OBJECT-ORIENTED ENTERPRISE ENGINEERING

William D. Barnett, Assistant Professor
College of Business
Northwestern State University, Natchitoches, LA 71497

M. K. Raja, Professor
Department of Information Systems and Management Science
University of Texas at Arlington, Box 19437 Arlington, TX 76019

Introduction

Businesses around the world are attempting to position themselves to operate in a highly competitive global economy. To compete, producers of products and services must be able to meet rapidly changing customer demands. Drucker (Drucker, 1990; Zachman and Finkelstein, 1992) describes the emergence of a third wave of manufacturing organization. This new production philosophy is referred to as "produce to order". In a produce to order environment, the ability of an enterprise to react quickly to changes in customer needs and to supplier capabilities is paramount. This requirement has lead to increased emphasis on:

- Maximizing the efficiency of enterprise operations
- Establishing close relationships with both customers and suppliers (Gitlow, Gitlow, Oppenheim, and Oppenheim, 1989)

Accomplishment of these two activities requires that an organization focus on the processes that are used to provide goods or services to the end customer (ARRI, 1991; Davenport, 1993; Gitlow, et al., 1989; Hall, Rosenthal, and Wade, 1993; Hammer and Champy, 1993; Harrington, 1991; 1993). The emerging discipline of Enterprise Engineering has taken up the challenge of applying the rigor of engineering discipline to these very issues (Liles, Johnson, and Meade, 1995). One of the issues facing this new discipline is the development of tools and techniques to support the work of analyzing, designing, and implementing organizational systems. These tools must assist enterprise engineers in the initial transformation of functional, often disjoint, operations into a set of integrated business processes replete with supporting information and control systems. Enterprise engineers must also be able to perform the activity of reengineering business processes throughout the life of the organization.

This paper explores the application of object-oriented modeling concepts in the analysis and design of business processes. The limitations of traditional modeling and representation methods for supporting Business Process Reengineering (BPR) are discussed. The object-oriented systems paradigm is gaining support as a means of addressing the difficulties that have been identified in the process modeling literature (Barnett, Presely, Johnson, and Liles, 1994; Shelton, 1994; Taylor, 1995). The concepts for applying object-oriented modeling to Enterprise Engineering have been examined through active research with small and large companies, funded in part by the NSF and the Agile Aerospace Manufacturing Research Center (AAMRC). This paper reports on the results of an expert review that examined the application of object-oriented techniques for modeling business processes. Some of the lessons learned are presented.

Current Process Modeling Support Methods

The current literature describes a number of methodologies and techniques for performing BPR. These methodologies include the I-CAM Definition (IDEF) modeling, Petri Nets, Simulation, Information Engineering (IE), and Activity Based Costing (ABC) (D.Appleton, 1993; Davenport, 1993; James Martin & Co, 1993). Each methodology addresses some particular aspect or phase of an organizational BPR project. Representation and analysis is performed as a series of continuing projects. Successive changes in the competitive environment prompts new waves of re-analysis and re-design (D.Appleton, 1993; Hammer and Champy, 1993). BPR teams must develop or rework process models, information from supporting techniques like ABC is applied in a piece-meal fashion to evaluate design decisions.

Despite the widespread use of techniques like IDEF to support BPR, these techniques suffer from a number of significant shortcomings that hamper the activity of process modeling (Curtis, Kellner, and Over, 1992; Martin, 1993; Martin and McClure, 1988; Taylor, 1990; Taylor, 1995). The BPR literature has addressed the strengths and weaknesses of individual techniques (Davenport, 1993). However, examination of current process modeling practices (analysis and redesign) reveals severe problems at the paradigmatic level. A detailed review of analysis and design specification methods has identified three significant problem areas.

1. Disjoint Modeling of Process Perspectives
2. Difficulty in Validating Process Models
3. Lack of Support for Enterprise Modeling

Each of these shortcomings will be discussed in detail in subsequent sections of this paper.

<u>Disjoint Modeling of Process Perspectives</u>

A business process is a complex system of interacting elements. The complete description of a business process requires that a number of different perspectives or views of that process be defined. Curtis et al. (1992) has identified organizational, behavioral, functional, and informational aspects that are used by organizations attempting to understand their component processes. The CIM-OSA architecture identifies several such representational requirements (Jorysz and Vernadat, 1990).

Traditional process modeling methodologies typically emphasize one aspect of enterprise operation over another (Curtis, et al., 1992; Martin, 1989). In the area of systems development, data and activity have been historically modeled as distinct perspectives. The IDEF suite of methods formalize the split of organizational aspects into data, distribution, activity, and socio-technical aspects (Sowa and Zachman, 1992; Zachman, 1987). Each aspect is modeled from vastly different conceptual perspectives. The Functional aspect of processes or the enterprise are modeled using the IDEF-0 or similar activity oriented syntax. Informational and Business Rule aspects are typically captured using an entity-relationship based scheme (D.Appleton, 1993; Jorysz and Vernadat, 1990; Sowa and Zachman, 1992). In addition to separate representational schemes across process aspects, current BPR support methods provide varying representations at different levels of abstraction. Each row of the Zachman Information Systems Framework (IDEF Framework) uses a different representational scheme. A case in point is the transition from IDEF-0 to IDEF-3 in the functional aspect of the IDEF framework.

Separation of the perspectives or aspects of a business process into a number of separate models is an unnatural representation method (Barnett, et al., 1994; Martin, 1989; Taylor, 1990). The distinct pieces of a business process exist and have meaning as a single unit (1989; Martin and McClure, 1988). Curtis et al. (1992) maintains that a complete process description requires that all important aspects of that business process be present and consistent. Reintegration of these separate perspectives, both across perspectives (Who, What, When) and levels of abstraction (Analyst versus Builder level), tends to be quite error prone. The ongoing debate over the integration of IDEF-0 and IDEF1x is a case in point. Applying traditional process modeling methods, the enterprise engineer must contend with the possibility of degrading the enterprise design through (Martin and McClure, 1988):

- A loose of information content as models are converted
- An alteration in the meaning contained in the model

Difficulty in Validating Process Models

The issue of validating and verifying process models is closely related to the unnatural separation of business process views. Automated support for validation and verification is hampered by the large conceptual differences in the representational techniques used to portray the various aspects of a business process. Specific representations portray different information about the process. Information or business rule models typically provide a static representation (precedence and time are unimportant) of the process. Issues of timing and precedence are more germane in functional representational forms. The typical method used to link representations together is through accompanying narrative descriptions (Curtis, et al., 1992; Martin and McClure, 1988). Curtis et al. (1992) identifies this requirement for manual re-integration as a significant threat to the validity of the process representation.

An additional integration problem is the incorporation of the results of analysis methods into a process design. Current business process modeling methods are intended to provide a means of communication and understanding. The capability to represent the live operation of the business process and supporting systems is outside the scope of traditional representation schemes. Process analysis methods such as ABC or simulation require that yet another representation of a business process be constructed (D.Appleton, 1993; Taylor, 1995). Information derived from supporting analysis techniques must be manually applied to the process design.

Lack of Support for Enterprise Modeling

The process orientation in management emphasizes tight coupling of business processes throughout the entire customer-supplier chain (Davenport, 1993; Davenport and Short, 1990; Hammer and Champy, 1993; Harrington, 1991; 1993). The representation of an enterprise and the business processes that comprise that enterprise requires a means to portray inter-organizational interaction. Systems and processes that must cross organizational boundaries are by definition very complex (Markus, 1984). The development of organizational structures, information systems, and coordinative mechanisms is a highly time consuming activity. The modern competitive environment require that an organization be able to respond quickly to change (ARRI, 1992; Barnett, et al., 1994; Drucker, 1990). Support systems and business process configurations must be easily modifiable to address constantly changing business needs. Communication must be enabled across functional areas, between adjacent activities within a process, and between external participants in the customer-supplier chain (Davenport, 1993; Gitlow, et al., 1989).

Conventional methods, such as structured analysis and design technique (SADT) or IDEF-3 do not support the modeling of process interfaces at the inter or intra organizational level. In structured techniques (SADT, IDEF, Yourdon and Demarco) elements outside the purview of the system are modeled as external actors. Translation mechanisms must be constructed to reformat communications from these external actors into the semantics of the internal organization. These translation mechanisms represent additional overhead for the business process and adds little value to product. An alternative view that includes all elements of a process, both inter and intra organizational components, is equally problematic. The translation of inter-organizational semantics is performed as part of the analysis effort when the process is modeled. However, these semantics become inseparable parts of the model.

In either conception of using traditional structured techniques to model the enterprise, the time and effort required to redesign processes is prohibitive in a turbulent competitive environment. Interfaces must be built or semantics internalized to support the flow of material and information across the customer-supplier chain or divergent functional areas within the enterprise. Each adjustment to the business process requires adjustment to the process model (Curtis, et al., 1992; Taylor, 1995; Taylor, 1990).

Object Oriented Business Process Modeling

Proponents of a systems approach to Enterprise Engineering have suggested that any environment characterized by fast paced change, should be modeled as a series of interrelated component parts (Barnett, et al., 1994; Liles, et al., 1995; Rummler and Brache, 1990). Business processes are would in essence be modeled as "black boxes" (Schoderbek, Schoderbek, and Keflas, 1990). Black box processes permit information and material flow across the virtual enterprise while simultaneously protecting the proprietary nature of member organization business processes. Such an approach would also by definition address the issue of enabling rapid reconfiguration of processes by member organizations. In essence, business processes must be modeled in such a way as to permit "plug and play" compatibility between organizational processes.

The methods that are currently in use in the area of process design and reengineering are not adequate to support business process modeling in a high change competitive environment. The ability of traditional methods to present consistent and verifiable models is limited to the capability of the modeler to manually cross check different representations (Curtis, et al., 1992; Taylor, 1994a). Processes are dissembled into component views and must be re-assembled to gain a complete understanding. Further, more traditional methods do not support the construction of models from component parts or black boxes. A complete understanding of a process that flows cross organizational (internal or external) boundaries must include translation activities or necessarily include detail from other organizations. This exposes decision makers at the enterprise level to unnecessary complexity (Shelton, 1994).

The object-oriented paradigm provides a powerful solution to the process modeling needs of organizations in a competitive environment characterized by high change (i.e. the Agile or Virtual Enterprise) (Barnett, et al., 1994; Business Object SIG, 1995; Shelton, 1994; Taylor, ; Taylor, 1994b). The fundamental characteristics of the object-oriented approach listed below, provide capabilities that could greatly improve the quality process models and support the development of enterprise-level models.

- Classification: Under the object-oriented approach, real world phenomena (a process, activity, or actor) are perceived as whole entities. At a high level of abstraction, similar real world phenomena (processes, activities, or actors) are classified or grouped based on their shared characteristics and behaviors. These abstract groupings form classes.

- Encapsulation: All aspects of a given phenomena are encapsulated or contained within the representation of a class. Current work in object-oriented systems emphasizes the representation of characteristics (information view) and behavior (activity view). However, this model could be extended to also encapsulate other process perspectives, such as the resource and organizational views (ARRI, 1992; Curtis, et al., 1992). This property of the object orientation is referred to as encapsulation (Yourdon, 1994).

  The physical occurrence or instance of a class of phenomena is represented by an object. Objects are the instances of a class that have been identified within the organizational environment. The business environment is represented as the interaction of these real world phenomena or objects (Barnett, et al., 1994; Shelton, 1994; Taylor, 1995; Taylor, 1990; Taylor, 1994b). This arrangement provides a more natural representation of the organizational environment and improved coordination of the different contextual views of the enterprise.

- Message Based Communication: This representational method also formalizes the communications relationship between elements in the model. Communications is achieved through messages from one representation to another, no other interaction or access is permitted (Taylor, 1990; Yourdon, 1994). Objects are perceived as "black boxes" that maintain their own information and know how to perform their own basic behaviors. An object receives messages to perform behaviors, if the object does not possess this behavior then the message is ignored.

Standardization of these interfaces is particularly important to managing the customer-supplier chain given that communication is across physical organizational boundaries. The distinct competencies of individual organizations and trading partners is protected through this type of approach. Customers and suppliers can interact or competing enterprises can cooperate in a temporary venture since interaction is through a common set of messages or interface mechanisms. Adjacent activities or processes only perceive the interface and not the proprietary set of actions behind the interface.

- Inheritance: Specification of classes within the business environment can be conducted at varying degrees of abstraction. A feature of object-oriented systems is that classes have the ability to share their properties with more specialized forms of themselves. This ability is referred to as inheritance and is touted as a fundamental advantage for the object-oriented paradigm of systems (Taylor, 1990; Yourdon, 1994). A class can be refined into more specialized forms. Each of these specialized classes inherits the properties of the more general parent class. Subclasses may use the characteristics (properties and behaviors) as is and add new ones, or these characteristics can be modified to better support a specific need.

  In terms of modeling the enterprise, the class-subclass structure and inheritance are very useful properties of the object-oriented approach. One hurdle to process modeling in the virtual enterprise is the near infinite number of configurations that any particular process can assume. Production in any single member can involve any number of specific forms (drilling, plating, etc.). However, the fundamental structure of a production process remains the same; resources are changed and sent to a customer. The general case of a class structure provides a conceptual starting point for modeling the specifics of a given circumstance.

  Additionally, the effort involved in performing process modeling is eased through the reuse of general structures in modeling the more specialized process requirements The process modeling effort is more timely in that only changes from the general model must be specified. This characteristic is of immense importance in competitive environments characterized by persistent and rapid change.

- Polymorphism: Related to inheritance between classes in the object-oriented approach is the ability of classes, or more specifically object, to hide information. The property of polymorphism is also considered a fundamental characteristic of object-oriented representations (Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen, 1991). Polymorphism in terms of an object-oriented model means that a single message or command can have many meanings based upon the recipient of that message. A subclass interprets the meaning of a message based upon the internal definition of that message. Hence, the behavior invoked by a message to CUT could mean something different to a subclass that controls lathing than to a subclass that controls sawing operations. The type of CUT is determined by the specified behavior of the subclass performing the activity, not by an external controller or the activity requester.

Developing Business Objects

The principles of object-oriented modeling provide a set of stable components that can be reconfigured in a plug-and-play fashion to suit current business needs. Such a process representation supports the dynamic nature of the modern competitive environment (Business

Object SIG, 1995; Taylor, 1995). The independence of process activities and functions is achieved by representing business processes as a set of relationships between objects. These objects inherit the characteristics and behaviors of their all their parents up through a class hierarchy. New business entities are formed by creating subclasses or instances that inherit the definitions of their parents (i.e. a drilling activity class would be an subclass of a machining activity class). This enables the rapid reconfiguration of the enterprise and consistency in the way business processes are defined.

Encapsulation, message based communications, and polymorphism enables the standardization of the interface between activities and processes. Process objects do not require detailed knowledge of other objects. The *coordination mechanisms* required to allow business activities to function together can be standardized and implemented separately from the core activity that comprises the actual *business functionality*. Figure 1 portrays this separation of coordination mechanisms and business functionality. A change in one process (implemented as a change in the internal

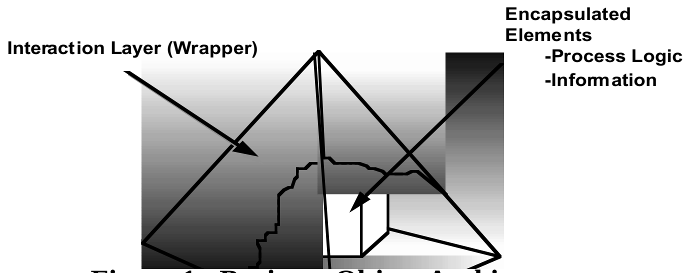

Figure 1. Business Object Architecture

workings of a process object) has no effect on the remainder of the system. Likewise, the substitution of an instance of a business object with another instance of the same class should have no impact on the overall enterprise. The new object will perceive the same messages for action and issue the same messages in return.

The class hierarchy proposed in this paper is intended as an initial starting point for the development of business object class definitions. Object concepts such as inheritance and classification can be applied to this base model to form specific representations of business entities for particular organizations and organizational environments.

Although much of the work done in the use of business objects emphasizes the development of computer information systems, the concepts of object-oriented business modeling are not limited to information system construction. The distinction that is made in Figure 1 between managerial and automation facets emphasize the separation of these two supporting, but different issues.

Object Modeling of the Enterprise

One of the main criticisms of applying the object-oriented paradigm to Enterprise Engineering is the computer programming orientation of most major object-oriented methodologies. Current literature on the object-orientation is primarily limited to low level systems development. Object-oriented systems *analysis* is a relatively new concept in the information systems area (Taylor, 1990; Yourdon, 1994). However, there is interest in elevating the focus of object-oriented thinking from the systems level to the business analysis level. This section will examine several of these efforts.

<u>Taylor Business Object Model</u>

Taylor (1990; 1994b; 1995) introduces the concept of "convergent engineering" or the "model-based" enterprise. Taylor's (1990; 1995) criticism of more traditional enterprise modeling schemes is that they are problem oriented. Each new problem or environment change necessitates the recompilation of the business models and supporting systems. Taylor (1990) proposed that base models of the enterprise should be constructed using objects. This model would represent a stable structure of the business over time. Specific process scenarios could then be built from this stable model.

Taylor (1994a) has proposed a base object-oriented class structure as a basis for business process modeling. In this class model the foundation class is the "business element". A business element consists of the rules of operation (formulas, facts, and rules) for that element. The business element also has the capability to control itself. This self control is accomplished by the ability to schedule activity, take action, and remember previous activity (history). All subclasses of the business element inherit the properties of this foundation class.

Taylor (1994a) proposes that Organization, Process, and Resource are the next level of specialization beyond the business element. The organization class is defined as "groupings of resources that are responsible for carrying out processes" (page 3). A process is defined in terms of consuming and generating resources to achieve some organizational objective. This definition of process as the fundamental conversion mechanism in the enterprise is not in line with the accepted use of the term in the BPR literature. The resource class is defined as an enabler and result of a business process. This definition of resource is equivalent to the input, mechanism, and output roles used in SADT and IDEF-0. According to Taylor (1994a; 1994b) the enterprise can be modeled as related objects based on these classes.

Jacobson Use-Case Approach

Recent work by Ivar Jacobson (Jacobson, Ericsson, and Jacobson, 1995) advocates the application of *use-case* method as a means of modeling the enterprise for process redesign. This approach is also grounded in information systems development. Information systems based on the object models of a redesigned (and optimized) business process is the goal of Jacobson's methodology.

In the use-case approach, the enterprise is seen as a series of scenarios of external actors *using* or interacting with the organization. The enterprise modeler must identify the pertinent actors and the scenarios that describe the organization. Each of these major use-cases represent the fundamental business processes of the enterprise. Within each use-case, a number of possible situations are possible as the actor uses the business process. Textual descriptions of each major scenario, and situational differences are developed, similar to the text descriptions developed as part of an IDEF-0 model (Marca and McGowan, 1988). Use-cases are further defined in terms of the interactions of object types that represent interface mechanisms, tangible entities, and transformation operations.

This approach is very valuable in partitioning the enterprise into core business processes. The nature of Jacobson's use-cases focuses on the way that customers interact and receive value from the enterprise, which aligns closely to the philosophy of business process reengineering and continuous improvement.

Object-Oriented BPR

Barnett, Hari, and Liles (1995) have drawn from the work of Taylor (1994a) to develop the foundation business object class model shown in Figure 2. The model was presented in a standard object-oriented modeling syntax, specifically that of Rumbaugh et al (1991). One of the shortcomings of most works on the use of business objects, Jacobson Use-Cases being the notable exception, is the lack of an established syntax to describe business domains. Using a standardized syntax aids in communicating the meaning and semantics of the model to other researchers and practitioners. Better communications in model presentation should improve the state of the research literature by permitting improvements to the base model.

The BUSINESS ELEMENT class is the abstract class that describes the fundamental aspects of a person, place, thing, activity in the business environment. This class defines basic characteristics (rules, formulas, and facts) and actions (scheduling and history keeping) of business entities in the most general terms. All of the subclasses of this class that are shown inherit these base characteristics and behaviors.

The ORGANIZATION class represents elements that impose control and constraints. Instances or subclasses of this class perform the basic role that control perform under the IDEF-0 modeling syntax. Barnett et al (1994) defined coordinative measures or constraining forces as part of the encapsulated characteristics of a business object. The explicit representation shown here provides a more meaningful definition for the construct of organization. Further, representing an organization as an object in relationships with other objects provides a more natural representation of the connection between organizational units

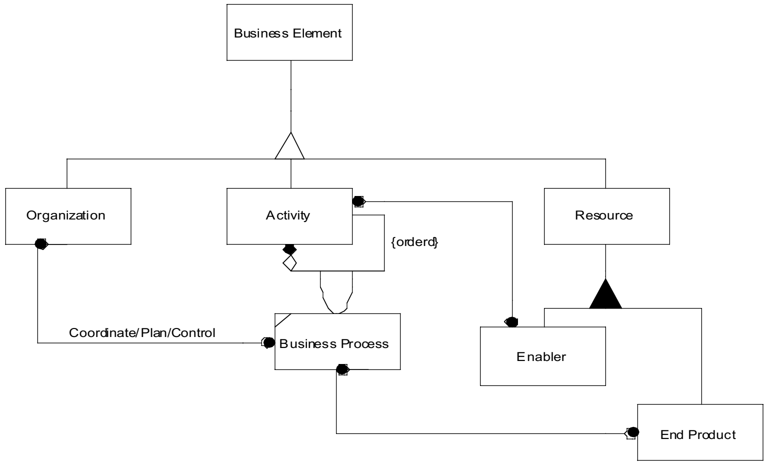

Figure 2. Proposed Object Class Hierarchy

Adapted from Barnett, W. D., Hari, H., and Liles, D. H. (1995). Object-Oriented Business Process Modeling. In B. W. S. a. R. Uzsoy (Ed.), 4th Industrial Engineering Research Conference, (pp. 769-778). Nashville, TN: IIE.

and the activities they perform. Specific behaviors, characteristics, and relationships for the organization class are derived from the control/constraint role that it plays.

Establishing business activities as the primary task component of a business process conforms to the way that business processes are defined in the BPR literature (Davenport, 1993; Hammer and Champy, 1993). Taylor (1994a) does not capture this distinction using process as a distinct subclass under business element. This portion of the class hierarchy is conceptually the same at that presented by Barnett et al (1994). A BUSINESS PROCESS class is an aggregate class that is composed of an ordered set of business activities, represented by the ACTIVITY class. Additionally, this model highlights the point that a single activity, such as lathing, could be involved in a number of business processes. The BUSINESS PROCESS class is involved in a relationship with the ORGANIZATION class to represent business constraints/controls on processes.

The RESOURCE class is defined in the same way as proposed by Taylor (1994a). However, this abstract class is further subtyped to refine the exact role played by the resource. Under Taylor's definition this role is somewhat ambiguous. The product class used by Barnett et al (1994) is also ambiguous. Subtyping into the roles of enabler and end product highlights the change of perspective inherent in the customer-supplier chain (Harrington, 1993). End products of one process or activity become the enabler of subsequent processes or activities.

Application of Object-Oriented Enterprise Modeling

The current study has addressed the utility and desirability of using the object-oriented paradigm as the basis for Enterprise Engineering in a highly dynamic competitive environment. However, the literature on the applying the object-orientation to BPR is very sparse. With the notable exception of the work by Jacobson et al (Jacobson, et al., 1995), descriptions of how to analyze business processes with objects remain grounded in software engineering. The Enterprise Integration Frameworks (EIF) group at the University of Texas-Arlington Automation & Robotics Research Institute (ARRI) are attempting to develop methods for applying object-oriented methods to the practice of Enterprise Engineering. This work on an object-oriented Enterprise Engineering method was followed by an initial pilot application.

The Technique

For the initial pilot study the foundation class model proposed by Barnett et al (Barnett, et al., 1995) was used to act as a guide for modeling a selected business process. The object modeling effort was accomplished by defining the elements within the enterprise that correspond to the foundation class types (ACTIVITY, ORGANIZATION, BUSINESS PROCESS, and RESOURCE) and the relationships between these elements. The characteristics and fundamental behaviors of the classes are identified in class definitions for each element that is identified. Specific types and sub-types of elements or classes are organized into specialization hierarchies or class hierarchies. The resulting class hierarchy provides a general structural understanding of the enterprise. This is the same general method employed by most methodologies for using business objects (Shelton, 1994; Taylor, 1994a; 1995).

Particular applications of technologies or business activities in are conceptualized as *class instances* or *objects*. In terms of Figure 1, the encapsulated operational logic, information, technology, and location are linked to the business process at the time of implementation rather than during design. These are the aspects of business process that are the most dynamic in the modern competitive environment. The separation of "*What is done*" from "*How things are done and by whom*" is referred to as late or *dynamic binding* (binding is borrowed from software development) (Taylor, 1995).

The Application

A sample application of the object-oriented process modeling method was undertaken to examine the usefulness of the fundamental classes (see Figure 2) and to provide a proof of concept for object-oriented business process modeling in general. Three staff members at the Automation and Robotics Research Institute with extensive modeling experience were asked to develop a process model using object-oriented modeling techniques. The sample process selected for object modeling was the "Feed the Family" process presented by Marca and McGowan (1988 page 201) in their widely used text on SADT modeling. Since each of the participants had used the text for reference and instructional purposes, the process seemed to be ideal for purposes of this test. The model presented in the Marca and McGowan text also provided a detailed IDEF0 model against which to compare the object-oriented representation. Using this example also provided the opportunity to examine the information content of an IDEF-0 and comparable object model.

The OMT diagramming syntax (Rumbaugh, et al., 1991) was used to derive an object model of the "Feed the Family" process. The models are not presented here due to space limitations. However, the results of the effort highlight one of the major problems in applying object-oriented modeling paradigms to Enterprise Engineering. The object model of the "Feed the Family" process entailed over 25 object classes, with each class involved in a number of relationships. The ability to understand this relatively simple process was compromised by the complexity of the representational scheme. This exponential explosion of complexity can most likely be attributed to the nature of the modeling technique. OMT, like most object-oriented methods, is primarily oriented towards developing specifications for object-oriented software. All aspects of the business process are modeled at a single level, hence a very large model. Given the inability of this representation to communicate the nature of the process, use of the OMT syntax was discontinued. Event trace and state transition models would also be developed as part of this modeling effort if a complete model is developed.

The class stereotypes defined by the foundation classes did prove useful in the identification of classes within the problem domain. The foundation class hierarchy provided the modelers with a map of how different aspects of the business process related. Inheritance of class properties aided the design effort by limiting the amount of documentation required.

Modeling with the Use-Case Approach

The same sample process was also modeled using the Use-Case approach described by Jacobson et al (Jacobson, et al., 1995). Application of the Use-Case approach proved to be more supportive of business process modeling. Figure 3 shows the definition of "Feed the Family" sample process in terms of the external actors affected and how those actors use the process. The ovals represent the use-cases or how the actors (stick people) use the process.

In a further decomposition of the *Eating a Meal* use-case, different scenarios or lines of activity were easily identified. Some of the alternative *eating* scenarios included *eating breakfast* and *eating holiday meals*. Greater understanding of the "Feed the Family" processes was developed under the use-case method of object-modeling than traditional IDEF-0 methods. An object model of the *eating* use-case was also developed (not shown). Again the foundation classes were very helpful and mapped very closely to Jacobson et al (Jacobson, et al., 1995) foundation object types. The distinction between types of resources was less useful.

Lessons Learned

Several fundamental lessons were learned as a result of this pilot study.

- The most fundamental result was that software oriented methods for object modeling must be modified for use in working at the business area or enterprise level. Subsequent work by the proponents of the two major object modeling techniques, Grady Booch and Jim Rumbaugh, have incorporate use-cases as part of their modeling syntax to support the definition of business process domains within an enterprise.

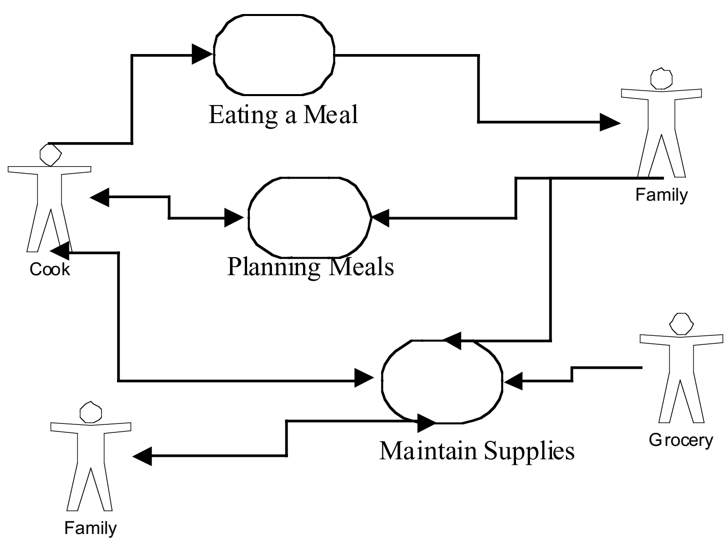

Figure 3. Use-Case Model of "Feed the Family"

- A base model of the types of elements and their relationships contained in a business process is very helpful in the modeling effort.

- Associated with this base class structure is the ability to use common descriptions in the development of process models. Common business object class descriptions can be used widely in the process model of an enterprise. Reuse of descriptions enhance the efficiency of the modeling effort and contribute to the integrity of the resulting model (Business Object SIG, 1995).

Conclusion

The ability to develop complete and consistent process representations is a critical requirement in any attempt to engineer the business processes of an enterprise. This study has examined current approaches to process modeling and found that there were major paradigmatic shortcomings. Since these shortcomings are rooted in the guiding paradigm of the methods studied, alterations in techniques or modeling syntax is not sufficient to remedy this situation. The adoption of object-oriented concepts and methods is proposed as a means to address these problems.

The application of object-oriented methods to the activity of BPR is a relatively new concept. The results of a pilot study on the application of object-oriented methods for process engineering has shown that this paradigm could be a valuable tool of Enterprise Engineering. Understanding of how to apply the object orientation to software applications and system modeling is fairly well developed in the literature and may provide some insight into the expected costs and benefits of developing enterprise object models.

The expected benefits to the organization from the ability to reconfigure business processes through the capability afforded by object technology are:

- Improved inter and intra organizational communications resulting from the use of the standardized message interface of the business object

- Improvements in the organizational ability to assimilate new technology into business processes. Process objects using old technology are removed and new ones are inserted, the interface to other organizational systems remain unchanged.

- Improvements in the development of supporting information and control systems by developing these systems directly from the business model.

Acknowledgment

Research for this project is funded in part by the State of Texas Advanced Technology Program Grant 003656-036 and by the National Science Foundation sponsored Agile Aerospace Manufacturing Research Center.

References

ARRI (1991). Perform Continious Enterprise Improvement Model (Working Paper No. Automation & Robotics Research Institute.

ARRI (1992). Business Process Templates for Small (Agile) Manufacturing Enterprises (Research Working Paper No. University of Texas at Arlington Automation & Robotics Research Institute.

Barnett, W. D., Hari, H., and Liles, D. H. (1995). Object-Oriented Business Process Modeling. In B. W. S. a. R. Uzsoy (Ed.), 4th Industrial Engineering Research Conference, (pp. 769-778). Nashville, TN: IIE.

Barnett, W. D., Presely, A., Johnson, M., and Liles, D. H. (1994). An Architecture for the Virtual Enterprise. In 1994 IEEE International Conference on Systems, Man, and Cybernetics, 1 (pp. 506-511). San Antonio, Texas: IEEE Systems, Man, and Cybernetics Society.

Business Object SIG (1995). Defining Business Objects (Special Report No. Object Management Group.

Curtis, B., Kellner, M., and Over, J. (1992). Process Modeling. Communications of the ACM, 35(9), 75-90.

D.Appleton (1993). Corporate Information Management: Process Improvement Methodology for DOD Functional Managers (2nd ed.). Arlington, VA: D. Appleton.

Davenport, T. H. (1993). Process Innovation: Reengineering Work Through Information Technology. Boston, MA: Harvard Business School Press.

Davenport, T. H., and Short, J. E. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. Sloan Management Review, (Summer, 1990), 11-26.

Drucker, P. F. (1990). The Emerging Theory of Manufacturing. Harvard Business Review, 68(3), 94-102.

Gitlow, H., Gitlow, S., Oppenheim, A., and Oppenheim, R. (1989). Tools and Methods for the Improvement of Quality. Homewood, IL: Irwin.

Hall, G., Rosenthal, J., and Wade, J. (1993). How to Make Re-engineering Really Work. 71(6), 119-131.

Hammer, M., and Champy, J. (1993). Reengineering the Corporation: A Manifesto for Business. New York, NY: HarperBusiness.

Harrington, H. J. (1991). Business Process Improvement: The breakthrough Strategy for Total Quality, Productivity, and Competitiveness. New York, NY: McGraw-Hill, Inc.

Harrington, H. J. (1993). Process Breakthrough: Business Process Improvement. Journal of Cost Management, 6(3), 30-43.

Jacobson, I., Ericsson, M., and Jacobson, A. (1995). The Object Advantage: Business Process Reengineering With Object Technology. Reading MA: Addison-Wesley Publishing Co.

James Martin & Co (1993). Business Re-Engineering: Quick Reference Quide (2nd Release ed.). James Martin & Co.

Jorysz, H. R., and Vernadat, F. B. (1990). CIM-OSA Part 2: Information View. International Journal of Computer Integrated Manufacturing, 3(3), 157-167.

Liles, D. H., Johnson, M., and Meade, L. M. (1995). Enterprise Engineering: A Discipline? In Society for Enterprise Engineering, (pp. This proceedings). Orlando, Florida: Society for Enterprise Engineering.

Marca, D. A., and McGowan, C. L. (1988). SADT: Structured Analysis and Design Technique. New York, NY: McGraw Hill Publishing.

Markus, M. L. (1984). Systems in Organizations. Marshfield, MA: Pitman Publishing Inc.

Martin, J. (1989). Information Engineering, Book I: Introduction. Englewood Cliffs, NJ: Prentice Hall, INC.

Martin, J. (1993). Principles of Object-Oriented Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall.

Martin, J., and McClure, C. (1988). Structured Techniques: The Basis for CASE (Revised ed.). Englewood Cliffs, NJ: Prentice Hall, INC.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). Object-Oriented Modeling and Design. Englewood Cliffs NJ: Prentice Hall.

Rummler, G. A., and Brache, A. P. (1990). IMPROVING PERFORMANCE: How to Manage the White Space on the Organization Chart. San Francisco, CA: Joseey-Bass Inc.

Schoderbek, P. P., Schoderbek, C. G., and Keflas, A. G. (1990). Management Systems: Conceptual Considerations (4 ed.). Homewood, IL: BPI Irwin.

Shelton, R. E. (1994). Business Objects & BPR. Dat Management Review, 4(11), 6-20.

Sowa, J. F., and Zachman, J. A. (1992). Extending and Formalizing the Framework for Information Systems Architecture. 31(3), 590-616.

Taylor, D. (1995). Drag & Drop Business Engineering. Enterprising Objects, 2(1), 3.

Taylor, D. A. (1990). Object-Oriented Technology: A Manager's Guide. Reading, MA: Addison-Wesley Publishing Co.

Taylor, D. A. (1994a). A Framework for Business Objects. Enterprising Objects, 1(2), 3.

Taylor, D. A. (1994b). Toward the Model Based Business. Enterprising Objects, 1(1), 1-2.

Taylor, D. A. (1995). Business Engineering with Object Technology. New York, NY: John Wiley & Sons.

Yourdon, E. (1994). Object-Oriented Systems Design: An Integrated Approach. Englewood Cliffs, NJ: Prentice-Hall.

Zachman, J., and Finkelstein, C. (1992). A Framework for Development. In Informatin Engineering Systems Corporation.

Zachman, J. A. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, 26(3), 276-292.