

# Business artifacts: An approach to operational specification

by A. Nigam  
N. S. Caswell

Any business, no matter what physical goods or services it produces, relies on business records. It needs to record details of what it produces in terms of concrete information. Business artifacts are a mechanism to record this information in units that are concrete, identifiable, self-describing, and indivisible. We developed the concept of artifacts, or semantic objects, in the context of a technique for constructing formal yet intuitive operational descriptions of a business. This technique, called OpS (Operational Specification), was developed over the course of many business-transformation and business-process-integration engagements for use in IBM's internal processes as well as for use with customers. Business artifacts (or business records) are the basis for the factorization of knowledge that enables the OpS technique. In this paper we present a comprehensive discussion of business artifacts—what they are, how they are represented, and the role they play in operational business modeling. Unlike the more familiar and popular concept of business objects, business artifacts are pure instances rather than instances of a taxonomy of types. Consequently, the key operation on business artifacts is recognition rather than classification.

Over the last six years, we have developed a notation and a methodology useful for business process design at the business operational level. Our goal was

to create an operational modeling technique that would be not only amenable to business people and intuitive for business communications, but also based on a formal structure suitable for use in rigorous design and design analysis. Moreover, the core objective of the technique was to create a representation that business people could use to analyze, manage, and control their business operations from day to day. The first incarnation of this technique was called IFF (information, function, and flow); the basic premise was to factorize the knowledge about the actual operation of the business into orthogonal information, function, and flow components. As suggested by the name, the method represents information, function, and flow in a unified manner and at a uniform level consistent with business semantic use. The technique had considerable success with the persons who were actually accountable for pieces of the business. The representation has evolved with practical experience and feedback from users. Its current incarnation is called OpS (operational specification). After working with a number of customers across industry segments, both inside and outside IBM, and implementing the technique in several different tools within IBM, OpS has emerged as a useful business-level modeling representation.

A variety of tools and techniques have been used for business process modeling over the last two decades.<sup>1-7</sup> It is not our purpose to compare each of

©Copyright 2003 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 this paper must be obtained from the Editor.

these with the approach that we have been practicing and that is described in this paper. The approach we describe here was inspired by IDEF0 (first in a family of methods on Integrated Definition for Function Modeling),<sup>8</sup> particularly in the use of consumed and nonconsumed resources and the mapping of a process to the flow of individual artifacts through a series of distinct steps. OpS focuses on the flow of business information artifacts rather than of material artifacts and also differs in its formal structure. Our work also bears a superset relation to activity flow diagrams in Unified Modeling Language (UML\*\*).<sup>9</sup> Another influential representation is John Zachman's Information Systems Architecture,<sup>10</sup> a matrix framework into which independent representations can be classified on the basis of perspective and purpose, that is, a business owner's view of data description. The framework has grown significantly over the years, as new rows and columns have been added.

The purpose of the representation distinguishes our work; that is, the operational model is targeted at a business user and yet retains the formality needed for reasoning and, where applicable, automated implementation. Usually the process-level representation is a means of documenting business requirements, and its purpose is to provide specifications that drive implementations such as workflow systems. Other tools allow process models to be simulated. Our purpose is to create a description of a business that will be used by business people to manage the business. The models that are created contain enough information for the implementation teams, but the real value comes from the ownership, design, and analysis of the models by the business owners. The OpS approach is also significantly different in terms of how the business semantics are incorporated in the underlying formal model; however, this aspect will be presented in a forthcoming paper.

The focus of this paper is on the model domain, that is, business artifacts. Business artifacts correspond to the information components in IFF, now OpS, and are a cornerstone of operational thinking. Operational thinking in turn lies at the heart of building successful and usable operational models of a business in OpS. In this paper we describe what artifacts are, how to construct the life cycle of an artifact, and how interacting artifact life cycles can be used to construct operational models of the entire business. The value of artifacts and artifact-centered thinking lies in the creation of a representation that

is manageable, analyzable, and flexible from the perspective of a business person.

Even though business artifacts and the operations on them have formal characterizations, in this paper we focus on intuitive characterizations targeted at a business audience. We use the terms artifact and business artifact interchangeably. Throughout the paper we provide examples of real situations that we have encountered and what we have learned from them.

### What is an artifact?

An artifact is a concrete, identifiable, self-describing chunk of information that can be used by a business person to actually run a business. Sometimes we also refer to artifacts as business records, and some business people seem to prefer that terminology. In order to be useful for the actual running of the business, as opposed to an abstract model for analysis, artifacts have to be recognizable; that is, they have to contain information in one place. We have all heard the usual statement of frustration, "But I do not have all the information I need here." The requirements of being business-sensible and self-describing are driven very much from a business owner's cognitive perspective.

*Artifacts are taken to be the only explicit information contained in the business; that is, the set of business records represents the information content of the business.*

A particularly useful metaphor for an artifact is to think of it as a piece of paper. This metaphor brings out both the concreteness and identity of artifacts. An artifact is concrete and distinguishable from any other artifact, hence the identity. Two artifacts can differ only in identity; that is, they can contain exactly the same information. In the health-care insurance business, a visit to a doctor may result in a claim being filed with the insurance company. Sometimes, by mistake, both the patient and the doctor file a claim. These claims are identical in information; for example, they identify the patient, identify the doctor, list the diagnosis, describe the procedure that was done, and specify the charge for services. However, they are distinct, and the business operation has to deal with both by accepting one and rejecting the other as a duplicate.

In the rest of this section we will present the characteristics of artifacts from several different perspec-

tives. It would be useful to state at the outset what artifacts may appear to be but are not. Artifacts are not business objects—a technical notion from object-oriented techniques to represent business-level entities. A key difference is that artifacts are pure instances, hence the identity requirement, rather than instances of a given predefined class.

**An axiomatic view of artifacts.** *Business artifacts constitute concrete information chunks that the business creates and maintains. In other words, business artifacts provide the mechanism for information localization.*

The key properties of business artifacts are captured by the following axioms:

- [Info1] A business artifact consists of two parts: an enterprise-wide unique identity and self-describing content.
  - [Info 1.1] The content may be represented as nested name-value pairs.
- [Info2] The identity of a business artifact cannot be changed.
  - [Info 2.1] Consequently, an artifact cannot be split into two or more pieces, each of which has the same identity (although a different artifact with the same content but different identity can be created).
- [Info3] The content of a business artifact can be modified arbitrarily; that is, values can be modified and new name-value pairs can be added.
  - [Info3.1] Content can be copied from one artifact to another.
  - [Info3.2] New information from computation, external input, or any other source can be added to an artifact.

**Artifacts are familiar to the business community.** At first, the concept of artifacts appeared somewhat foreign to business process professionals. In contrast, operational business people, that is, those who are actually responsible for the operations of a unit, grasped the concept quite easily.

The task of artifact elicitation and elucidation begins with the question, “What does your unit produce?” This query can often yield a response that is the name of a product or a widget. The follow-on questions are: “How would one know that the widget has been produced?” or “In information terms, what are you in the business of processing?” or “What is the key business record that you produce?”

In our experience, after the dialog proceeds along these lines, the discussion quickly settles on the key

artifacts of the business. Many businesses identify a “customer order” as the key artifact that they are in the business of processing. As we will see later in the paper, this artifact provides a starting point for discovering the other artifacts in the business.

---

**Many businesses identify a  
"customer order" as the  
key artifact that they are in the  
business of processing.**

---

The following examples show that how quickly the artifacts become apparent often depends on the information infrastructure of the business. We worked on an engagement with a national provider of cafeteria services, that is, an institutional food service provider. Its customers spanned a large and diverse set—large and small companies, schools and universities, and hospitals. The local units recorded all the operational information on a series of paper forms that were designed by headquarters. Periodically, the completed forms were dispatched to the data entry organization that converted the information on the forms to electronic media. Then the usual corporate processes took over. For this business, it was quite straightforward to deduce the artifacts from the forms. The unit manager was able to talk about his business in terms of artifacts quite easily.

We encountered a somewhat different situation when working with the operator of a regional chain of quick service restaurants. Each restaurant had a network of computer workstations. One of these was used as a managerial workstation, and the rest were used as sales workstations. Also, each restaurant used a vendor software package that ran on all of the workstations. After a discussion on the purpose of artifacts, the restaurant manager, who had intimate knowledge of the operations, had no difficulty in spelling out the artifacts in his business. The manager was also able to distinguish between business artifacts, usually those that headquarters would use to measure and manage the restaurant, and just reports that were produced for day-to-day operations. The reports (for example, the people who are on vacation next week), were printed out on demand, used by the manager, and then discarded. Clearly these reports had no identity, did not need to be tracked, and hence were not to be thought of as artifacts.

In other situations we were faced with business people, somewhat more system-oriented, who were not able to locate the artifacts at first attempt. We found it useful to pick a candidate artifact and start with a hypothetical scenario. As the business person walked this artifact through the operation, details of which we will see later, the artifact started to become embellished. Then at some stage the “Aha!” happened (point of comprehension was reached) and was usually stated as, “I see it now; what we are really processing is X. Of course, X is the artifact that I am in the business of processing!”

The essential observation, based on a variety of engagements, is that artifacts are a concrete and natural building block of how business people think about their operations. Our experience validated, albeit empirically, the factorization that the IFF approach was based on: Artifacts separate the information component from the function and the flow components.

**How are artifacts represented?** At the outset we required artifacts to be self-describing. As mentioned earlier, this requirement was motivated by the need for recognition; that is, a business person should be able to look at an artifact and determine if he or she can work on it. Toward this end we used a recursive name-value pair notation to describe artifacts. To some extent this also stemmed from our interest in artificial intelligence, where name-value pairs had been used as components in knowledge representation long before the adoption of relational databases in the business world.<sup>11</sup> An example of a fully processed “guest check” artifact, in name-value pair notation, is provided below.

```
guest-check (
  ID 123
  context ()
  customer(number 3)
  store(ID(55) server (2))
  item (desc HamB price 2.57
    cooked "13:23 04/17/1998")
  delivered "13:26 04/17/1998"
  tax 0.33
  tender (total 2.90
    cash 20.00 coupon 1.00
    change 18.10)
)
```

In technical terms, artifacts correspond to complex objects in the database world. Complex objects were the relational database answer to dealing with more

structured data entities that spanned multiple relations.<sup>12</sup> Much like complex objects, artifacts have a formal algebra associated with them. The other aspect of artifacts, namely identity, shows up in the evolution from relational databases to object databases.<sup>13</sup> The essential addition to incorporate was identity.

Extensible Markup Language (XML) provides a modern syntax for representing complex objects. In hindsight, if XML had been available when we started the IFF work, we probably would have adopted XML as the syntax for artifacts. However, it must be noted that the artifact interpretation places a particular model-level interpretation on XML.

**Examples of artifacts.** We conclude this section with more examples of artifacts for different businesses, in order to further clarify the concept.

A popular example in many object-oriented modeling texts is that of a restaurant. The key artifact that a restaurant business processes is a “guest check.” Other artifacts, as we shall see later, are a “menu,” a “cash balance” (for keeping track of cash), and a “material balance” (for keeping track of material inventory).

Let us now consider a different business—letter and package delivery. It is a complex business, and the models vary by provider; for example, Federal Express has an approach distinct from that used by United Parcel Service. However, the key artifact, around which the entire business operation revolves, is what Federal Express calls the “air bill.” The entire business is focused on processing air bill artifacts from creation to completion.

Artifact identification is as applicable to a specific piece of a business as it is to the entire business. Thus, if we were to consider the human resources part of a business, its key artifact is an “employee record.” All the other artifacts within human resources are dedicated to support the processing of the employee record.

Finally, let us consider an example from the publishing industry—the publication of technical journals. A key part of the business is to have submissions refereed by multiple experts so as to enable the editor to make a decision about publication. For the review piece of the business, the key artifact is a “referee report.” The editor creates one artifact for each of the referees. Each referee completes his



or her review after examining the contents of the submission. The original manuscript (an artifact) remains with the editor; the referees access a copy for their perusal. The editor examines the completed referee report artifacts, makes a final decision, and records it on an “editor’s report” artifact.

### Artifact life cycle

As shown in the preceding examples, artifact processing is a way to describe the operations of a business. An artifact life cycle captures the end-to-end processing of a specific artifact, from creation to com-

---

**Artifact processing  
is a way to describe  
the operations  
of a business.**

---

pletion and archiving. In describing how artifacts are processed, we introduce two critical OpS constructs: tasks and repositories. Just as artifacts constitute localization of information, tasks provide a localization of function. Repositories provide a means for archiving artifacts, both for those in progress and for those that have been completed. In this section, we illustrate the concepts with a detailed example of the end-to-end processing of a guest check in a restaurant.

**An axiomatic view of how artifacts are processed.** Earlier we presented an axiomatic view of artifacts. In this section we present an axiomatic view of the function and flow aspects of artifact processing, thereby completing the IFF (Information, Function, and Flow) factorization of business operations (now within OpS). The function ([Fun]) axioms deal with how information may be added to artifacts, and the flow ([Flow]) axioms deal with the transport of artifacts across functional units of the business.

*Places provide the physical context (i.e., localization of function) for the creation, transformation, and archiving of business artifacts. In other words, places enable the business to bring together its human and system resources to accomplish the goal of transforming artifacts.*

There are two kinds of places, those where changes can be made to artifacts and those where artifacts

can be placed to await future processing, if any. From here on, we refer to places as “business tasks” and “repositories” respectively.

Following are the function axioms:

- [Fun1] The goal of a business task is to perform an action and to record the outcome on one or more business artifacts that are in its possession.
- [Fun2] A business task is activated either by the receipt of a business artifact or through explicit triggering mechanisms (timer or direct activation by a person).
- [Fun3] A business task transforms one or more business artifacts that are resident in a place. This transformation consists of adding content to or modifying content of an artifact by using information available in the artifacts that are present in the business task. Multiple business artifacts can be resident in a business task, and content can be arbitrarily exchanged between these business artifacts.
- [Fun4] Business artifacts enter a business task in one of two ways. They are either spontaneously received or explicitly requested and obtained from some appropriate source.
- [Fun5] After a business task completes its processing, it ejects all artifacts that are resident within it. A business task has no residual information; that is, all artifacts are either sent out or are discarded.
- [Fun6] Artifacts can be inserted into a repository.
- [Fun7] A repository can respond to requests for artifacts or for artifact content; however, it cannot send out artifacts spontaneously.

*Business tasks and repositories can be connected through flow connectors which may be viewed as transport pipes. Through these pipes, artifacts or artifact content can be transmitted from one place to another.*

The semantics of these pipes are captured by the following ([Flow]) axioms:

- [Flow1] A flow connector is a directed connector between a “fromPlace” and a “toPlace.”
- [Flow2] A flow connector ensures reliable transmission of artifacts; that is, after an artifact is inserted into one end of the connector, it is ensured to be received unchanged at the other end.
- [Flow3] A flow connector, when connecting a business task to a repository, provides a reliable request-response style of communication. Consequently, a business task that sends a request to a repository is ensured to receive one or more ar-

tifacts (or artifact content) or a “none found” indication.

**Life cycle of a guest check.** The goal of a restaurant, much like any other business, is to generate revenue through the sales process. Of course, other processes in the business have to ensure that the customers are satisfied and that profit can be realized from the sales. We first concentrate on the “generate sales” process in a restaurant where dinner is being served.

The artifact that is processed during the generate sales process is a guest check. Let us follow the guest check through the life cycle of the process with a simple narrative.

As a customer walks into the restaurant, he or she is greeted and seated at an available table. The greeter creates a guest check and records the table number, number of people in the party, and other details on the guest check. (If a table is not available, and the customer is willing to wait at the bar, a guest check is created, and the customer is seated at the bar.)

The waiter presents the customer with the menu (artifact) and the “daily specials” (artifact) and records any initial items ordered (typically drinks).

After the customer has made his or her selections, the waiter records them on the guest check along with any special preferences indicated by the customer.

The waiter now creates a “kitchen order” (another artifact) that contains details of the items the customer has chosen along with any special preferences. On the guest check, the waiter records that the kitchen has been informed.

The kitchen works on the kitchen order, produces the appropriate dishes, and, when done, records this fact on the kitchen order.

The food is served to the customer, who proceeds with the dinner. (Any changes and discrepancies are handled by creating new kitchen orders.)

When the customer requests the check, the guest check is totaled and given to the customer.

When the customer is ready to pay, the guest check is tendered along with cash provided by the customer.

The monetary transaction is recorded in the cash balance (another artifact).

A copy of the tendered guest check along with any balance of cash is given to the customer, and the dining experience is complete (assuming that the restaurant is a “cash sales only” establishment). This is the completion of an execution of the generate sales process.

This example is intended to be illustrative rather than exhaustive; consequently, we have not tried to include all possible situations that can occur. The example brings out the fact that the generate sales process can be described by capturing the life cycle of the main business artifact, that is, the guest check. Other artifacts that come into play are:

- Menu and daily specials, which are consulted but not modified.
- Kitchen order, which is created by the waiter and processed by the kitchen staff.
- Cash balance, which is processed when the guest check is tendered.

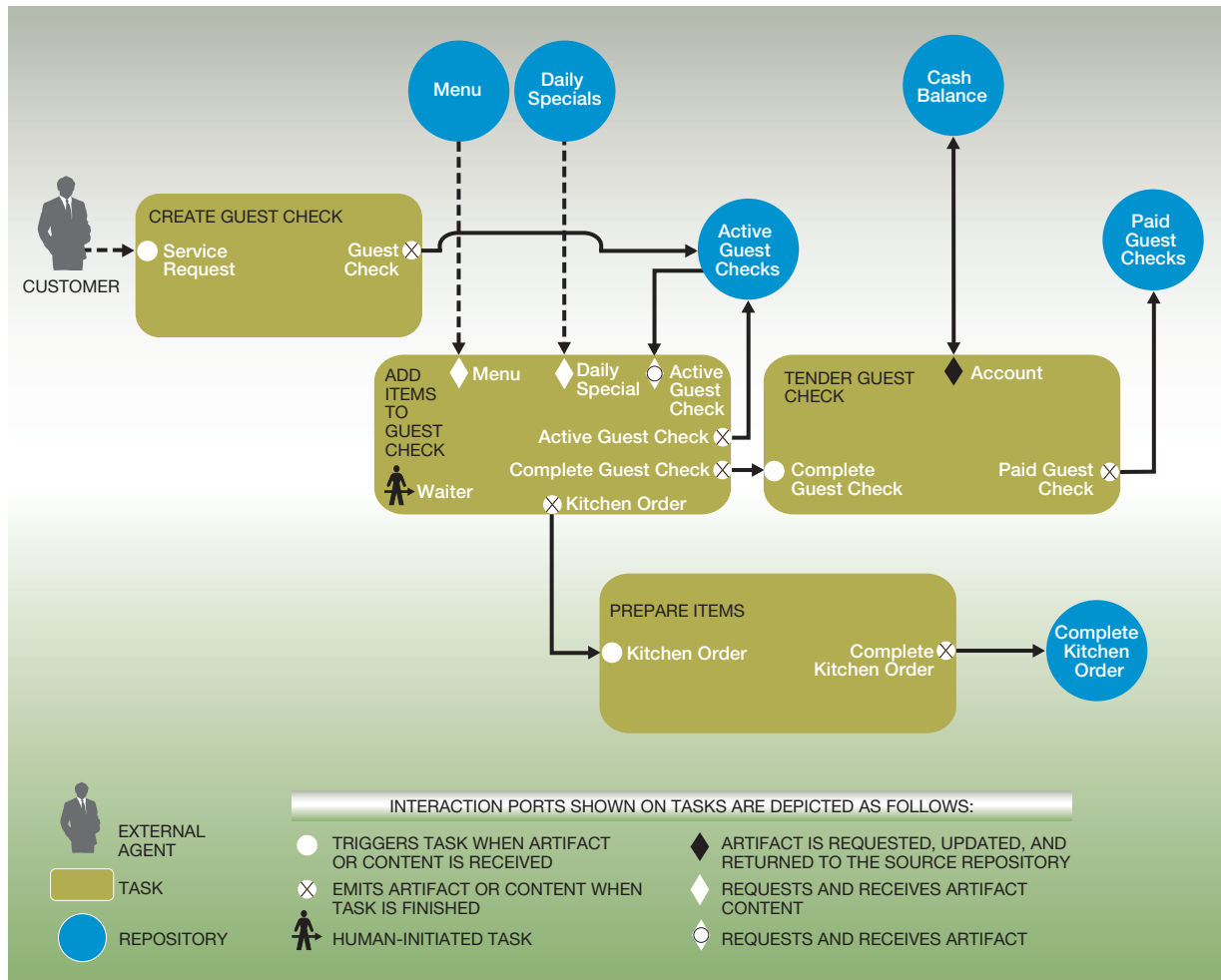
**An operational model in OpS.** We now examine an OpS model of the restaurant business that was just described. The model is an interconnection, or a graph, of the tasks and repositories that capture how different artifacts are manipulated in the course of the business. For the guest check and the kitchen order, the entire life cycle of the artifacts is captured in the model. As mentioned earlier, this model is one of many different kinds of restaurant operations. Our objective here is to establish that the intuitive semantics behind the two key OpS elements, business tasks and repositories, are consistent with the axioms presented earlier, and how these elements relate to the central concept of artifact.

Each task in Figure 1 has the following behavior:

- Some mechanism triggers the task (e.g., arrival of an artifact).
- As part of its action, the task modifies artifacts that are in its possession or creates new artifacts, or does both.
- The last step in the execution of the task is to send out the artifacts that were in the task’s possession or were created by it.

The behavior of a repository is even simpler. Any artifact emitted by a task can be transported into a repository. After the artifact is in the repository, it

Figure 1 Restaurant business: Life cycle of the guest check artifact



will stay there until it is explicitly requested by some task. While an artifact is in a repository, its content can be made available upon request, without removing the artifact from the repository. This is akin to providing a “read-only copy” of the artifact.

Tasks and repositories interact through two basic mechanisms: transport and request. In transport, an initiating task has an artifact and initiates transfer of it to another task or repository. In a request interaction, the initiating task sends a descriptor to a repository and receives any matching artifacts. Both these mechanisms can transfer either the artifact or the information content of the artifact. External messages, such as the report on the people who are on

vacation next week mentioned previously, are examples of content transfer.

The OpS diagram uses a variety of icons, the meaning of which will be brought out by the task descriptions. Let us elaborate first on the tasks that work on the guest check artifact:

- **Create guest check:** This task is initiated when a customer enters the restaurant conveying information to the greeter that he or she would like to be served. This task creates a unique instance of the artifact. Information added to the artifact is table number, number of people in the party, and the

name of the waiter. The guest check is placed into the Active Guest Checks repository.

- Add items to guest check: This task is initiated by the waiter when the waiter visits the customer table. The correct guest check is accessed and items, along with diner preferences, are added. In order to accomplish the task, the waiter consults (but does not modify) two artifacts, the menu and the daily specials. If items that need to be prepared are ordered, the task creates a kitchen order and copies the items and preferences from the guest order onto this new artifact. Both the kitchen order and the guest check are emitted from the task, and the task is complete. This task may be repeated as necessary until the customer is finished and the waiter marks the guest check as complete.
- Tender guest check: The task is triggered by the receipt of a completed guest check. The task receives payment and marks the guest check as paid. It requests and obtains the cash balance artifact, modifies it to reflect the sale amount, and returns it to the cash balance repository (again, the restaurant is assumed to be a “cash only” establishment to keep the example simple). Both artifacts are emitted when the task is complete.

The processing of the kitchen order has been packaged into a single task, namely *prepare items*. This task is triggered when the kitchen receives a kitchen order. When the food is ready and the waiter has been notified, the kitchen order is marked as completed and is emitted from the task.

The OpS model in the example employs several different repositories:

- Menu contains the menu artifact. The model shown in Figure 1 is incomplete because we do not show the tasks that create and maintain the menu artifact.
- Daily Specials contains the daily special artifacts; again the creation and maintenance of this artifact is outside the scope of the example.
- Active Guest Checks contains guest check artifacts pertaining to customers who are dining.
- Paid Guest Checks contains guest check artifacts that have been processed completely; that is, they have one or more items, have been tendered, and are marked as paid.
- Cash Balance contains an artifact on which cash-on-hand is recorded. This artifact is created and maintained by the “manage money” process, which is not covered in the model.

- Complete Kitchen Orders is where the fully processed kitchen orders are placed.

So far we have illustrated, through a simple example, how to build an operational model within the semantics specified by the axioms. Each task modifies artifacts in its possession or creates new artifacts, in accordance with the place axioms. Artifacts travel along connectors between tasks according to the flow axioms. The overall graph of tasks and repositories constitutes the operational model of the business.

**Autonomy of tasks.** A task specifies the business analog of a transaction in which one or more artifacts are modified atomically. Perhaps the easiest way to visualize tasks is to use a human metaphor; that is, assume that a person performs the task. Most tasks are amenable to this way of thinking. The granularity of a task is the amount of function that is packaged into it. This is a business decision which should satisfy the heuristic that it is “one job completed at one time.” An example is the addition of credit check information to an order. This task will be part of a larger “order acceptance” process that consists of several tasks and may contain detailed instructions such as “call Credit Bureau, identify yourself, and identify customer.”

It is important to design each task in relative isolation. For instance, a task may not depend on the knowledge of tasks before it and after it in the life cycle. All that a task has to go by is the information that is present in the artifact that it receives. Recognition as well as the subsequent processing is driven by the information explicitly present in the artifact. As mentioned earlier, if a task cannot recognize an artifact, it will send out the artifact unchanged, on a designated port.

Autonomy of tasks, as we will see later in greater detail, allows for localized transformations in the business. Let us illustrate this point briefly with the restaurant example that we presented earlier. We assume that the restaurant now wishes to expand its business to include take-out orders. This expansion can be accomplished by introducing a task, called *Add Items to Take-Out Guest Check*, that will edit the guest check while interacting with the customer over the phone. The rest of the processing remains the same. Also, a *Tender Take-Out Guest Check* task will be activated as the customer arrives at the restaurant to pick up the order.



## Interacting life cycles

In the course of describing the life cycle of the key artifact, many tasks along the way may need to consult or modify other artifacts in the business. Because the other artifacts have their own life cycles, these dependencies represent two sorts of life cycle interactions. Reference to another artifact by a task creates a dependency. The need to modify another artifact is a stronger interaction in which both artifacts are located in the same task. In the modification case, the artifact life cycles intersect because the task in question is part of the life cycle of both artifacts. The restaurant has examples of both types of interactions. The processing of a guest check refers to menu and modifies cash balance. A complete operational model of the enterprise consists of the life cycles of all the artifacts within the control of the business. These interacting life cycles capture the dependencies between the different artifacts of the business.

Our approach to constructing an enterprise-wide operational model has been to iterate the following steps:

1. Pick the “money” (or key) artifacts and construct the life cycles.<sup>14</sup>
2. Create a candidate list of all the artifacts that were either consulted or modified in constructing the “money” artifact life cycle.
3. Repeat the following until there are no more artifacts on the candidate list:
  - (a) Take an artifact X out of the candidate list.
  - (b) Construct the life cycle of X.
  - (c) Add any newly discovered artifacts to the candidate list.

The basic premise is quite simple. For each artifact in the business there must be a set of operations, hence life cycle, for maintaining it. According to this reasoning, the enterprise-wide view of the restaurant in our earlier example will need operational processes for maintaining the following artifacts:

- Menu and daily specials: what items to offer to customers, the prices to charge, and the recipes to use to prepare them
- Cash balance: all financial transactions and reporting, that is, accounting for all the money in the unit

Additional artifacts that will show up as we expand the model are:

- Employee record: all details pertaining to an employee since date of hire
- Time sheets: work record for each employee
- Labor plan: scheduling of employees over the plan period
- Vendor order: for food materials and supplies
- Material balance: all inventory-related processing (ordering, receiving, physical inventory, etc.)

In one of our engagements with a food service customer, this requirement was stated quite concisely by one of the executives: “In addition to the sales process, which you have described as the guest check processing, I need to do three more things to take care of my business; manage people, manage material, and manage cash . . .” His remark succinctly captures what an operational model needs to cover in order to be useful for analyzing and managing the entire business. Figure 2 is an OpS representation of the entire restaurant business.

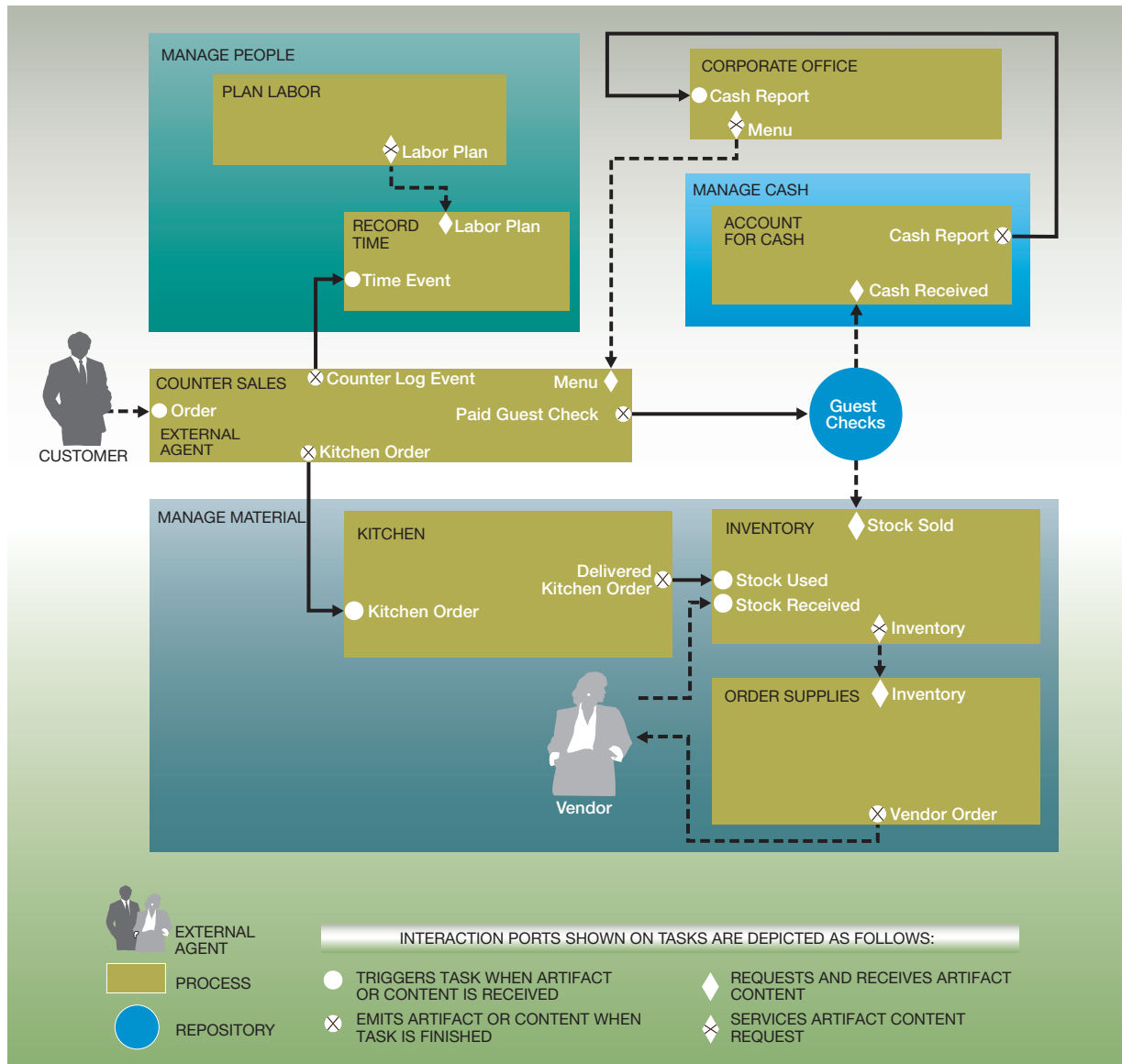
**Operationally centered view of the business.** It is our belief that an operationally centered view of a business, captured as interacting life cycles of artifacts, provides a rich representation for managing and analyzing the business. In this section we touch on some of the kinds of analyses that are enabled, namely strategy, organization, and application.

The operational model allows the capture of the following kinds of information about tasks, as applicable:

- *Strategic intent*, the specific strategic reasons why this task is being performed
- *Role or organization* that performs this task
- *System or application* that automates or supports the task

Since strategy is an area that is too broad and general, we focus on strategy as statements about the business. Often the connection between strategy, the operations, and the systems is informal. Consequently, it is hard to ensure the connections between a strategic initiative, its operational implications, and its subsequent realization in the information technology (IT) systems of a company. An operationally centered view allows the strategy statements to be cast in terms of the operational elements, for example, decisions about artifacts and their processing. Specific strategy elements can be attached to the basic modeling elements, tasks, and repositories as attributes. From a tool perspective, this allows a bus-

Figure 2 An aggregated model of the restaurant business



iness leader to highlight all the tasks that support a given strategy statement. Other aspects, such as what artifacts to focus on, are discussed in the next section.

Metrics are closely related to strategy because some aspects of strategy are in fact statements about the metrics. In most cases, the metrics will be derivable

from the information that is recorded on the artifacts of the business. Sometimes a special artifact needs to be created to deal with a metric that pertains to aggregate information, for example, widgets processed per hour. Similarly, the actual task of recording the metric is either incorporated in the function of specific tasks, or additional “measurement” tasks are added to the operational model. This ap-

proach can be used to incorporate KPIs (Key Process Indicators) into OpS models.

An organizational view is easy to extract from the operational model. One can easily transform the operational model graph into another graph where the nodes are the organizational units and the connections are derived from the transport edges. In an OpS diagram, such a view can be shown by drawing a box around all the tasks that are performed within the same organizational unit, for example, a department. Arranging the organizational units in a vertical stack with their operational content visible produces the familiar “swim-lane” view. However arranged, from such a view it becomes clear what artifacts are being sent between organizations, thereby giving a concrete depiction of the collaboration between the organizations. For example, analysis of this view might reveal that a specific artifact crosses the organization boundaries multiple times over its life cycle. This observation may suggest a transformation where certain tasks are moved from one organization to another.

The mechanics of creating an application-centered view are the same as those mentioned above; the only difference is the basis of the aggregation. The resulting graph may be seen as a kind of architecture diagram. The real benefit of such a view is that a business person can see why an application needs to be interfaced with another; for example, application A adds some information to the “my” artifact, and then application B needs to add more information to the same artifact. Analysis of the application view supports modification of the applications to match the business operation and vice versa.

The operational model can also be the basis for building an automation system from scratch. The operational model when fully developed has all the business information needed for creating an automation system. This is exactly what we did for a health-care insurance client that was transforming itself into a managed care company. The power of this approach is the correspondence between business tasks and the application programs that automate them. Consequently, changing the IT implementation as the business changes becomes very efficient; the business person can tell us very precisely what change needs to be made at the OpS level.

A more contemporary attempt to develop automation is presently underway. The business-level model is the operational model that we have discussed so far. A prototype has been developed to compile the

business-level model directly to a middleware or Web services skeleton.

### Business value of artifact thinking

In this section we draw from experience in using artifact-centered thinking and operational modeling. We focus on the business value realized in three main areas: flexibility in representation, ability to analyze change, and ability to manage implementation systems that support the business. The business value shown here is enabled by specific characteristics of operational modeling that, in turn, depend on artifact-centered thinking.

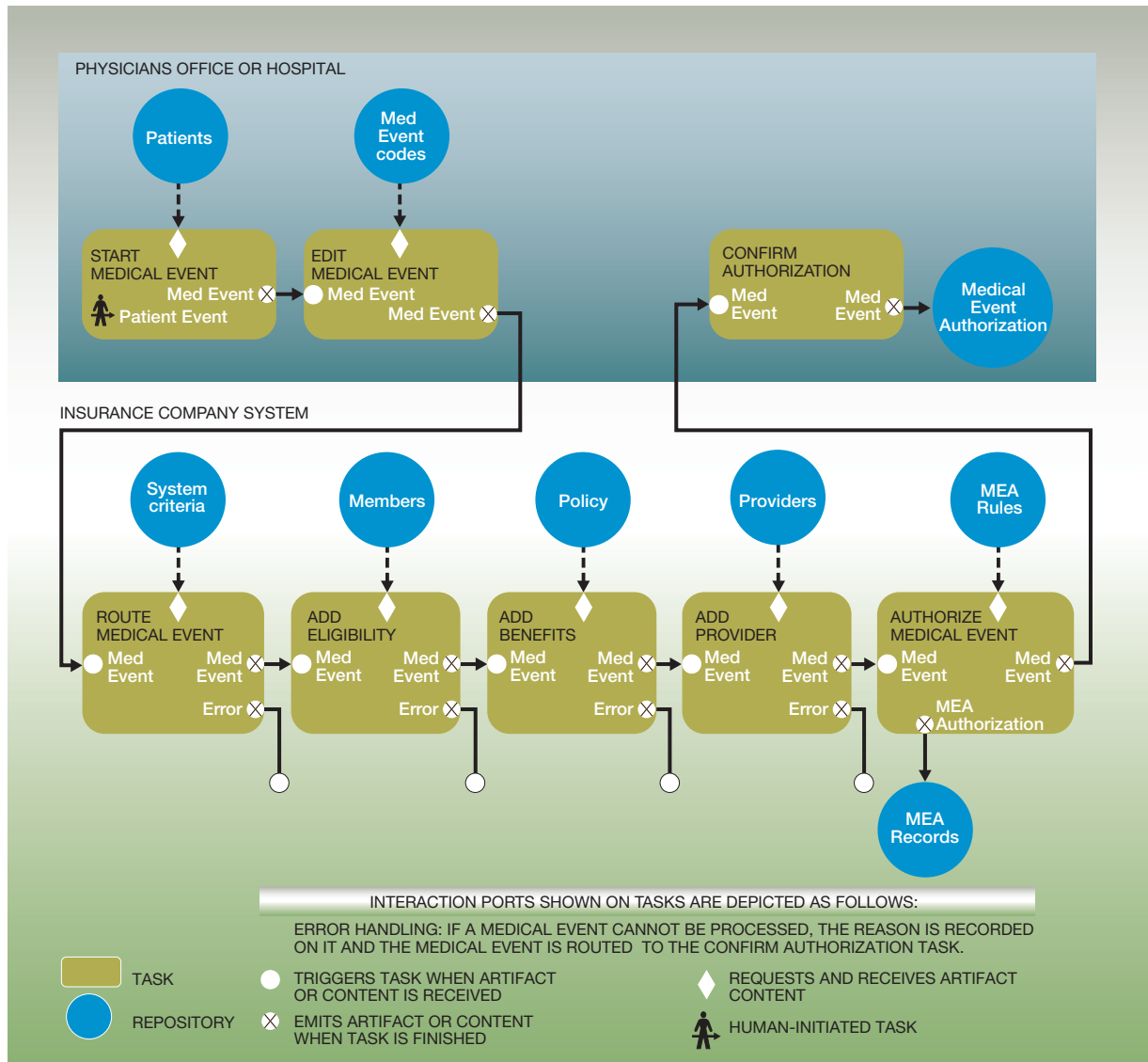
Three examples illustrate the application of artifact thinking to business organizations. Figure 3 and Figure 4 show this thinking applied to analysis of change in a health-care organization and a supermarket-chain organization, respectively. Figure 5 shows an extension to the health-care operational model.

A basic characteristic is that operational models provide an integrated analytic view of the whole business (or as much scope of a business as is desired). In the earlier discussion of interacting artifact life cycles, we saw how it is possible to describe the operation of an entire business in a single integrated representation. Integrating the operational view depends on representing information along with activity in the same model. Uniting information (data) and function (process) in the integrated representation means that all information and functional dependencies across the scope of the business can be analyzed and validated during design and transformation.

The ability to create an integrated model, we believe, stems from the fact that the model is constructed from well-localized, independent atomic elements. Choosing artifacts as the information granularity and tasks as the activity granularity means that the operational behavior of the modeled business is defined only by the behavior of the tasks and the artifact-based interactions between them. Tasks or aggregates of tasks, at any scale, can be replaced without operational behavioral change, as long as the artifact interactions remain unchanged. Conversely, major changes in behavior may be produced by reorganizing the interactions of a set of tasks.

**Flexible business representation.** Artifacts are pure instances (recall the piece-of-paper metaphor we used earlier), so it is conceptually and technically easy

Figure 3 Medical event authorization in a managed health-care business



for a business to gather additional information in its artifacts. This is equivalent to scribbling notes on the metaphorical paper. Because tasks work by recognizing artifacts that contain the specific pieces of information that they depend on, additional information does not get in the way. While we were working with a regional quick service restaurant chain, a manager pointed out that in an ideal world the organization would like to record information such as

weather on the guest check because he believed that weather was one of the predictors of traffic in the restaurant. Because the artifact is a business construct, and not a technical data model, such enhancements are a business decision. At some later stage in the business, there might be an operation to examine guest checks in terms of weather information. This particular operation would recognize a guest check as a weather report and not be affected by ad-

ditional details that might be in the artifact. Thus, a single artifact can be used to serve different business purposes, as desired.

A common complaint from business owners and operators is that all the information they need is not present in one place. The goal of artifacts is to provide precisely such a business-sensible amount of information that the business can modify and control.

There appears to be a close correspondence between some artifacts and the “dimensions” of a dimensional data warehouse.<sup>15</sup> We believe that artifacts allow the business to control the “dimensions” quite explicitly, rather than after the fact, as in the case of an information warehouse.

**Analyzing change.** Artifact thinking plays a very useful role, especially in situations where a business is in the process of transforming itself radically. Our health-care-insurance client was in the process of transforming itself from being a provider of indemnity insurance to a managed care company. Moreover, the business pressures were such that rapid transformation was needed. In their as-is business, there were two key artifacts, a “claim” and a “medical event authorization” (MEA).

The original claim essentially specified the information about the patient, the provider, the diagnoses, the services that were provided, and the charges for the services. The claim was adjudicated to make the financial determination, that is, what the insurance would pay, what the provider would absorb, and the patient’s responsibility.

An MEA, in contrast, recorded the care management determination; for example, the referral was appropriate, the procedure was medically necessary, or the length of stay in the hospital was within the norm. Claim adjudication, in some cases, required that the MEA be present, for example, to ensure that services being billed in the claim had been authorized. In their existing business, matching the two artifacts was difficult and imprecise; the consequence was more manual intervention and thus higher costs.

As the client started to define its to-be business, it had to decide what its key artifact would be. A new artifact, “encounter,” was proposed. An encounter recorded an interaction between a patient and a provider, and consisted of two important pieces: order and claim. Order was the set of instructions for the

treatment, including services that were to be provided. After a great deal of debate, it was decided that because the new business had to do with care management, order was the more essential piece. In other words, there could be encounters for which there was nothing to bill, and yet they were medically important. For instance, if you call a doctor and are given some instructions, an encounter has occurred. Collections of encounters could be organized into “episodes” of care, and that was their new business. This evolved into MEA processing as shown in Figure 3.

The basic lesson learned was that the choice of the key artifact may require extended and animated discussion. This choice helped to articulate what business the client wanted to be in.

In a different engagement, we came across the same situation. A large supermarket chain was considering a technology-based program for their loyal customers. These loyal customers were to be given wireless-enabled and suitably enhanced personal digital assistants (PDAs) that they would use to create their shopping lists. Upon entering the store, the customer would turn on his or her PDA, the device would interact with the wireless LAN (local-area network) within the store, and the customer would be recognized. The in-store computer would access the shopping list that was on the PDA. In return the store would provide a variety of services and offers on the customer’s PDA to make the shopping experience enjoyable and profitable. Our first question was about the key artifact.

On the first iteration, the supermarket chain believed the artifact would have to be the order. The shopping list would somehow be used to construct the order. After some discussion, it became clear that an order was quite different from a shopping list. Orders were precise; for example, they had the stock-keeping unit (SKU) for each item that was being ordered. Shopping lists were essentially imprecise and often just memory aids for the shopper. Therefore, what was the key artifact that the loyalty program was going to process? Finally, the chain realized that the artifact that would be processed in the scope of this project was a “customer profile!” For each visit, both access to the shopping list upon entry and access to the eventual order that was created at checkout would add information to the customer profile. The better the customer profile, the better would be the ability of the store to provide additional information and services to its loyal customers. Here again



Figure 4 Two models of the grocery shopping business

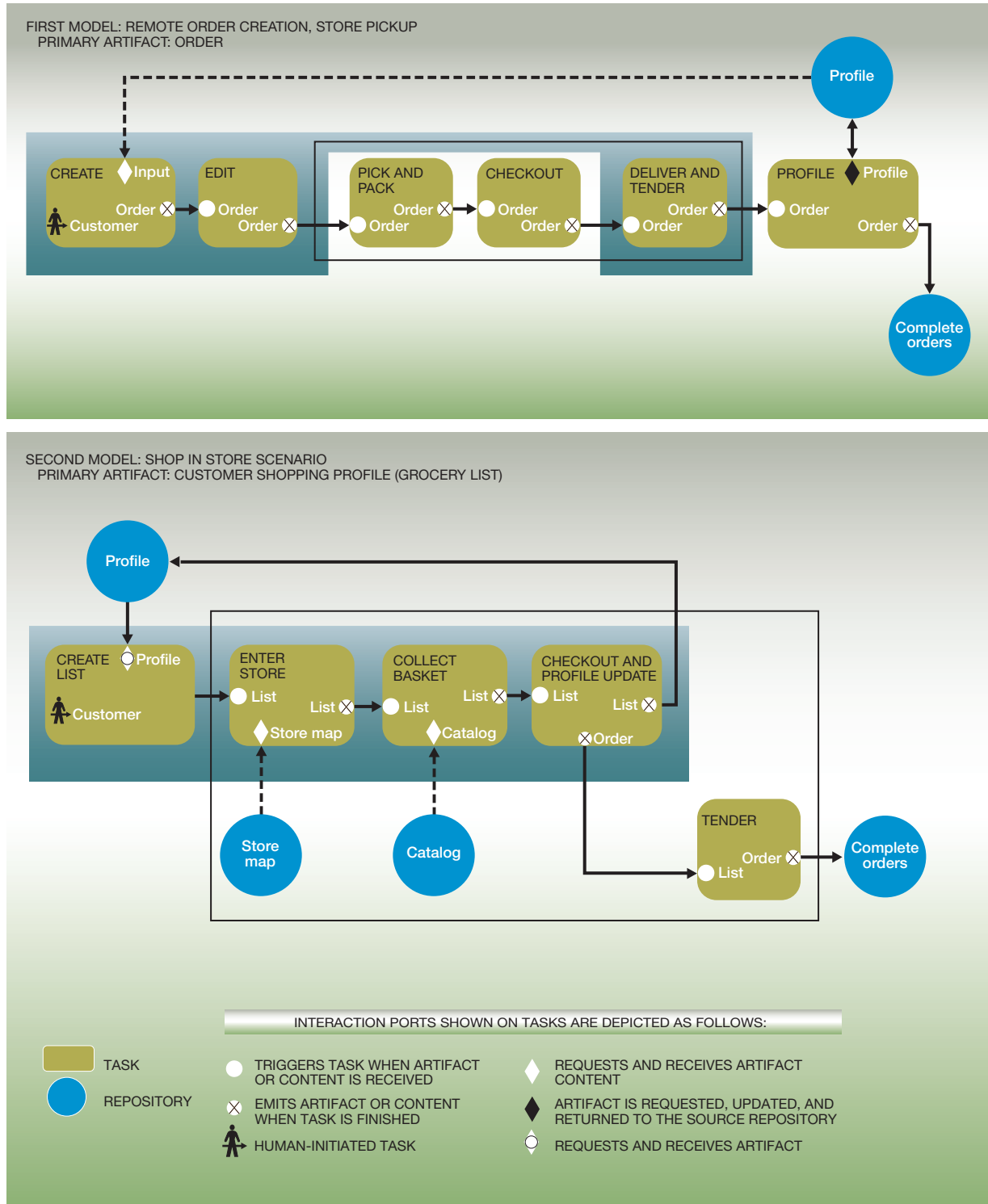
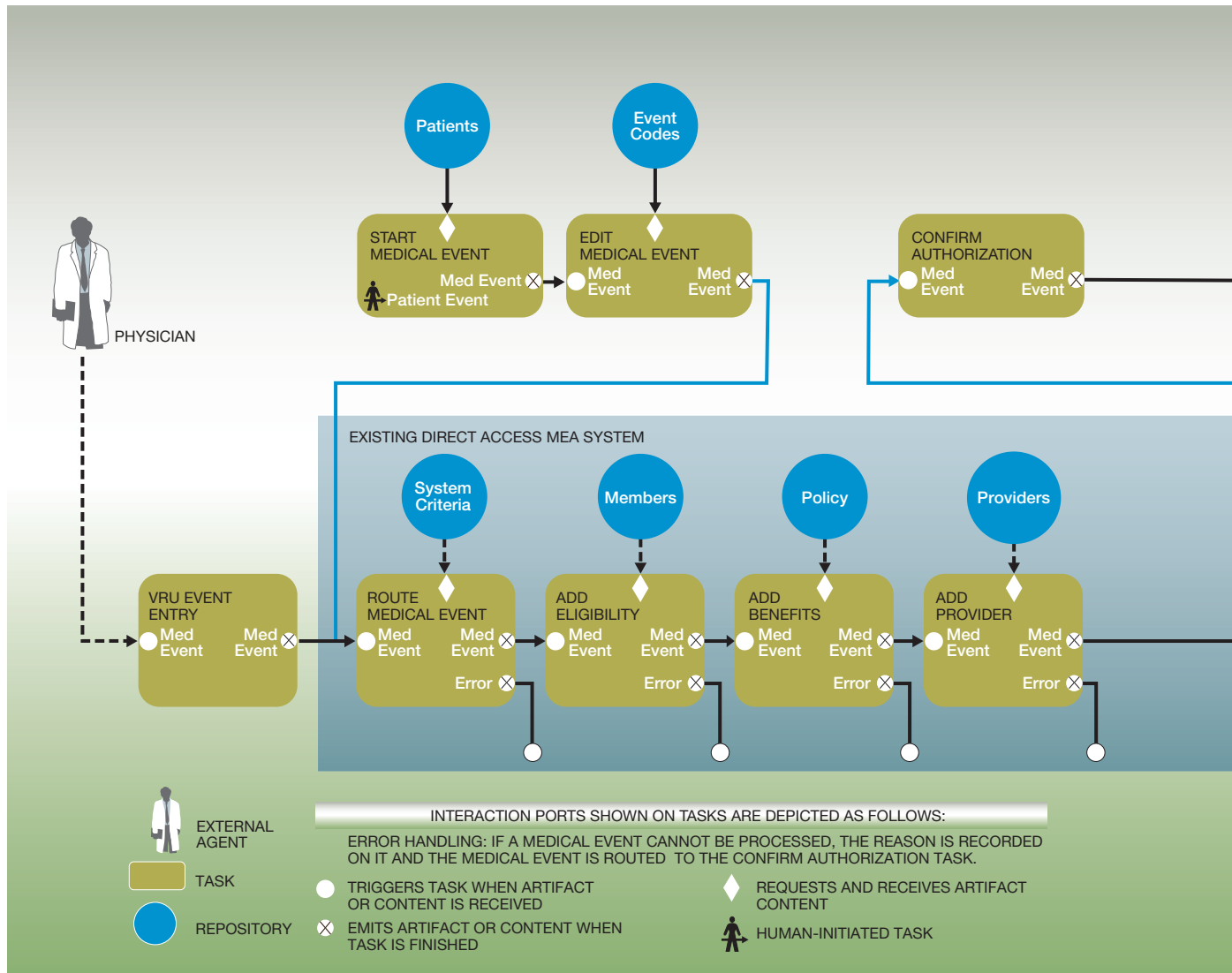


Figure 5 Modifying the medical event authorization processing to support broader access

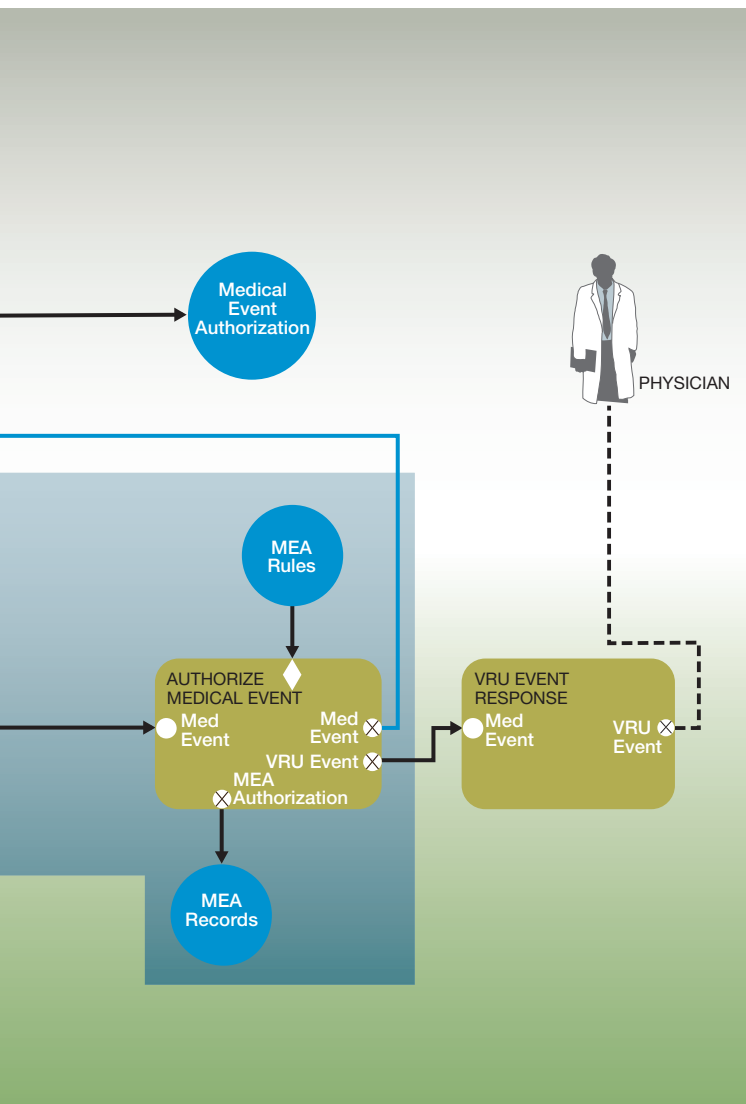


the experience showed that focus on the key artifact allowed for analysis and clarification of the business goals.

We have included two OpS diagrams in Figure 4 that capture two different programs. The first allows the customers to keep building their orders on their PDAs at their convenience, to transmit their orders to the store electronically, and to pick up the goods at the store. The second deals with a more conventional scenario of showing up at the store with a shopping list.

Important distinctions in the business analysis of the projects represented by these diagrams were: (1) whether tasks were performed by the customer or store personnel, and (2) whether tasks were performed in the store or not.

These two different dimensions are shown in Figure 4 by a shaded area to aggregate tasks that are performed or participated in by the customer role and an area outlined by a box to aggregate tasks performed at the store location.



**Managing applications.** An operational model of an entire business provides a business user with a powerful tool to manage the business as well as the underlying applications that automate the business. Typically, a business employs a variety of applications that automate different functions for parts of the business. As the business changes, the applications need to be modified to cope with the changes. Often the applications take on a primary role, and

the business has to operate within the constraints laid down by the IT people.

In the OpS approach, business people own and maintain the operational model. We have already seen how this operational model is built from interacting artifact life cycles. The operational model is also linked at the task level to the applications that support and automate the task. Hence, the operational model also becomes the mechanism for communicating changes, and the business rationale, to the application programmers. We illustrate this point with some observations from the system that we built for health-care management.

We built a system to automate the authorization of medical events—the key artifact as discussed earlier was an MEA. The system was a direct implementation of the operational model for MEA processing, shown in Figure 3. Distinct software modules, some of which interfaced with nonintegrable legacy applications, supported each task. This system allowed medical providers to create an MEA using the software that they were given and to submit it to the insurance company for authorization. The decision logic was contained in a set of rules that were maintained by health-care management personnel, usually nurses. The authorizations were delivered to the provider’s workstation quickly. Doctors and hospitals loved this system, because they could now complete authorizations instantaneously as compared to the time spent on extended phone calls (including the wait times).

However, this capability was only available to doctors and hospitals that had a relationship with the insurance company and had been given proprietary software. Business needs demanded that the MEA processing capability be made available to all the contracted providers in the state through a VRU (voice response unit). This wider availability would allow any doctor’s office to call the system and obtain an authorization for a referral or an admission to a hospital. The business people examined the operational model and modified it so that an MEA artifact could also be created through a phone interaction. After it was created, the MEA could be fed into the same network of tasks that had been implemented for processing MEAs that originated from the provider workstations. At the end, processed MEAs that had originated from a phone had their results read out over the phone, for example, “Your request has been approved; the authorization number is A123456.” Empowered by this analysis, and the OpS diagram shown

in Figure 5, the providers approached a developer of VRU applications and asked to have the following two basic functions implemented:

- Interact by phone to gather the information needed for an MEA and create the information package
- Take the processed MEA, in a specific format, and read the results to the user on the phone

Using this approach, the insurance company was able to deploy the functionality in six weeks. The alternate path, which was suggested by the VRU vendor, was to build or customize a VRU application. As a specification, the vendor needed the database definition and the entire authorization logic; and the estimate for completion was one year. The lesson here is that an accurate operational model allows business users to specify application needs in a precise and efficient manner.

We refer to a specific benefit that results from a disciplined use of this approach as “linear scaling” of effort. It means that small changes in business require a small IT effort, and large changes in business require a large IT effort. How significant and extensive a change is depends on the amount and nature of change needed in the operational model. In the VRU example above, the change was quite simple because it only required the addition of two tasks to the operational model. Hence the speed with which it was implemented and deployed should not be surprising at all.

## Conclusions

Business artifacts lie at the heart of the OpS (Operational Specification) approach to operational modeling of businesses. In this paper we introduced the concept of artifacts and demonstrated that artifacts are a familiar concept for business people. For each artifact in the business, its life cycle can be constructed using the OpS constructs called business tasks and repositories. The collection of the life cycles of all the artifacts of the business, and the interactions between them, specify the operational model for the entire business. Operational models also provide a concrete underpinning to which strategy, organization, and application issues can be connected. Finally, we used our experience from engagements to crystallize some of the business value that results from using an artifact-centered operational approach. The benefits that we focused on were flexibility of the representation, ability to analyze change, and ability to manage applications. Elements of ar-

tifacts and operational thinking, as embodied in OpS, continue to influence engagement approaches as well as business-level modeling techniques at IBM.

## Acknowledgments

Many people have contributed to the OpS research effort over the years in different ways, as customers who were open to a new approach while insisting on results, and as colleagues and managers. Our memories are not perfect but, in rough chronological order, we would like to thank Kevin McAuliffe, Channing Verbeck, Kathleen Lutz, Lynn Bricking, Kathleen Jamison, Arthur Ciccolo, Dave Bennett, Karl Sandreuter, John Mackay, Grace Lin, Ying Tat Leung, Steve Buckley, Gene Brandon, Ko-Yang Wang, Laura Ketner, Michael Limanni, Nick Novis, Don Weaver, Kerry Jones, Pascal Negros, Santhosh Kumaran, Rob Guttman, and Kamal Bhattacharya for their help and encouragement. We thank the referees for their helpful suggestions.

\*\*Trademark or registered trademark of Object Management Group, Inc.

## References and note

1. T. DeMarco, *Structured Analysis and System Specification*, Yourdon Press, Upper Saddle River, NJ (1979).
2. W. Hodges, *A Shorter Model Theory*, Cambridge University Press, Cambridge, UK (1997).
3. J. P. Morrison, *Flow-Based Programming*, Van Nostrand Reinhold, New York (1994).
4. F. Leymann and D. Roller, *Production Workflow*, Prentice Hall, Inc., Upper Saddle River, NJ (2000).
5. M. Holcombe and F. Ipate, *Correct Systems: Building a Business Process Solution*, Springer-Verlag, London (1998).
6. R. Anupindi, S. Chopra, S. D. Deshmukh, J. A. Van Mieghem, and E. Zemel, *Managing Business Process Flows*, Prentice Hall, Inc., Upper Saddle River, NJ (1998).
7. M. Jackson and G. Twaddle, *Business Process Implementation Building Workflow Systems*, Addison-Wesley, ACM Press Books, Boston (1997).
8. IDEF Family of Methods, Knowledge Based Systems, Inc., College Station, TX. The most current information on the IDEF family is available from <http://www.idef.com/>.
9. Unified Modeling Language, Object Management Group, Inc. Detailed and most current information is available at <http://www.uml.org/>.
10. J. A. Zachman, “A Framework for Information Systems Architecture,” *IBM Systems Journal* **26**, No. 3, 276–292 (1987).
11. R. Brachman, “On the Epistemological Status of Semantic Networks,” *Associative Networks—Representation and Use of Knowledge in Computers*, N. V. Findler, Editor, Academic Press, New York (1979).
12. F. Bancilhon and S. Khosafian, “A Calculus for Complex Objects,” *Proceedings of the 5th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (1986), pp. 53–60.
13. S. Koshafian and G. P. Copeland, “Object Identity,” *OOPSLA Proceedings* (1986), pp. 406–416.

14. In certain situations there can be more than one “money” artifact that is central to the running of the business.
15. R. Kimball and K. Strehlo, “Why Decision Support Fails and How to Fix It,” *ACM SIGMOD Record* **24**, No. 3, 92–97 (September 1995).

*Accepted for publication March 24, 2003.*

**Anil Nigam** *IBM Research Division, Thomas J. Watson Research Center, P. O. Box 218, Yorktown Heights, New York 10598 (anigam@us.ibm.com).* Dr. Nigam is a research staff member in the Supply Chain Analysis group in the Mathematical Sciences 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, including VLSI design systems, parallel processing architectures and database machines, logic programming and databases, knowledge representation, qualitative reasoning, and operational business modeling. He has authored many journal and conference papers, has served on program committees of several conferences, and has been on the editorial board of *The Journal of Advanced Engineering Informatics* (formerly *Artificial Intelligence In Engineering*). Dr. Nigam has received a Research Division Award, a Research Commercialization Award, and an Invention Achievement Award, and he holds several patents.

**Nathan S. Caswell** *IBM Research Division, Thomas J. Watson Research Center, P. O. Box 218, Yorktown Heights, New York 10598 (ncaswell@us.ibm.com).* Dr. Caswell is a research staff member in the Supply Chain Analysis group in the Mathematical Sciences Department. 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, Berkeley. His initial research at IBM focused on interacting 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. His recent work has involved developing formal models of business operational behavior. Dr. Caswell has provided leadership in several projects in the health care, retail, food service, manufacturing, and business transformation area with both internal and external customers to demonstrate their practical application. He has received project-related awards, holds several patents, and has authored a number of journal articles.