing context?
- If a business component crosses four tiers and the object life cycle, how does the average programmer deal with this complexity?
- If a “customer” or “patient” object has multiple instantiations in the enterprise, how does a user work with a single component?
- Is UML enough for representing a Business Object Architecture?

- How does a component “drop into” a workflow?
- Does the current workflow standard handle agents? Distributed computation?
- Important accounting systems must be adaptive. Classes are not adaptive. How can we fix this?
- Can we implement adaptive frameworks without losing the benefits of strong type checking?
- What is the best way to build “logical units of work,” i.e. long transactions?
- Is the OMG BOCA useful? What else does it need?
- What is the best way to introduce a high level adaptive process into a BOCA?
- What is the relationship between workflow and a business process?

Workshop participants agreed to immediately post a call for papers for a 1999 Workshop. Papers will be due in February 1999 and published in a book shortly thereafter.

For an extended version of this report and listing of workshop participants, see: <http://jeffsutherland.org/oopsal98/>

References

-
- ¹ RM-ODP Home Page.
<http://enterprise.systemhouse.mci.com/>
 - ² Manola, Frank. *Towards a Web Object Model*. Position Paper for the OMG-DARPA-MCC Workshop on Compositional Software Architectures. Object Services and Consulting, Inc., 1998.
 - ³ *OOPSLA'98 Workshop on Business Object Design and Implementation*.
<http://jeffsutherland.org/oopsla98/>.
 - ⁴ Patel, D., Sutherland, J., Miller, J. (Eds). *Business Object Design and Implementation: OOPSLA'96-98 Workshop Proceedings*. Springer, 1998.
 - ⁵ Cowan, George A., David Pines, David Melzer (Eds.) *Complexity Metaphors, Models, and Reality. Santa Fe Institute Studies in the Sciences of Complexity Proceedings, Vol 19*, 1994.
 - ⁶ Holland, John H. *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley 1996.
 - ⁷ Odell, James. *Agents and Beyond: A Flock is Not a Bird. Distributed Computing, April, 1998*.
 - ⁸ Minsky, Marvin. *The Society of Mind*. Simon and Schuster, 1988.
 - ⁹ Hofstadter, D.R. *Godel, Escher, Bach: An Eternal Golden Braid*. Vintage Books, 1989.

-
- ¹⁰ Hersum, Peter. [The Business Component](#). OMG Document ormsc/98-09-01.
- ¹¹ CoCreate, Concentus, CSE Systems, Data Access, Digital, DSTC, FileNet, Fujitsu, Hitachi, Genesis Development Corp., IABG, IBM, ICL, NIIP, Oracle, Plexus, SNI, SSA, Xerox. *jFlow Initial Submission to the Workflow RFP*. OMG Document bom/97-08-05.
- ¹² Paul, S., Park, P., Chaar, J. Essential Requirements for a Workflow Standard. In Patel, D., Sutherland, J., Miller, J. (Eds). *Business Object Design and Implementation: OOPSLA'96-98 Workshop Proceedings*. Springer, 1998.
- ¹³ Anderson, Francis. [Object Models For Business Transaction Processing Systems OOPSLA'97 Workshop on Business Object Design and Implementation](#). Atlanta, 6 Oct 1997. <http://jeffsutherland.org/oopsla97/>