

Artifact-centered operational modeling: Lessons from customer engagements

K. Bhattacharya
N. S. Caswell
S. Kumaran
A. Nigam
F. Y. Wu

For almost a decade, the artifact-centered operational-modeling approach for modeling business operations, also referred to as the “business artifact method,” has been practiced and refined. This approach has been used in a variety of engagements, and each engagement has brought forth innovations that have enriched and strengthened the approach. In this paper, we describe three of these engagements in order to illustrate the method and highlight some of the lessons learned. The main objective of this paper is to establish the value of operational modeling in business transformation and to incorporate the lessons we have learned into a more comprehensive account of the method. We also describe the model-driven business transformation toolkit, which adds a unique value proposition to the method—the rapid and effective transformation of operational models into implementations that are manageable and can be monitored.

INTRODUCTION

Our approach to modeling business operations based on business artifacts¹ was developed in the mid-1990s. The design objective was the linear scaling of business transformation effort—that is, small business changes should require a small information technology (IT) effort and large business changes should require a large IT effort. The key to achieving this objective was the design of a formal model, based on factorization of business operational knowledge into information, function, and flow components.

The modeling language used in our approach had some similarities with data-flow diagramming² and flow-based programming,³ but was distinguished by

identifying one information structure, the *artifact*, that traveled from end to end in a process, hence the name *artifact-centered operational modeling* (ACOM). Another interesting but distinct approach focuses on objects, processes, and states, and is targeted at system development and specification rather than on business analysis.⁴

At the outset, the ACOM approach was seen as simply an alternative to more familiar approaches,

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/07/\$5.00 © 2007 IBM

such as the process-modeling and activity-flow approaches.⁵ However, as the method evolved, we discovered that the approach provided novel kinds of articulation and analytical possibilities. Using it to model business operations in a few dozen engagements has led to refinements of the approach and utilization of the method. The experience gained suggests that models created with the ACOM approach serve as boundary objects⁶ linking business design, organizational structure, performance measurement, resource management, and the original process focus.

A related development in the last few years was the model-driven business transformation (MDBT) effort at IBM⁷ that provided a principled and semiautomatic transformation of operational models into running applications. An MDBT toolkit has been developed to support modeling and model transformations in a model-driven development environment. Thus, a new value proposition for the approach emerged, namely the rapid realization of IT solutions, which led to its being used in more engagements.⁸ The experience from the engagements resulted in further refinements to the approach.

This paper focuses on the definition of the ACOM approach, our experience, the value it has delivered, and the lessons that have been learned from its use. We describe three broad classes of value: (1) reduced complexity and business level articulation, (2) the ability to analyze and reconcile business changes from multiple perspectives (e.g., process and organization), and (3) the ability to create IT realizations rapidly (as a result of the MDBT toolkit.)

Based on our experience from the engagements described here and other engagements, we suggest that the utility of artifact-centered operational modeling is most evident when it is used as a technique for business transformation engagements from their inception. We believe that creating models with ACOM can serve different purposes for different situations, depending on the business context and the nature of the transformation being pursued.

In the next section, we present a brief introduction to the ACOM approach. Next, we develop a framework for IT-enabled business transformation,

based on MDBT, with artifact-centered operational modeling providing a crucial underpinning. We also present a summary of the way MDBT creates IT realizations from an operational model. The bulk of the paper describes three engagements where the approach was employed, the results that were delivered, how these impacted the business, and some specific lessons learned. We also discuss some situations where the approach was not successful. We demonstrate how operational models provide a base representation with interpretations in each of the different concerns of business transformation engagements, while providing direct utility as the primary view of operational management.

THE ACOM APPROACH

In this section, we provide an overview of the basic concepts of the ACOM approach; details, including references to other tools and techniques for business process modeling, can be found in our earlier paper on business artifacts.¹

Operational modeling

The goal of an operational model is to represent business operations at a granularity that is sufficient for validating and managing progress toward goals and to clarify the dependencies between goals. Details of the execution are encapsulated in order to achieve localization of operations.

The modeling language we used is an extension of dynamic entity flow representations, such as Petri nets.⁹ Dynamic entities flow through a static network of nodes and arcs. Some of the characteristics of the model are occupation patterns, flow rates, and node populations. These properties allow us to model overall system behavior through a composition of smaller underlying units. The distinguishing extension of artifact-centered operational models is that of recording identity and properties modified at network nodes directly on the flow entity. The fundamental elements that are used in creating an operational model are business artifacts, tasks, repositories, and flow connectors.

A *business artifact* is an instance of a flow entity in the network. It has an identity and thus can be tracked as it progresses through the network. It also has an arbitrary set of attributes that are created, updated, or deleted at the network nodes. Business artifacts are self-contained and may be represented

as nested structures of name-value pairs. In business terms, this means that the artifact represents the explicit knowledge concerning progress toward a business operational goal at any instant. Operational goals, such as processing a customer order, are measurable results that individually and in the aggregate satisfy the purpose of the business. The information contained in the set of business artifacts records all the information about the operation. Hence, at any stage, the state of the modeled business can be determined by examining the collection of its artifacts.

A *business task* is an active node, encapsulating function that makes a change to one or more business artifacts. We use the terms *business task* and *task* interchangeably. In order to do its work, a business task must have exclusive control over one or more artifacts. Artifacts may move into and out of connected tasks or repositories (as described in the following subsection). The arrival of an artifact triggers the actions of the task; the task is completed when all artifacts have been sent out. A task may also be triggered by external events. It then creates a new artifact or retrieves one from a repository, to continue processing.

Additionally, resources which are required to perform the task may be associated with the task. The granularity of a business task can be viewed in two ways. From an operational perspective, the task granularity is the smallest change in one or more artifacts that makes a material difference in the scope of the model. For example, obtaining credit-check results may be an observable step forward in the context of completing an order. Dialing the phone may be necessary for performing the credit-check function, but the significance of the credit check is independent of the details of how that function is performed. This distinction leads to a view of task granularity from a process perspective. Within practical constraints, the amount of resources consumed or the complexity of activity within the task is not limited. This granularity may vary considerably depending on the business situation.

An *artifact repository* is a passive node containing a collection of artifacts that are not being worked on by any task. For example, these may be artifacts that have been fully processed or those that are awaiting further processing.

Artifact flow connectors are the conduits in the network through which artifacts may flow from task to task, from task to repository, and from repository to task. These pipes provide for reliable neutral transport of artifacts without any change to the contents of the artifacts that traverse them.

The artifact flow connectors attach to nodes at specific connection points, called ports, which govern the flow of artifacts into and out of tasks. Ports can have various properties, as described in Reference 1. Non-artifact interactions are also provided, which represent information transfer between tasks and agents external to the scope of the model. A phone call from a customer and a letter that is mailed to a vendor are examples of such interactions. These and other details are omitted here since they do not have a direct bearing on the objective of this paper.

Method for capturing operational models

The method presented here for capturing business operational models relies on the business interpretation of the formal model described previously. The key definition is that of a business artifact as a concrete representation of the knowledge available to the business concerning progress toward a specific operational goal. A business task is the smallest unit of progress toward one or more goals (if multiple artifacts are involved) and it records the progress in the artifact. The objective of the method is to create a network of tasks and repositories, and the artifacts that flow through it. This network is a faithful representation, for purposes of analysis, of the dynamic behavior of the business under consideration.

Business transformation engagements typically identify and prioritize problems and opportunities for improvement based on the functions or organizational areas in which they occur. The first objective of creating an operational model is to refine the identified problem area into a precise scope. For simplicity, we will assume that an entire business is being modeled, although the same procedure is applicable to a smaller scope as well.

Most engagements start out with a strategy already in place or help establish the strategy for a specific business context. The strategy is then concretized in terms of business goals and the metrics that will be

used to track them. An examination of the business goals and the related metrics provides essential clues as to what the business needs to do in order to achieve these goals. This then leads to the operational emphasis that has been the focus of all the engagements discussed in the paper.

Artifact-centered operational modeling begins by creating a set of operational goals that will be sufficient to realize the business goals. Often this requires examining and analyzing a wealth of information that may be available in terms of organizational structures and business processes as well as IT systems in place to support the business. Another way to formulate operational goals (a “bottom up” approach) is to consider the operational metrics that are needed to compute the business metrics. The exercise of determining the operational goals is essentially one of factoring the operational knowledge into meaningful and reasonably separable units.

Most often the precise scope chosen involves a result; progress toward the goal of achieving this result is embodied as the “principal” artifact. The engagements described later in this paper provide insights into how these artifacts may be discovered. Once an artifact has been identified, its life cycle is constructed, from creation to eventual archival. This requires establishing, in order, the tasks that will modify the artifact. It also requires determining points or stages where the artifact needs to be stored in a repository before further processing.

Because operational goals may stem from different kinds of business goals, the kinds of questions that must be asked will be based on the appropriate organization, process, or IT system models. From an organizational perspective, it is appropriate to ask, “What business artifact does the organization produce?” The same question can be asked of an operating process or a supporting IT system. The basic assumption is that the business artifact is targeted at a business-sensible level; this is discussed in greater detail in Reference 1. In the case of health-care management, the business goal of being more responsive to the needs of primary-care physicians leads to the operational goal of quickly processing medical orders. This in turn leads to the identification of a “medical event authorization” artifact.¹

Even though the business artifacts are concrete and distinct, they need not be independent. This means that changes to related business artifacts may need to be coordinated; that is, when an artifact is modified, the related artifact also needs to be updated in a consistent manner. This coordination appears in our operational models as one or more business tasks that need to modify more than one artifact atomically and reliably. The coordination can also be understood as a linkage (or dependency) between operational goals.

Once the business artifacts have been identified, the operation of determining how each artifact is processed involves identification of the tasks that produce measurable progress toward the operational goal. Note that for coordinated goals the task is defined by producing measurable progress for each of the goals.

In order to design a business task, one needs to perform “information accounting.” This means asking questions as to what information is available in the artifact that starts the task, what information must be added in order to accomplish the goals of the task, and where the information comes from. The information that is added is either created by the task (e.g., a computation or external source) or comes from another artifact present in the network that the task must acquire in order to complete the processing. This is how tasks implement the dependencies between artifacts. In artifact-centered operational modeling we consider tasks that modify multiple artifacts as *correlation tasks*. These tasks correlate business artifacts by effecting coordinated changes on them. A caveat that is significant when designing tasks is that a task has no knowledge of the tasks that may precede it or follow it; a task has to work with the information that is contained in the artifacts in its possession. Once a task completes all the artifacts on which it is working, it ends. A task has no business state as such; all such states are carried in the artifacts.

The following example illustrates a situation in which an artifact is placed in a repository between tasks. A series of business tasks may need to modify an artifact in no specific order. In such situations, the artifact is placed in a repository, and each task can access the artifact from the repository, add the pertinent information to the artifact, and return the artifact to the repository. Of course, the tasks need to

be designed so that they can cope with the unavailability of an artifact in the repository when another task is accessing it.

In conclusion, this method for artifact-centered operational modeling is an effective approach that starts with an artifact and constructs its entire life cycle, representing it with a graph of tasks and repositories. In constructing the life cycle, further artifacts are discovered and their life-cycle graphs are constructed in turn. Tasks whose artifact life cycles intersect provide coordination within the business operation. Further discussion of operational modeling and how it contributes to business agility is available in Reference 8.

A FRAMEWORK FOR IT-ENABLED BUSINESS TRANSFORMATION

In this section, we examine the role that operational models play in business transformation and explain how the content that ACOM provides is extended to build IT assets.

The objectives of business transformation efforts range from strategic assessments to the design and implementation of an IT solution for a set of business problems. A framework of the operational models used in MDBT is shown in *Figure 1*. It consists of four abstraction layers, each providing a model of the enterprise behavior from a different viewpoint. The layers are described by the strategy model, the operations model, the solution composition model, and the IT implementation model. The use of the ACOM model in this approach provides an innovative direct linkage between the strategic and development layers. From the ACOM layer through to deployment, the models are formally connected using model transformation techniques.

Applying ideas from model-driven architecture (MDA)**,¹⁰ a team at IBM Research has developed the MDBT toolkit. The toolkit contains models, method elements, and tools that support different stakeholders in various phases of the solution-development life cycle. ACOM is at the heart of the toolkit, both conceptually and in its embodiment of built-in techniques to map from the ACOM model to a solution composition model. This toolkit has been used in the engagement examples discussed in this paper and has evolved into a robust methodology.¹¹ Information about the MDBT toolkit can be requested from the authors. In the following, we describe the four layers of the framework.

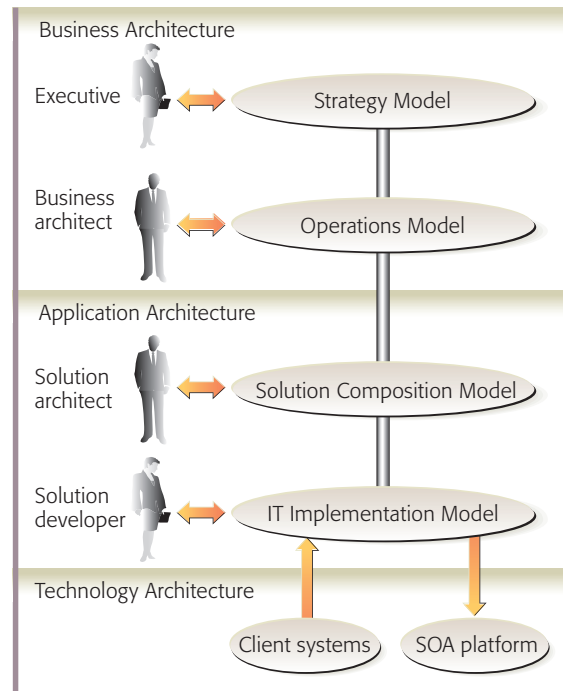


Figure 1
Four-layer framework for MDBT

Strategy model

The strategy layer, in which the business objectives are specified, is informal at present, utilizing existing tools. For example, the objectives may be specified in terms of the well-known “balanced-scorecard” perspective.¹²

Operation model

The ACOM approach is at the heart of the operation layer. This layer models the structural and behavioral aspects of business. Additionally, business rules based on the vocabulary derived from the business artifact information can be modeled at this level. In the MDBT toolkit, the IBM WebSphere* Business Modeler¹³ is used to implement operational modeling.

Solution composition model

The solution composition model is targeted to IT stakeholders who are responsible for the design of IT solutions that support business operations. The solution design is a platform-independent blueprint of the solution to be implemented. We model solution design using UML** (Unified Modeling Language) and UML Profiles,^{14,15} as these modeling

languages are deemed most appropriate for IT stakeholders. The MDBT toolkit uses Rational* Software Architect (RSA)¹⁶ as the tooling platform for the solution design model. The purpose of the solution composition model is to integrate the behavior of the business (captured in the artifact life cycle and the execution details of tasks), the structure of the business (captured in the information model), and the access of clients to operations and information (captured in the access model). Business artifacts and artifact-centered operation models are central to establishing this linkage.

The behavior of the business artifacts as expressed in an operation model is captured as a UML2 finite state machine, which is a representation of Harel state charts.¹⁷ Each information entity that is stereotyped as a business artifact has an associated state machine model. Details of the mapping from artifact operations to the state machines can be found in Reference 8.

The behavior model allows for solution extension. Specifically, the finite state machine model supports the notion of transition actions, that is, actions invoked on a specific transition. Business tasks do not model their internal behavior; rather, they are modeling primitives that allow one to reason about the composition and results of activities that are executed to complete the business task, without describing what these activities are. The execution of a business task is modeled in the solution design model and the finite state machine model, allowing the activities executed within a given business task to be modeled. Each transition action has a defined interface to specify information that will be exchanged at the transition. The interface definitions can also be defined as standard Web Services interfaces using Web Services Definition Language (WSDL).¹⁸

The information model contains business entities that represent both artifact information and referential (non-artifact-related) information. The entity relationships are represented in object model fashion, where containment relationships are modeled through UML aggregations and references through association. We apply business entity stereotypes to regular business items and the business artifact stereotype to business items that have an associated life cycle.

The access model defines the client interaction with the solution. It specifies the information to which a role has access for a given task. The information may be specific to a business artifact but can also contain any number of elements or bulk queries from any part of the information model. Furthermore, the access model describes which business actor can trigger operations to claim ownership of and complete a business task. The client access model has its own stereotype, called the views stereotype.⁸ Business tasks are modeled through UML business use cases. Roles are defined as business actors and can be associated with business use cases. The access model entities can be associated with one or many business tasks.

The MDBT toolkit contains transformation technology that maps business operation models into the solution design model. Solution extensions can be defined by customizing the client access model and the transition actions. The behavior of the business artifacts and their information content cannot be modified at the solution design level because any changes in the behavior of the operations must be driven by the business stakeholders and hence, must be performed in the operational model.

Implementation model

The implementation model is a platform-specific realization of the solution design model. This model uses the service component architecture (SCA) model¹⁹ to specify the assembly of the solution. SCA is a set of specifications that describe a model for building applications and systems by using a service-oriented architecture (SOA). SCA extends and complements prior approaches to implementing services and builds on open standards such as Web Services. The main component assembled in the implementation model is the business state machine (BSM), which is used to manage the life cycle of the artifact. The BSM uses a state-machine model to manage states, transitions, and invocation of transition actions. The assembly diagram connects the BSM with the solution extension components that implement the work performed within a task. The solution extension components are typically services that need to be implemented based on system requirements. The access model facilitates the generation of client components that allow integration with the information. The current version of the MDBT toolkit supports client access component

generation based on service data objects (SDOs).²⁰ The SDO metadata reflects the information specified in the client access model in the solution design phase. A user interface or any other client can use the generated SDO to interact with the solution.

ENGAGEMENT CASE DESCRIPTIONS

We have worked on a variety of business transformation engagements, applying the artifact-centered operational-modeling approach. In this section, we present some selected engagements that provide guidance for those using the method and bring out some of the finer points of the method.

The engagements chosen cover a range of business areas. First, we describe aspects of the primary value chain of businesses, exemplified by supplier and customer transaction processing in the health-care insurance industry. Internal processes directly connected to the optimizing value chain are exemplified by the forecasting process of a major retailer. Finally, the area of service management is represented by an engagement in which the outcome depended on increased clarity in understanding the strategic value provided by the business. In all of the engagements described here, the final client deliverable included the realization of the business transformation as a deployed application.

The engagements are described in a manner that focuses on the practical details and decisions involved in solving the client need, rather than illustrating the formal approach. In each case, we describe the business context, business problem, ACOM involvement (i.e., the factorization of the engagement scope into operational goals and the artifacts that represent them), analysis and insights, and a summary of the value ACOM delivered to the business and the challenges encountered in doing so.

Provider management at a health insurance company

A major health insurance company had been growing by acquisitions and had decided to embrace the Six Sigma** methodology²¹ to improve and consolidate its processes. They had trained several key personnel as Six Sigma “black belts” (i.e., trained experts) and initiated a few Six Sigma projects. We approached them to propose a collaborative effort grounded in operational modeling. We met with the business operations executives in the

provider management side of the business, who immediately saw a good opportunity to try our methodology on a recently begun Six Sigma project to redesign their provider data management processes.

This health insurance company was struggling to keep the database of physicians in its provider networks up-to-date. The main problem was that some requests were taking many months to complete, delaying the processing of claims filed by those physicians. The company had to process large volumes of data coming from physicians, such as requests to be added to an insurance network, to update physician information (e.g., a new office address or phone number), and to be terminated from networks (e.g., upon the physician’s retirement or relocation). Processing these requests was performed at numerous offices throughout the geographic service areas of the company, each of which eventually updated a centralized provider database. The processing was not complex, but it frequently required contacting the physician multiple times to ensure the completeness and accuracy of the data. Some cases required verification of the physician’s credentials. Because the company had grown by acquisitions, each office had a somewhat different process for handling the data requests. These local processes were supported by local ad hoc tools including spreadsheets, local databases, and document templates for keeping track of the data requests. Although operations management had attempted to institute monitoring systems to identify problems, the reported metrics were unintelligible due to the lack of process consistency and the inaccessibility of data in the disparate local tools used in the various regional offices.

The initial focus of the project was to arrive at a set of provider data management processes that all regional offices could adopt. As part of the Six Sigma effort, operations management had asked representatives of the regional offices to draw models of their provider management processes. The result was a set of drawings that appeared to be describing very different processes, although all the representatives agreed that they were doing essentially the same thing. It was at this point that management wanted to see whether operational modeling could help define the processes in a standard manner and help them gain control over the business.

Resistance to organizational change was perceived as a significant obstacle. It was understood that the way of doing things would have to change at some of the locations. On the other hand, it was also made very clear that the process definitions had to allow for regional variations that were dictated by state regulations or the local business environment.

Another aspect of the project was to implement a centralized Web-based tracking system, as well as a data extraction utility and monitoring display that could generate status reports for management on demand. Management understood that some of the changes involved in standardizing the processes might have implications for the organizational structures in the regional offices. To avoid unnecessarily influencing the redesigned processes with political considerations, the operations executives declared that no organizational changes would be made as a result of the process redesign. A direct result of this decision was that a person playing a role in one region might belong to a different organization than a person having the same role in another region. However, it was felt that this situation would not have any significant negative impact.

The first step in redesigning the processes was to review the drawings created by the representatives of each of the regional offices. This process revealed the commonality among and the differentiation between the regions. The next step was to ask these experts to look at their operations from a different point of view, namely the artifact-centric view. When asked to define what business artifacts were critical to the provider management process, the process experts quickly agreed that there were four types of requests (and their corresponding artifacts): add provider, add provider with credentialing, update provider, and terminate provider. An operational model for the processing of the update-provider artifact is shown in *Figure 2*. Each of these request types was grounded in clear operational goals. Interestingly, when the team defined the data that comprised these requests, these were identical. Yet the tasks that needed to be performed for each type of data request differed significantly.

For example, to add a new physician with credentialing, there were tasks to perform the credentialing, which was a known source of process delays. The life cycle of the terminate provider artifact

included tasks to analyze the impact on the patients in the network of terminating the provider and to reassign patients to one or more physicians in the same network. These differences made it important to distinguish between the four types of artifacts and to report the performance metrics for these processes separately. Some of the business performance metrics were specific to one of the four data request types, for example, average time to perform credentialing.

In this engagement, the provider data management processes were transformed into uniform, standard processes that were approved by all the representatives of the regional offices. An example of this was the practice, in some regional offices, of creating a data record for the physician before credential verification had been completed. As a result of the modeling exercise, this practice was enforced as a uniform policy.

A key factor in achieving consensus was the specification of the tasks and business results that permitted uniform performance metrics throughout all regions, yet allowed necessary variations of procedures within a task to conform to local government regulations and business practices. Even so, in certain cases, simply reaching the goal of consistent metrics required real changes in operations in certain regional offices. This process transformation was successful because the granularity of the tasks was defined based on the criterion that a task must add significant business value to the artifact. This level of granularity was fine enough to allow definition of consistent, meaningful business metrics, but coarse enough to permit necessary procedural variations.

As a result of the redesign effort, the standardized processes were approved by the regional representatives, and work on designing the tracking system was begun. At the same time, operations executives specified the business metrics that they wanted to see in the management display and reports. The project concluded when the users had been trained and the systems were deployed. Feedback indicates that operational managers now have the type of metric reporting they need to manage processes effectively.

The value delivered included clear identification of the only operational goal, namely “update provider

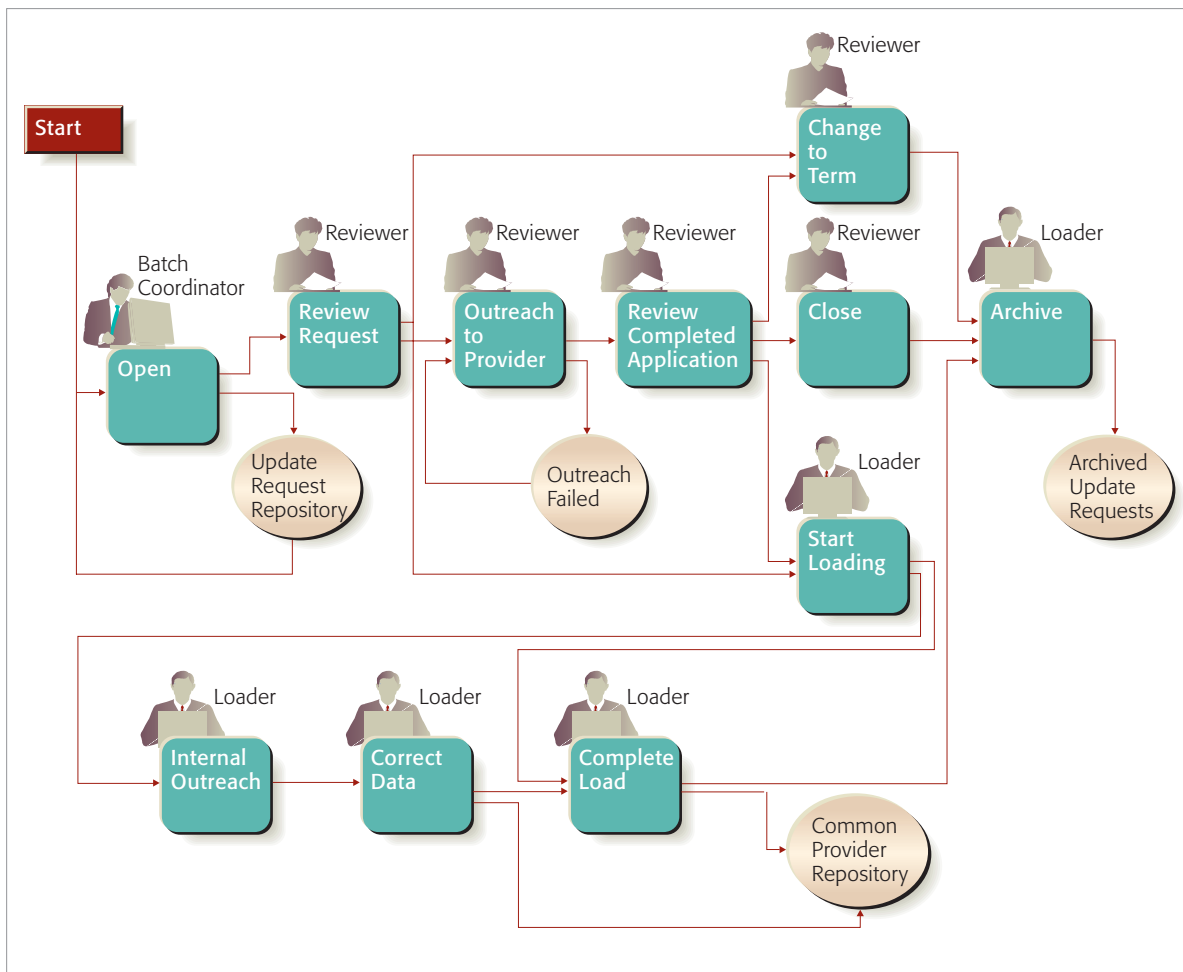


Figure 2
Operational model for update provider artifact

database from provided information.” This identification led to measurable states of progress toward the goal that were easily agreed upon and that removed organizational barriers by showing how individual tasks could be mapped onto existing job descriptions. Artifact-centered operational modeling expedited consolidation and standardization of processes by removing focus from detailed activities. As discussed previously, focusing on the operational model allowed us to expand our perspective from one that focused on the existing disparate processes within the business.

Flow forecasting for retail

In this engagement, the client was a fashion retailer with approximately 3000 stores in the United States and several hundred throughout the world. The

company managed several distribution centers in the United States. The client’s problem pertained to the processes used to capture the forecast for shipments in and out of distribution centers and into stores. The intent of this forecast was to provide an estimate, three months in advance, to the distribution center and store managers of the number of units of products to be shipped for a given week. This estimate was used for planning requests for resources to unload and load shipments from the trucks at distribution centers and stores.

High variability of forecasted values forced resource planners to overestimate the number of resources required to ensure smooth off-loading and shelving. The business intent of the transformation was to increase the quality of the forecast to enable more

accurate resource planning. The cost estimate indicated that a reduction of forecast variability by 5 percent could save approximately one million dollars in distribution costs.

The client realized that improvements in the analytics alone would not lead to the desired accuracies. The problem they were facing concerned the actual process of creating the forecast and the way in which the information created was managed. At the point of engagement, the entire process was entirely based on spreadsheets, which captured forecast, historical data, and guard rails (i.e., safety stock levels) to validate forecasts. The spreadsheets used for different parts of the forecasting process grew over time and were augmented with sophisticated macros. For example, the distribution forecast to predict flow in and out of distribution centers contained approximately 15 worksheets that were connected through macros. The spreadsheets were shared across the departments by use of a shared disk drive that contained almost 60 gigabytes of spreadsheet data. Obviously, searching for previous forecasts, analysis of patterns, and other functions were almost impossible.

Creating forecasts is a collaborative process involving a large number of stakeholders. The forecasts were typically created at the product level and aggregated at a division level. At the division level, a commitment was made to the consumers based on the forecast. This process of manually aggregating multiple spreadsheets into one was a tedious task that prevented the organization from providing forecasts on a weekly basis; instead, monthly forecasts were used.

In a two-day workshop with the client, we identified the end-to-end scenario and the different organizations that participated in the production of forecasts for distribution centers and stores. We focus here on the “distribution forecast,” that is, the prediction, three months in advance, of the quantity of each product that will flow out of each distribution center in a specific week.

We identified different types of forecasts as the key business artifacts. For the distribution forecast, we used the existing spreadsheet as the basis of discussion. When the client stakeholders were introduced to the ACOM approach, they started redesigning the spreadsheet to reduce its complexity

and to identify the information entities that were most important to them. They identified the following four key informational aspects:

1. *Distribution forecast for a given product*—This is the product of the forecasting process used by the distribution centers.
2. *Sales forecasts*—This is the key driver of the distribution forecast. The sales forecasts were provided by a forecasting application that was also used to set the targets for the sales teams, thus typically representing the upper bounds for what should be shipped out.
3. *Guard rails*—For example, the current inventory on hand, which is used to calculate the distribution forecast.
4. *Historical data for the previous year*—This assists in evaluating the pattern of flow and provides key input to the distribution forecast. Particularly interesting historical data are the “actuals,” that is, the actual shipments that occur during a given week. The *variance* of the forecast is the difference between the forecasted and the actual value for the past week. The actuals allow forecasters to account for adjustments that might have been made in the supply chain or for trend-related deviations.

The distribution forecast business artifact contains four parts for these four informational aspects. The next step was to identify the operations by which this artifact is updated by different stakeholders in the course of a week. In our analysis, we found the collaborative pattern shown in *Figure 3*. At the end of each week, a distribution forecast had to be committed to the distribution centers.

Each node in the hierarchy shown in the figure creates its own forecast, and each node that has children typically creates forecasts as an aggregation (or “roll-up”) from the forecasts of its child nodes. This represents only an example of a very small subsection of the overall merchandizing hierarchy. The actual hierarchy is much more complicated and not entirely consistent with respect to the number of layers in the graph. Some parts of the organization had only two layers from “leaf” departments to the point of the forecast commitment, whereas others had four.

In the following, we describe the operations from the perspective of a product department owner. The

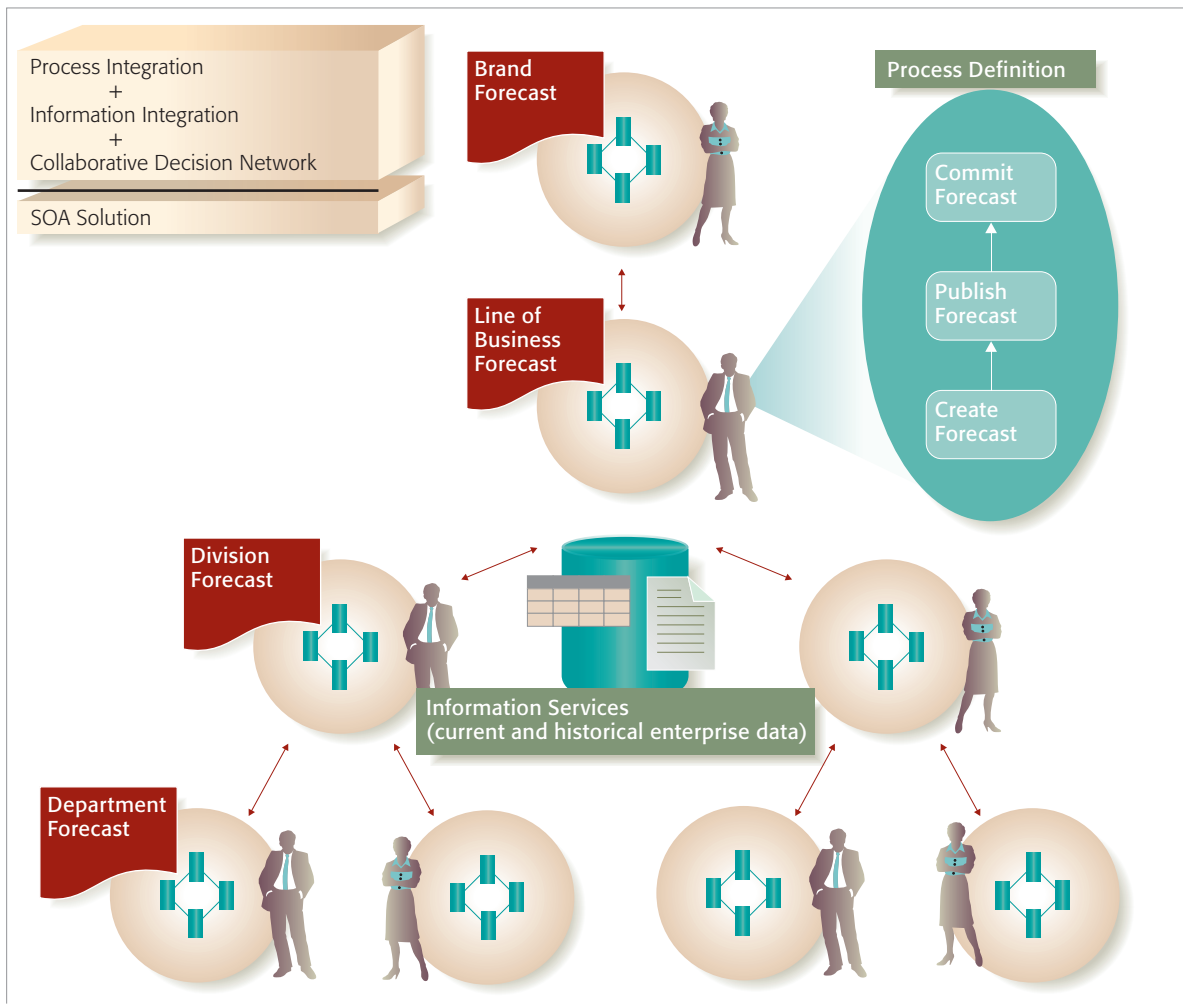


Figure 3
Collaborative flow forecasting process

first step is to create a new forecast that is based on the previous week's forecast. The stakeholder can use different guard rails, the sales drivers (i.e., factors which affect sales), and historical data to calculate a forecast estimate for the relevant metrics. The next step is to publish the forecast and make it available for review. A published forecast can be evaluated by the manager of the department. The manager might ask to rework the forecast but eventually will integrate it in his or her own forecast and publish it again. From the perspective of the product owner, the forecast is changing its state from *created* to *published* and eventually to *committed*. The stakeholder can publish the forecast and make it available for his or her manager to review. From the perspective of the department owner, the

work is completed only when the owner of the root-level product has committed the root-level (or brand) forecast to the distribution centers.

An instance of a distribution forecast artifact is thus created at each node of the merchandizing hierarchy. The instances of the forecasts are connected by means of arcs in the merchandizing hierarchy graph. When the root-level product owner commits the forecast, all instances that were involved in contributing to the root-level forecasts are considered as committed.

We modeled the operation of this collaborative forecasting process in the following way. The information model of the forecast artifact contains a

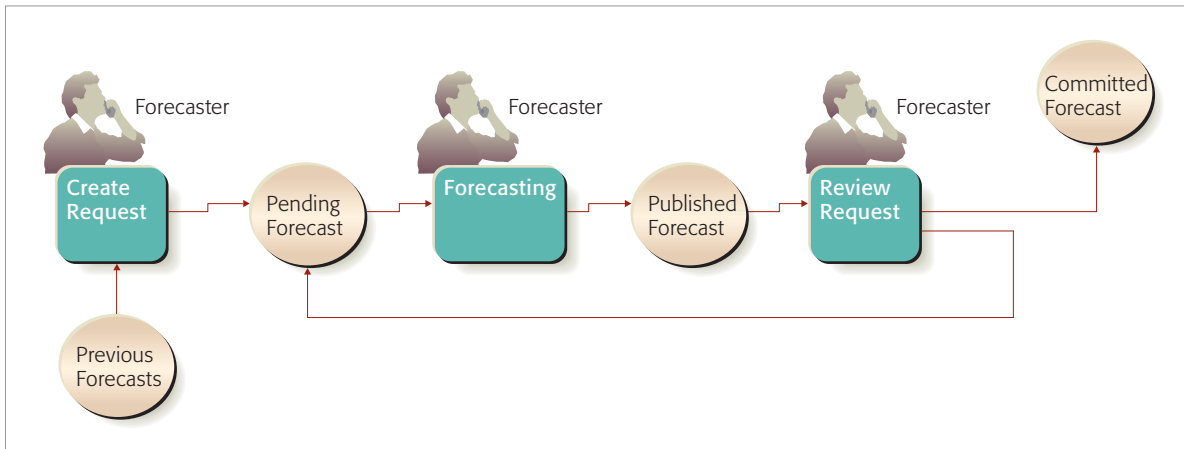


Figure 4
Operational model for distribution forecast artifact

unique identifier for a distribution forecast, metrics (e.g. sales drivers, guard rails, historical data, distribution forecast), a merchandizing identifier to associate an instance of the artifact with a node in the merchandizing hierarchy, and associations between instances of forecasts to manage forecast aggregations. The operational model for the distribution forecast business artifact is shown in *Figure 4*.

Each instance of a forecast at each node of the merchandizing hierarchy goes through the *created*, *published*, and *committed* states. Additionally, we identified business rules that manage the propagation of events across all associated instances. For example, if the root-level product division commits the distribution forecast that was created from lower-level forecasts, then all the lower-level forecasts are moved into the committed-forecast repository as well. The business rules were captured outside of the operational model and implemented in this engagement through a rules engine. The distribution forecast artifact model contains all the information needed to evaluate the business rules.

Our approach led to a redesign and significant simplification of the information architecture of the distribution forecast and consequently of all other artifacts. The operational perspective allowed us to design a system that would manage the information that was produced. By analyzing the requirements for business tasks, we identified several opportuni-

ties for automating activities that were previously tedious, manual, and error prone. Using the MDBT approach as described previously, we were able to implement a system that would allow for automated creation of aggregations, automated and context-sensitive integration of historical data, and dissemination of the distribution forecast to consumers through appropriate interfaces rather than through e-mailed spreadsheets. In our system design, we decided to retain the spreadsheet simply as an extension of the user interface. The data contained in the spreadsheet was retrieved from and used to update the information management system through Web Services interfaces.

Automation of distributed enterprise services

In this engagement, the client had developed a business strategy and a supporting functional architecture for an emerging independent business unit. Their business strategy was to provide site-customized support and IT services to an enterprise comprising a large number of geographically distributed small sites. Small sites were defined as those for which providing local, dedicated support resources would not be cost-effective. The strategic value proposition was to replace their part-time, ad hoc, underskilled, and informal support with a fully skilled, professionally managed service. Provided services included IT provisioning, installation, and maintenance, as well as general support. The operation was geared for end-to-end accountability and full life-cycle management.

The overhead costs of planning, resource allocation, and management dictated that a profitable scale for such a service provider started at thousands of sites. The target markets were franchise or chain operations such as food service, specialty retail, and hospitality providers. The strategy also encompassed enterprises such as drug trials or consulting projects where the operations (or their location) were transient or embedded in a larger organization. The client provided a range of services to the umbrella organization to contact, sell, and support the individual sites, including integrated marketing programs, deployments, and site scheduling.

The client was seeking to make progress on the IT implementation of its functional architecture. This progress was problematic for several reasons. Independently staffed legacy contracts inhibited integration of the program office. Unclear financial and execution responsibilities between the program office and project management caused redundant activity and over-management. The general lack of clarity contributed to significant organizational resistance.

We recommended that procedures for existing tools and organizational practices be documented and used as a basis for new process development. Additionally, the development of multiple transaction-recording tools that exposed local and legacy application data models was under way when we started.

The scope of the engagement was restricted to site installation services, as that was the primary focus of legacy contracts. The main business artifact that measured progress toward the key operational goal of site installation completion was called the *schedule* artifact. It contained planned and actual content of the installation project plan including any mid-stream modifications to the plan and the working documents transferred between tasks as part of the execution. Note that the term “schedule” derives from the fact that an outline project plan is generally attached as a schedule to the contract statement of work (SOW). The other significant artifacts identified were *site*, *customer*, and *delivery model*.

The *site* artifact tracked historical and present status, inventory, and service activity content. It also provided the basis for tracking the goal of

supporting the varying needs of property owners, as captured in the *customer* artifact. The *site* artifact was also a source of information needed for the creation of new *schedule* artifacts, thus contributing to full life-cycle management at each site.

The *delivery model* artifact contained the allowable range of equipment configurations, template project patterns for deployment, and maintenance and service catalogs. Instead of creating configurations and plans on a site-by-site basis from a large catalog of parts, the business strategy required that creating this content become part of the initial customer setup. Creation of a new *schedule* artifact involved the selection and instantiation of parameters from the preestablished patterns. An operational model for the *schedule* and related artifacts is shown in *Figure 5*.

Analysis of the artifacts identified led to significant new insights into the nature of the operation and value proposition of the business. Two ideas dominated the functional architecture—the need for a factory operation and the idea of “managing by exception.” The *schedule* artifact factory, in the form of the program office, was intended to deliver economies of scale by replacing dedicated project teams with a shared, fully utilized resource pool. Managing by exception meant that management resources were not spent tracking progress, but rather were invoked only to resolve exceptions to a predetermined plan.

Managing by exception led to the delivery portion of the *schedule* operation being divided into at least two phases: the binding of resources and completion dates to tasks in the delivery plan contained in the *schedule* artifact, followed by recording of actual checkpoints and intervention when necessary.

Analysis of the operational model revealed that the goal structure had not been factored properly. There were in fact two independent operational goals—obtaining agreement from the customer and obtaining agreement from the vendor (i.e., the people who actually installed equipment at sites)—that were being tracked by a single *schedule* artifact. This led to the discovery of the *task* artifact that recorded and tracked interactions with a vendor. For a particular *schedule*, delivery would begin when tasks are created and continue to administrative

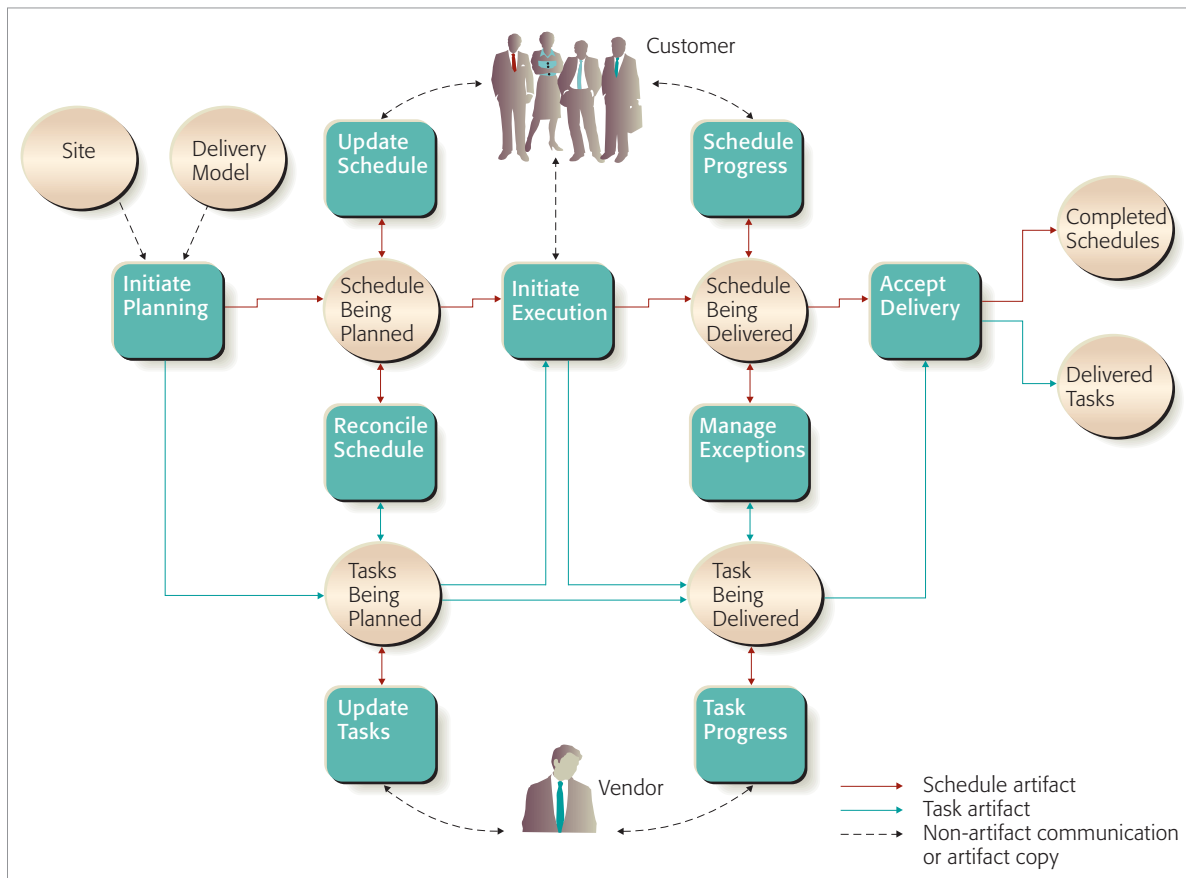


Figure 5
Operational model for schedule and task artifacts

completion when all of the tasks have been either completed or canceled.

The schedule-task artifact structure added concrete operational meaning to managing by exception. A vendor would agree to a start date and time, a completion date and time, and any intermediate measurement points that may be defined. At each of these times, it would be the responsibility of the vendor to notify the service provider that the milestone had occurred and the status of the task had been updated. The project plan would define which of these milestones are used to update schedule progress. Any other situation would be an exception, requiring management activity. The structure also provides a natural hierarchy for indicating the severity of the exception. These exceptions may require the management of changes to a single task, rescheduling of multiple tasks to

maintain sequence dependencies, or restructuring of the overall plan.

Operational separation of the customer and vendor interactions provided significant new insights into the relation of role structure within the program office and business goals as well as insights into the core “value add” of the service provider. One of the client objectives was the creation of a “single point of contact” for the customer. In practice, the complexity and volume of the work required roles within a multicustomer “factory” program office to be assigned along sequential segments of the operation: setup, planning, execution, and completion. An alternative that became available was that of separating roles by customer and vendor, providing better continuity and service to both.

The operational model revealed that the value being created was the coordination of customer expecta-

tions with vendor performance. Focusing on the value add of managing coordination can lead to increased economies of scale for both the vendor and service provider and thus result in lower costs and higher value to the customer.

The ACOM analysis provided clear articulation of the business value proposition. It led to better communication between business, operational, and IT support teams and highlighted the need for substantial attention to vendor management in the fully implemented business. A similar clarification of issues concerning roles and staffing was recognized as a potential future opportunity.

Operationally, simple replacement of process-based thinking with artifact-based thinking led to significant improvements. In some legacy accounts, the schedule artifact was already present as a physical folder. With the activity-based process view, folders tended to accumulate and significant efforts were necessary to find them. Simply adding repositories in the form of accessible “round tables” removed that frustration and extra effort. In some cases the administrative time required per schedule was reduced from over 20 hours to less than four hours. This is particularly significant because the 16 hours of administrative time saved was often matched with a similar amount of savings in expensive project management time. At the same time, artifact-based thinking radically improved the staff’s understanding of the operation.

The clarity in communication resulting from the ACOM approach extended into the executive domain as well. The use of operational diagrams to explain how the program office factory worked was very persuasive for customers who were skeptical that the services could be provided at the price and schedule proposed, enabling deals to be closed.

An implementation of the operational model using the MDBT toolkit was built and deployed. Independent estimates assessed the improvement in development effort to be 200 percent. Of course, this does not account for the process and application integration costs, which would have been required to implement the functional architecture.

Situations where ACOM is not appropriate

We have encountered situations where prospective clients have not been receptive to ACOM or where

ACOM has not been suitable for addressing the client’s problems. Sometimes clients demand improvements to existing processes in terms of the detailed steps that need to be performed. In these situations, clients are reluctant or even opposed to rethinking the operation to assess how well it is aligned to the overall goals. For such incremental process improvements, the ACOM approach offers little help. To put it simply, the improvements often lie in the design of what would be a single ACOM task, and for this the ACOM approach would constitute overkill.

Another common situation is one where the entire engagement has the scope of a business function that has been realized in a packaged application. Here the business operation and the IT realization are inextricably interwoven. Clients can be unwilling, and at times unable, to separate the business from the IT application, rendering the ACOM approach infeasible.

SOME OBSERVATIONS ON THE METHOD

The previous section presented a number of engagements that were selected to illustrate our experience with the artifact-centered operational-modeling approach. In this section, we revisit and expand on some of the key observations we made concerning the use of the ACOM approach.

Themes and lessons learned

In the following, we present some of the most useful lessons we learned from the engagements presented previously.

1. *Analysis of the operational model may bring to the fore problems inherent in the business transformation solution being considered*—A situation that we have encountered, in the case of distributed enterprise services as well as other engagements, is that the operational goal is too coarse; this leads to a very complex operational model. Often the solution is to split the operational goal into two or more parts. Naturally, this leads to the discovery of artifacts that were missed in the first attempt, and eventually to a simpler operation.
2. *Identification of the artifacts is valuable in itself*—Most of the significant business insights arise from artifact identification rather than from the details of the operational model. This has been

valuable even in very simple examples. A brief conversation with the owner of a small business providing Web design services, for example, resolved a persistent problem. Artifact analysis of the single process of designing a customer site suggested two artifacts: the *customer design* artifact, which tracked the customer interaction activities of the proprietor (including collaborative design), and the *implementation order* artifact, which was the assignment submitted to a subcontractor for the actual coding. Simply naming the artifacts led the proprietor to recognize that the problem consisted of the inclusion of the implementor in the collaborative design task. The distributed enterprise case described previously is an enterprise-scale example of the same insight.

It has been our experience that once identified, the artifacts usually seem obvious to the business domain experts. This suggests that the artifacts are, in some ways, reflective of the “real world” business situation. Thus ACOM makes artifacts and related insights available through analysis which, in turn, leads to more accurate IT requirements.

3. *Issues thought to require process- or IT-focused transformations often require rethinking of strategic considerations instead*—The distributed-enterprise and retail-forecasting cases are examples where clarity in the operational model led to a deeper understanding or facilitated positive change in the business strategy. In the first case, the contribution was the clarification of the source of value in the business, and in the latter case, the transformation resulted in dramatically shorter cycle times. The interpretation of artifacts as concrete representations of progress toward operational goals is what allows this contribution.

Strategic considerations, while not the immediate purpose of a transformation engagement, should be kept in mind and managed actively. The focus on operational goals and how they satisfy business goals provides a manageable linkage to strategy.

4. *Business performance measurement, monitoring, or management is integral to the ACOM approach*—Because artifacts record progress towards operational goals, monitoring is performed

by accessing and perhaps aggregating the content of the artifacts. A requirement of performing the access and aggregation activity is that it be encapsulated in a business task that works on some artifact. The operational goals of governance, monitoring, and management are represented by these artifacts.

Any ACOM effort almost automatically includes consideration of the governance of the processes being transformed. Another important lesson is that operational models apply to all of the activities of the business. Governance and management are valuable to the business only to the extent that they perform actions, that is, to the extent that they are representable by an operational model.

5. *Integration of legacy IT is dramatically simplified when viewed as an operational problem rather than an IT challenge*—Maintaining a focus on the business operational goals leads to low-resistance paths for the reuse and integration of legacy applications. MDBT tools provide a low-resistance path for new development. Moreover, this suggests that an important role of the MDBT toolkit is that of generating the coordination and integration layer on top of existing IT systems.

Reexamination of the transformation framework

In this section, we present a reexamination of the four-layer framework presented previously. Although the MDBT framework that was established and has evolved over the last few years to implement business goals is clearly effective, the lessons learned from its use suggest that more than a linear flow of models is involved. An alternative view is that ACOM provides a central model that is linked to the other models. The core idea is that ACOM represents a common model with elements that are interpretable in the process, IT-solution, and organization domains.

A popular, albeit limited, characterization of resources that are subject to business transformation is in terms of people, processes, and technology. Similarly, the software tooling community also focuses on people, processes, and technology when building business-level tools. Business transformation consultants similarly focus on organization, processes, and IT. The goal of an engagement is to understand the problems that a business is facing, to

analyze the business, and to develop recommendations for change. Usually, the consulting practices are specialized with respect to the change insights they can produce, for example, organizational change, process change, and IT change. Each practice relies on broad knowledge, techniques, and tools to create results for the customer. All of these depend on a variety of models. The interpretation of the operational model provides “anchor points” for further development in each domain and a mechanism for communication between domains.

For example, the organizational domain includes management of skill-based resources. An ACOM task represents one job done in a finite period of time by one or a few people. The task may specify the skills and quantities of resources needed, provide feedback that the task is not accomplishable with existing skills, or indicate that the granularity of the task is too small or too large. This then impacts the definitions of measurable progress and may result in negotiation.

The alignment of skill or capability with task granularity also occurs in the process domain. In present practice, processes are often depicted in role-based “swim lanes” (i.e., each swim lane contains activities performed by a specific role). The origin of the practice was the problem of work falling between roles as articulated by Rummler and Brache.⁵ The drawback of this practice is that it still relies on the assumption that the roles are responsible for understanding and moving the business goals forward, while being constrained by specific workflows defined in the overall process. The operational model provides a mechanism for modularizing the workflow activities, but explicitly introduces the recognition of progress toward a goal. Assigning a particular role to a collection of tasks makes explicit their value to the larger business, while still providing the skill-specific workflows necessary to accomplish them.

The relationship to IT is as described in the earlier discussion of IT tooling as exemplified by the MDBT toolkit. The connection of ACOM to services and service-oriented businesses is discussed in a companion paper.²²

CONCLUSIONS AND FUTURE WORK

The preceding examples strongly suggest that ACOM provides benefits beyond a simple improve-

ment to process-focused approaches. These benefits may include changing the discussion such that unarticulated relations between existing roles and local tools can be overcome, exposing structural similarities previously unrecognized in a focus on process-specific activities, or clarification of the strategic goal with traceable positive impact on required support structures, organizational options, operational measurement, and efficient IT implementation. This business model insight can directly impact the effectiveness of an IT implementation by using the service delivery model.²³ The ACOM focus on the operational goals of the business, embodied in the artifacts and tasks, provides the basis for alignment with and between strategic business intent, operational management, organizational and role structure, and appropriate functional-activity modules. The net result is a dramatic improvement in the ability to efficiently map organizational goals onto well-structured and effective IT implementations.

Counterexamples where ACOM may not provide significant value are engagements where the primary concern for improvement of processes is at the level of functional-implementation details. The functional focus of these engagements prevents connection to the operational goals of the business. As has been observed in many process reengineering efforts, these engagements are generally successful at improving the function which is focused on, but overall business improvements in these cases are difficult to quantify. Further, once developed and deployed, the resulting local technical improvements become simply another part of the complex interactions that characterize legacy application environments. Although the close affinity of ACOM for modular SOA solutions provides an implementation advantage even in these cases, for significant overall impact the SOA solutions need to be directly tied to the overall business and operational goal structures.

With respect to the larger question of business transformation, it is our position that even though an engagement usually starts with a specific focus (i.e., organization, process, or IT), producing significant business results requires taking into consideration one or more of the other aspects. For instance, if the solution to the customer’s business problem is the creation of a new job role, then the processes and IT systems may need to support this

new role. Consequently, we believe that creating a holistic description of what needs to be done to solve the business problem constitutes an essential starting point. We advocate the creation of an operational model as described in the previous section, starting with business goals. This holistic description can be related directly to the organization, process, and IT models. Thus the analysis performed for business transformation can affect and can also be affected by considerations that stem from organizational, process, and IT concerns.

The demonstrations of value described here and the lessons learned from them open several promising avenues for future work. Extensions that are under investigation include structuring and enhancing the interaction of ACOM with methods for capturing and prioritizing strategic aspects of the business, formalizing analytic methods both at design time⁹ and runtime for better visibility and control of operational processes, and extension of the MDBT transformations.

ACKNOWLEDGMENTS

Over the years, a number of people have contributed in different capacities to the research as well as to the engagements that this paper draws on. We thank our management for support and encouragement over the years, especially David Cohn, Kumar Bhaskaran, and Robert Guttman. We acknowledge the efforts of many people who have contributed to the design and realization of the MDBT toolkit; we especially thank Henry Chang, Pankaj Dhoolia, Terry Heath, Prabir Nandi, and Piyawadee (Noi) Sukaviriya for their efforts. Finally, we express our appreciation of the executives who believed in the approach, supported the engagements, and challenged us; we are particularly grateful to Steve Dickey, Claudia Girrbach, Caroline Miley, Robert Moulton, Deborah Putt, and Allan Smith.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Object Management Group, Inc. or Motorola, Inc. in the United States, other countries, or both.

CITED REFERENCES

1. A. Nigam and N. S. Caswell, "Business Artifacts: An Approach to Operational Specification," *IBM Systems Journal* **42**, No. 3, 428–445 (2003).

2. T. DeMarco, *Structured Analysis and System Specification*, Yourdon Inc., Englewood Cliffs, NJ (1978).
3. J. Paul Morrison, *Flow-Based-Programming: A New Approach to Application Development*, Van Nostrand Reinhold, New York (1994).
4. D. Dori, *Object-Process Methodology: A Holistic Systems Paradigm*, Springer-Verlag, New York (2002).
5. G. A. Rummler and A. P. Brache, *Improving Performance: How to Manage the White Space in the Organization Chart*, Jossey Bass, Wiley Press, Hoboken, NJ (1995).
6. S. L. Star and J. R. Griesemer, "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39," *Social Studies of Science*, **19**, No. 3, 387–420 (1989).
7. S. Kumaran, "The Model-Driven Enterprise," *Proceedings of the Global EAI (Enterprise Application Integration) Summit 2004*, Banff, Canada (2004), pp. 166–180.
8. K. Bhattacharya, R. Guttman, K. Lyman, F. F. Heath III, S. Kumaran, P. Nandi, F. Wu, P. Athma, C. Freiberg, L. Johannsen, and A. Staudt, "A Model-Driven Approach to Industrializing Discovery Processes in Pharmaceutical Research," *IBM Systems Journal* **44**, No. 1, 145–162 (2005).
9. R. Liu, K. Bhattacharya, and F. Y. Wu, "Modeling Business Contexture and Behavior Using Business Artifacts," *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE'07)*, *Lecture Notes in Computer Science*, 4495, Springer (2007), pp. 324–339.
10. *Model Driven Architecture*, Object Management Group, <http://www.omg.org/mda/>.
11. N. Caswell and A. Nigam, "Agility = Change + Coordination," *Proceedings of the IEEE CEC Workshop (2005)*, pp. 131–139.
12. R. S. Kaplan and D. P. Norton, "The Balanced Scorecard—Measures that Drive Performance," *Harvard Business Review*, 71–79 (January–February 1992).
13. *WebSphere Business Modeler*, IBM Corporation, <http://www.ibm.com/software/integration/wbimodeler/>.
14. *Unified Modeling Language*, Object Management Group, <http://www.uml.org/>.
15. *UML 2 State Machine Diagrams*, Agile Modeling, <http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm>.
16. *Rational Software Architect*, IBM Corporation, <http://www.ibm.com/software/awdtools/architect/swarchitect/index.html>.
17. D. Harel, "Statecharts: A Visual Formalism for Complex Systems," *Science of Computer Programming* **8**, No. 3, 231–274 (1987).
18. *Web Services Description Language (WSDL) 1.1*, World Wide Web Consortium, <http://www.w3.org/TR/wsdl>.
19. *Service Component Architecture*, IBM Corporation, <http://www.ibm.com/developerworks/library/specification/ws-sca/>.
20. B. Portier and F. Budinsky, *Introduction to Service Data Objects*, IBM Corporation, <http://www.ibm.com/developerworks/java/library/j-sdo/>.
21. P. Pande and L. Holpp, *What is Six Sigma?* McGraw-Hill, New York (2001).
22. N. Nayak, M. Linehan, A. Nigam, D. Marston, J.-J. Jeng, F. Y. Wu, D. Boullery, L. F. White, P. Nandi, and

J. L. C. Sanz, "Core Business Architecture for a Service-Oriented Enterprise," *IBM Systems Journal*, **46**, No. 4, 723–742 (2007, this issue).

23. S. Kumaran, P. Bishop, T. Chao, P. Dhoolia, P. Jain, R. Jaluka, H. Ludwig, A. Moyer, and A. Nigam, "Using a Model-Driven Transformational Approach and Service-Oriented Architecture for Service Delivery Management," *IBM Systems Journal* **46**, No. 3, 513–529 (2007).

Accepted for publication May 13, 2007.

Published online November 1, 2007.

Kamal Bhattacharya

IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Dr. Hawthorne, NY 10532 (kamalb@us.ibm.com). Dr. Bhattacharya leads a team of researchers in the area of business-driven IT management. His research interests are in business value-driven strategies for managing IT, model-driven development and service-oriented architecture. He has been leading and participating in several model-driven business transformation engagements over the past years and has received an Outstanding Technical Achievement Award for his work. Prior to joining IBM Research he worked in various development-oriented positions at IBM Global Services in Germany. He received a Ph.D. degree in theoretical physics from Goettingen University in 1999.

Nathan S. Caswell

IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (ncaswell@us.ibm.com). Dr. Caswell is a research staff member in the Business Informatics department with 30 years experience studying complex and dynamical systems. He joined the Watson Research Center in 1981 after earning a Ph.D. in Physics at the University of Chicago and holding an IBM Fellowship at the University of California at Berkeley. His initial research at IBM focused on material systems with complex dynamics, including multitrapping kinetics in long persistence CRT phosphors, lifetime of photolithographic CRT cathode materials, and effects of continuous trap distributions on TFT response time. Recent work has involved developing representational approaches to business engineering that enable analysis of the full complexity of dynamic evolution at a large economic scale. Dr. Caswell has provided leadership of projects in the health-care, retail, food-service, manufacturing, and business-transformation areas with both internal and external customers, to demonstrate practical application. He holds several patents and has authored a variety of journal articles.

Santhosh Kumaran

IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (sbk@us.ibm.com). Dr. Kumaran leads a team of researchers in the area of model-driven business integration. His research interest is in using formal models to explicitly define the structure and behavior of an enterprise and employing these models to integrate, monitor, analyze, and improve its performance.

Anil Nigam

IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (anigam@us.ibm.com). Dr. Nigam is presently a research staff member in the Business Informatics department. He joined the Watson Research Center in 1981 after earning a Ph.D. in Computer Science at the University of Rochester. His research at IBM has spanned a broad range of areas: VLSI

design systems, parallel processing architectures and database machines, logic programming and databases, knowledge representation, qualitative reasoning, operational business modeling, and business design. He has received research division awards, including a Research Commercialization award and an Engagement Excellence award, has published extensively, and holds a number of patents.

Frederick Y. Wu

IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. Dr. Wu is the manager of the Model-Driven Enterprise Solutions department at the Watson Research Center. He has worked in the area of electronic commerce and business integration for the past 12 years. Recently he has focused on business operation modeling and transformation of operation models to IT implementations. Dr. Wu has S.B., S.M., and Ph.D. degrees from the Massachusetts Institute of Technology. ■