Chapter 9. Analyzing Business Processes

Robert J. Glushko and Tim McGrath

9.0 Introduction

In Chapter 8 we introduced the notion of context as predictable combinations or clusters of requirements, and we looked at various context dimensions sometimes used to organize them. The context dimensions included aspects such as Product Classification, Industry Classification, Geopolitical, Official Constraints, System Capabilities, and Business Process Roles.

It should be clear that business process has an important place in context of use. We emphasize processes because unlike other context dimensions, we often have the freedom to refine or reengineer them to create new value. We can't easily change the country we're in, its regulatory environment, or the industry conventions or practices that strongly shape how business gets done (the latter are sometimes called the implied terms and conditions of an industry or business relationship).[1] But we can change many of the processes we carry out. So while it makes no sense to talk about the As-Is and theTo-Be geography, in analyzing business processes we often contrast the way things are with how we would like them to be. After we describe the As-Is model, we can improve its processes by applying existing patterns or best practices (see Chapter 10), or we can invent completely new ones.

This flexibility is greater for processes that are completely internal to an enterprise than for those that involve other enterprises. But the promise of service oriented architectures implemented using loosely coupled document exchanges is that as long as the interface doesn't change, the processes that create and consume the documents can.

Internal processes can change but the external business interface should not.

This flexibility has both positive and negative aspects. It is desirable because it allows us to satisfy the specific requirements of our situation. But it is undesirable because it introduces ambiguity in our definitions and descriptions of the processes. This can make it difficult to align our processes with those of other businessess with which we want to do business, because different businesses may exploit the flexibility in incompatible ways. Without a sufficient amount of detail, it is unlikely that any two process models can be meaningfully compared. For example, a business whose process models are very abstract can't easily respond to a buyer asking, "Will you accept my UBL purchase order?"

So the lesson of this chapter is how to describe processes in unambiguous and compatible ways. We advocate the metamodel proposed in the ebXML Business Process Specification, which specifies three levels of abstraction: *processes,* which are defined in

terms of *collaborations*, which are in turn, described using *transactions.*[2] The patterns at the higher levels help us identify appropriate patterns for reuse at the lowest level, where transactions and documents are visible together and most easily implemented.

Furthermore, when we describe processes in terms of document exchanges, we can more easily align and interconnect processes from different organizations or businesses to enable patterns such as straight-through processing, supply chains, or virtual enterprises. The documents are the interfaces to these loosely coupled business processes, and they can easily be realized in highly tangible ways according to the conventional notion of a document as a container or message with information components.

Business process models will contain some information components and document models will contain some processing rules.

The complementary nature of processes and documents is another reason for emphasizing process analysis in Document Engineering. In Section 3.4.5 we called this the yin and yang of Document Engineering. That description is perhaps a bit fanciful, but it is undeniable that documents and processes have an inseparable and complementary relationship. Documents contain the information that represents requests to and responses from a business process, and business processes produce and consume documents. Business process models will contain some information components and document models will contain some processing rules. We cannot know the true meaning of the information exchanged in documents unless we understand the processes involved.

9.1 The Levels of Abstraction Challenge

Consider the question, “What are you doing now?” You can answer this question at many levels of abstraction. You might say:

- “I’m living in Berkeley and taking courses at the University.”
- “I’m studying Document Engineering.”
- “I’m reading section 9.1.”

All these answers may be true, but they may not be equally useful or informative to the questioner. How you answer the question depends on how much context you share with the person asking the question. What do they already know about you and what you are doing? Did you last talk to them 10 minutes or 10 years ago? If you have a common context, it makes sense to answer the question with a very specific answer. If you don't, a general or more abstract answer is more appropriate.

This simple example illustrates a fundamental challenge when we analyze anything. Some things have a conventional level of description, and some levels may seem more intuitive or natural than others, but there are almost always alternatives to any description.

Business processes can be described at many levels of abstraction.

Business processes are particularly subject to this description ambiguity. Often we can't directly observe the processes we want to analyze. We can see them more easily when they deal with tangible or physical objects, but many business processes involve intangible goods or only information about goods. Modeling business processes is difficult because the key involvement of people and organizations, as opposed to mechanical or physical factors, can result in models that have idiosyncratic or unexpected characteristics.

We will attack the level of abstraction problem by systematically decomposing our process descriptions into a three-level hierarchy. We will use business reference models as a guide because their hierarchical organization of processes has been designed to reinforce different levels of granularity. We will use metamodels for process descriptions at each level that provide us with standard metadata for defining what the processes mean and how they are carried out.

9.2 Analyzing Business Organization

We analyze a business to create a common understanding of how it works and the domain in which it operates. The level at which we start our analysis, and the amount of

detail in the resulting analysis, depends on where our emphasis lies on the continuum from strategic initiatives to merely tactical projects.

We'll present a modeling approach in this chapter that starts with the most abstract perspective and works its way down to progressively more granular models. Some business organizational patterns are described using the B2C, B2B, and the other acronyms we discussed in Section 4.1.2 that characterize business relationships by their *commerciography*.[3] Even these extremely coarse patterns raise predictable issues and challenges about producer-consumer relationships, legacy technology, competition, governance, and regulations.

When we look inside a business, we might be tempted to rely on its organizational model as an analogy to its process model. But from a business process perspective, the functional business areas of any organization, such as manufacturing, engineering, marketing, sales, finance, and human resources, are purely logical entities that exist to carry out a company's business model. There is no necessary relationship between business process patterns, an enterprise's management structure, and the support for carrying out the processes in facilities, technology, and systems.

UC Berkeley's organizational model is appropriately complex for an enterprise with thousands of employees. Its organization charts depict an enterprise headed by a CEO called the Chancellor, with dozens of staff and academic units arranged in a multi-level hierarchy of departments and schools, each with an executive manager called the Department Head or Dean. But these organization charts don't capture the unique character of a university, where the principle of academic freedom is fundamental, with each professor and researcher free to pursue the work that most interests him or her. This autonomy in academic affairs has a parallel manifestation in the operational side of the university, and there is substantially less top-down management than in a commercial corporation of similar size. What this means for the Event Calendar Network Project is that there are no enterprise standards or procedures for event calendars and that any organizational unit is free to create its own calendar.

There are no necessary relationships between business processes, management structure and facilities, technology, and systems.

This is a subtle but important point. The fact that an enterprise performs a purchasing process does not imply that it has a purchasing organization, or that it uses a purchasing application. And even a phrase like Enterprise Resource Planning (ERP)—which usually

suggests an application from SAP, Oracle, or PeopleSoft—can be used in a purely functional or conceptual way to describe a business that has standardized on data models to create a synchronized and consistent view of the business's processes. Most ERP systems use a shared information store to ensure that purchasing, inventory, and accounting functions are tightly coupled so they can yield an accurate and consistent view of an enterprise's processes, orders, and accounts. But a business might achieve the same view by exchanging information between separate purchasing, inventory, and accounting applications. In this latter sense, we can describe the business as "doing ERP processes" even though it doesn't have a conventional ERP system.

Of course, an enterprise's business processes, its organization, and the information technology it uses can reinforce or constrain each other. For example, a functionally organized business is very hierarchical and usually reflects a bureaucratic management philosophy that believes in centralized authority and direction. Strategy and plans are developed, goals and directives are issued, and then each part of the company follows the plans to achieve the strategy.

The model of business organization shapes the need to exchange information or coordinate across organizational boundaries. For example, functional organization enables an enterprise to focus on efficiency within each business unit and can minimize exchanges and interactions with other organizations to carry out its core business processes; a purchasing department can focus on purchasing and a finance department can focus on invoicing and accounting. But these functional units would still need to share information to reconcile orders and invoices.

The model of business organization shapes the need to exchange information across organizational boundaries.

The nature of interorganizational information exchange (or the lack of it) reflects an assumption behind functional organization that the business environment in which the business operates is relatively stable and that operational efficiency is the key to its

success. Such a business might have carefully documented processes and be relentlessly focused on both following them and improving them.

But a business can't be good at everything; one business may view operational efficiency as its key to success, while another may strive for product innovation, and another may aim for unsurpassed customer satisfaction.[4]

A focus on satisfying customers is often the motivation for a cross-functional organization in which some of the core business activities are duplicated across product lines, customer segments, or geographies. A cross-functional organization requires more coordination and information exchange between business units, but this overhead can yield substantial benefits if it is used to create a more responsive and value-focused business. Such businesses are likely to tolerate less rigorously specified processes, and some might even encourage employees to ignore them if they get in the way of satisfying customers.

Few companies need to develop all functional business areas to the same extent, because the relative emphasis and resources they require depends on their role in the enterprise value chain. A successful business focuses on the activities that are essential to its definition of success and doesn't squander attention and resources on those that are not.[5]

This idea of *core competency* is the essence of a high-level description of a business. A model of a business at this very high level helps us understand it independently of its current or future technology. It is a strategic view that can identify some of the gaps, inefficiencies, overlaps, and opportunities in what the business currently does or does not do. At this level of modeling, the view of a business is highly qualitative and usually recorded in narrative form, perhaps with some accompanying diagrams like organization charts.

9.3 Analyzing Business Processes

Our ultimate goal when we model business processes is to describe what the business does in a hierarchy of detail from an abstract level down to the level where documents

and specific information components in document exchanges are visible. But when we analyze processes, the information we discover will come from many sources and at many levels of abstraction.

It helps ensure consistency and completeness if we try to answer the same questions for each process we encounter. If our goals are strategic, we will be taking a top-down approach and interviewing senior executives or managers with a big picture view of an enterprise. This method tends to yield processes that are very abstract or very generic, partitioning activity into large, goal-oriented chunks. Questions whose answers describe processes at this level are

- What is the name of the process?
- What are the goals or purposes of the process?
- What industries, functional areas, or organizations are involved in the process?
- Who are the stakeholders or participants in the process?
- Are there any problems with the current process?
- How could the process be improved?

Asking questions and recording their answers in a disciplined way rapidly creates a web of related information about interconnected processes from which we can develop models. We will get more useful information if we ask our questions and record the answers using the standard vocabulary and definitions for the concepts and processes within the domain we're working in, if such a business reference model exists (see Section 9.3.2).

There is no single correct way to model business processes.

But the simple truth is that there is no single correct way to model business processes and no set of questions that will magically lead to the models. For example, if we ask these same questions of less senior people in the organization, or ask people who have an operational focus or role, the analysis will take on a more bottom-up and more

technology-driven character. This will yield a greater number of transactional processes, often identified by the specific documents they produce or consume. This view is necessary for implementing and integrating the applications that will carry out the processes, but the processes will be at a vastly different level of abstraction than those identified by top-down or strategic approaches.

To truly understand a business process we need information from both the top-down and bottom-up points of view. Informants higher in the organizational hierarchy with a strategic focus are less likely to know process details or problems. But they might advocate and clearly articulate an end-to-end, customer-oriented philosophy that describes the process in an idealized form. Conversely, the salespeople, customer service representatives, order processors, shipping clerks and others who actually carry out the processes will be experts about the processes, their associated documents, and problems or exception cases they encounter but rarely recognize the conflicts in priorities between functional departments that undermine the company's overall success at satisfying customers.[6]

In any case, using only abstract organizational-level and concrete transactional-level models leaves a gap in the middle, and we can't connect business issues to technology concerns unless we can cross it.

9.3.1 Business Processes, Collaborations and Transactions

There seems to be an emerging agreement that to bridge the level of abstraction gap there needs to be a third level of abstraction in process models that fits in between the process and transactional levels. We use the three-level terminology from the ebXML business process metamodel in which a process is composed of a set of related business collaborations, which in turn describe the sequence and transitions between business transactions. Each of these levels represents a different view of the enterprise; the process view, sometimes called the business domain view (BDV) describes the processes most broadly. Models of collaborations create a perspective known as a business requirements

view (BRV). The finer granularity of the transactional perspective is sometimes known as a business transaction view (BTV)[7].

This hierarchical or compositional relationship between business processes, collaborations, and transactions is shown in Figure 9-1.

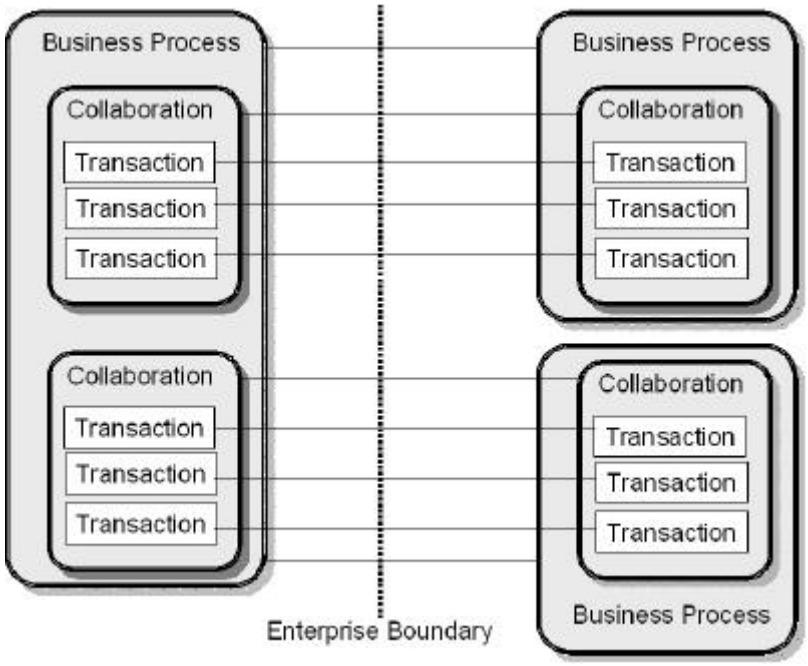

Figure 9-1. Business Process, Collaborations, and Transactions Conceptual View

Within each level of abstraction we need to synchronize the various processes and collaborations as well as the transactions that implement them. Another significant requirement implied in Figure 9-1 is that one organization's business processes may need to synchronize with more than one external process, some of which may be undertaken by different organizations. For example, what a buyer sees as a single process for procurement may include one set of collaborations involving the seller supplying the products and a separate collaboration involving the carrier who delivers them.

Business processes are synchronized by loosely coupled information exchanges using documents.

This synchronization of processes within and between enterprises requires information exchanges of some kind. As businesses adopt web services or service oriented architectures, interenterprise exchanges have increasingly become loosely coupled document exchanges. Many of the intraenterprise exchanges have also become loosely coupled, but a wider range of integration architectures and patterns are required and feasible when the information doesn't cross an enterprise boundary.

And of course, business processes do not operate in isolation. They form part of the overall business activity that defines the existence of the organization. So if we redraw the Figure 9-1 depiction to include the entire business organization, we see that there are both private (within the organization) and public (extending outside the organization) processes to synchronize. Figure 9-2 illustrates this conceptual view of an enterprise with connections between each level of process abstraction.

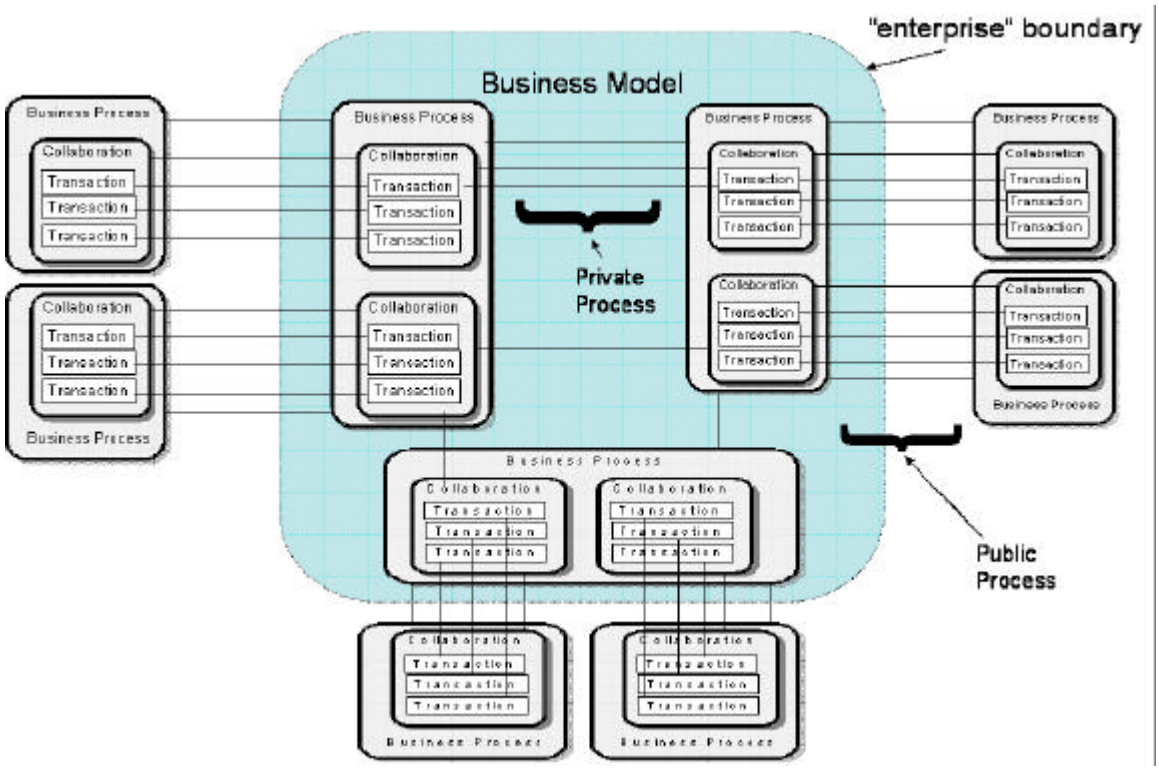

Figure 9-2. A Business Model Conceptual View

We've used the phrase *enterprise boundary* because it is often used to distinguish between processes that can be controlled and those that can't. But the same distinction can apply to the interaction between the head office of a single large business and other divisions or subsidiaries that have the autonomy to operate in ways that best fit their environments. Domain of control or service domain, are more general phrases that fit both the within-enterprise and between-enterprise situations.

9.3.2 Business Reference Models

A business reference model captures the consolidated wisdom about how to think about and carry out the most important or frequent business processes. It standardizes the vocabulary and definitions for processes within a particular industry or domain. These standards enable unambiguous communication between participants and facilitate the measurement, management, and improvement of their processes. For example, SCOR is an influential reference model for describing supply chains.

A reference model can be the default To-Be model for a business.

A reference model focuses a business modeling effort on determining whether it is possible to close the gap between what the company is and what it would like to be. Because it embodies the best practices in an industry, a reference model is the default To-Be model for a business. A reference model focuses a business modeling effort on identifying the gap between the As-Is and the reference model and determining whether it is possible to close it.

Many reference models organize processes using a three-level hierarchy, which supports our argument that a third level is needed to bridge the abstraction gap between processes and transactions. Reference models are highly reusable precisely because of the significant care taken in their development to create a hierarchical framework in which the process descriptions at each level are consistent in abstraction and detail. If a reference model exists in an industry, it would be foolish not to use it because such models consolidate a great deal of domain knowledge. Nevertheless, many businesses fail to take advantage of them.[8]

We expect the Federal Enterprise Architecture of the U.S. government to become an extremely influential reference model for the many e-government initiatives now underway throughout the world.

SIDEBAR: Federal Enterprise Architecture

The U.S. government consists of a bewildering number of departments, agencies, programs, and other organizational entities that do not interoperate well because of legacy technology, processes, policies, and politics. Consider the challenge of creating the Department of Homeland Security from 22 different agencies, with 22 different personnel systems, 7 payroll systems, and more than 170,000 employees.[9] At least 11 of these agencies have some responsibility for border security.

The Federal Enterprise Architecture is an extremely ambitious and important effort to improve how the U.S. government does business by taking a cross-agency perspective on products, services, and processes and recommending XML and web services throughout. The FEA Business Reference Model (BRM) is one of several interrelated reference models.

The BRM organizes what the government does in four business areas: Services for Citizens, Mode of Delivery, Support Delivery of Services, and Management of Government Resources. In turn, these four areas contain 39 lines of business, 19 of which are in Services for Citizens and are called external. The rest are the internal ones that support the external ones. The lowest level in the BRM hierarchy is that of subfunctions, of which there are 153. For example, the Community and Social Services line of business contains subfunctions for Homeownership Promotion, Community and Regional Development, Social Services, and Postal Services.

By describing the U.S. government in terms of business areas and activities instead of according to the agencies, bureaus, and offices that provide them, the FEA BRM will identify and reduce redundant capabilities, activities, and infrastructure. It is hoped that this will facilitate standardization of data models and business processes and encourage shared technology investments. But because it will improve the delivery of products and

services to the government's customers, the ultimate beneficiaries of the FEA BRM will be any business or person who interacts with the U.S. government.

9.3.3 Business Process Modeling Artifacts

We've stated numerous times and in numerous ways that it is the model that matters, not the notation or set of specific artifacts in which it is represented. We might draw a diagram by hand on a piece of paper or use a general-purpose graphical design application or a UML or XML-based modeling tool. But if we haven't done the hard work to develop a good model, no depiction can make it valuable.

The information needed to create a model comes from many sources and emerges over time. We have found it useful to organize what we learn in a set of worksheets whose fields provide a checklist for capturing both descriptive information and the metadata needed by more formal notations.[10] Figure 9-3 is a business domain view worksheet, the first of several business process modeling worksheets that we introduce in this chapter. This worksheet records our initial high-level observations about the Event Calendar Network project.

BUSINESS DOMAIN VIEW WORKSHEET	
Worksheet ID	UCBCalendar-BDV-1.0
Business Domain Model Name	Event Calendar Network
Industry Segment	Public University
Relevant Standards or Reference Models	SKICal[11]
Domain Scope	Describe upcoming events and publish them on one or more calendars
Business Justification	Improve efficiency in producing calendars and publicizing events Enrich the academic, cultural, and social experiences of members of the university community

Figure 9-3. Business Domain View Worksheet for the Berkeley Event Calendar Network Project

A more formal modeling artifact for process models is a UML use case diagram (Figure 9-4a). It is relatively straightforward to derive a use case diagram from information collected in a business process area worksheet (Figure 9-4b) or a business process use case worksheet (Figure 9-4c). These capture the progressively refined answers to the questions about the process that we posed at the beginning of section 9.3. Naming each process by following a verb-noun pattern ("Submit Event," "Review Event") with optional adjectives makes the analysis and its recording more consistent.

The primary goal for our Berkeley Event Calendar project was to create a service that could describe events taking place on campus.

From this we identified the two major activities as maintaining information about events and creating calendar documents that describe these events.

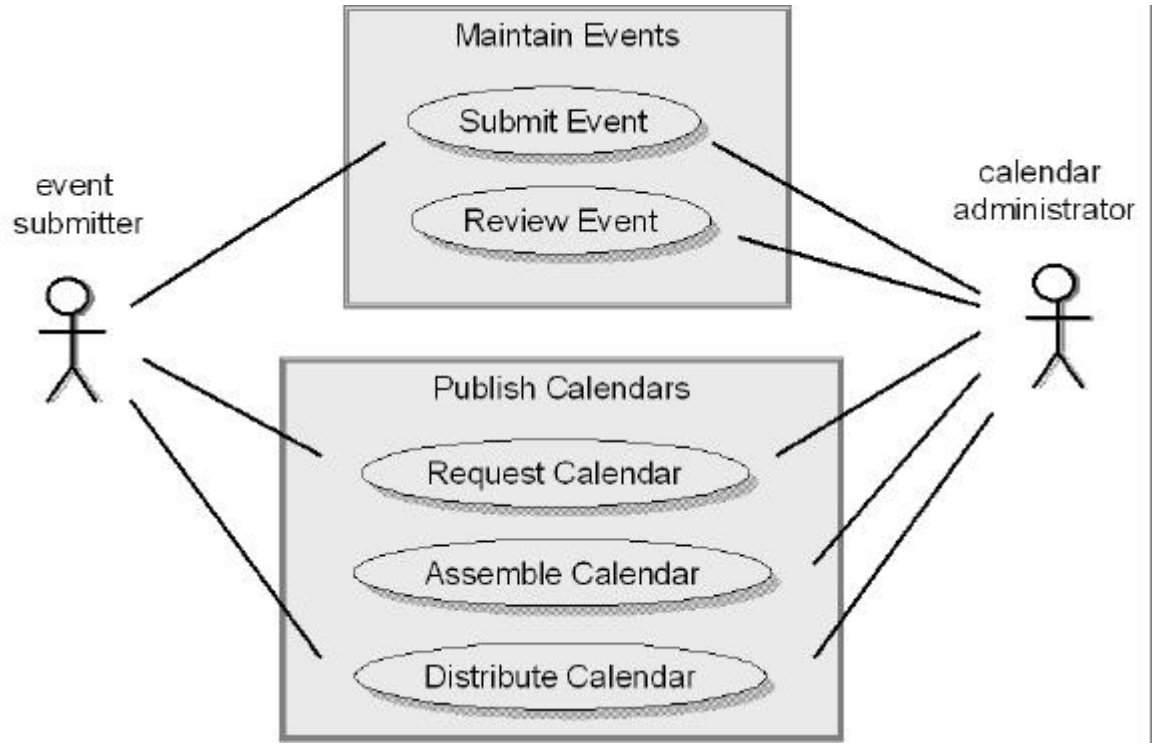

Figure 9-4a. Business Process Model of the Event Calendar project

We can represent these business processes using the use case diagram in Figure 9-4a. The diagram portrays most of the information in the business process area worksheet (Figure 9-4b) and business process use case worksheet (Figure 9-4c).

BUSINESS PROCESS AREA WORKSHEET	
Worksheet ID	UCBCalendar-BPA-1.0
Business Area Name	Central calendar
Description	Parties submit event information to Public Affairs Department for publication in university calendar
Scope	Decentralized culture of university rules out a general-purpose content distribution system; focus on semantics and processes of event calendars
Stakeholders	Primary: event submitters, calendar administrators Secondary: students, staff, faculty, public
Process Areas and Business Processes	Maintain events • Submit event • Review event Publish calendars • Request calendar • Assemble calendar • Distribute calendar
Process Goals	Efficient event submission Secure and reliable event maintenance Prompt publication to interested parties and relevant calendars
Constraints	Need a common model of "event" Calendar administrators must be able to approve events before publication

Figure 9-4b. Business Process Area Worksheet for the Event Calendar project.

BUSINESS PROCESS USE CASE WORKSHEET

Worksheet ID	UCBCalendar-BPA-MaintainEvents-1.0
Business Process Use Case Name	Maintain Events
Description	Events submitted to main calendar are first reviewed by calendar administrator Submitter is informed of approval or rejection. Approved events are entered into the central calendar Changes to approved events may be updated in the central calendar
Actors	Event submitter, main calendar administrator
Preconditions	Event submitter must be authorized individual or organization
Begins When	Event submitter fills out "submit event" form
Ends When	Expired or cancelled events are deleted from the central calendar
Constraints	Event submitter must be notified of acceptance or rejection within reasonable time (TBD)
Exceptions	Event rejected
Postconditions	Event published in main calendar

Figure 9-4c. Business Process Use Case Worksheet for the Event Calendar project.

Worksheets and UML diagrams of various types are complementary representations of models that are highly useful for people. However, neither format is directly able to drive or be interpreted by an application, so a more computer-processable format is ultimately necessary. In a web service application, for example, the model's final implementation is likely to be in XML.

Application interfaces require a computer-processable model format.

Automating the link between a model and its implementation empowers the business analyst, but it is also valuable to follow the linkages in the opposite direction so that

developers (or other applications) can understand the business processes that software is carrying out. This end-to-end *traceability* from implementations to the original business requirement and vice versa is very difficult to achieve because it requires a huge amount of discipline to ensure that every modeling artifact can be related to those that precede and follow it.

High-level process level models are unlikely to be directly executable because of the abstraction gap, between them and the specific transactions that ultimately carry them out. But they can be indirectly connected by links between process, collaboration, and transaction models. So even if we don't expect to realize complete traceability, the goal is worth keeping in mind.

Making the effort to maintain accurate modeling artifacts is essential when the work crosses enterprise or organizational boundaries. Detailed worksheets and diagrams can be critical mechanisms for communicating requirements in strategic projects of broad scope where a large team of designers and developers must work together.

Accurate modeling artifacts are essential when the work crosses enterprise or organizational boundaries.

But ultimately, it is the end result that matters, not the intermediate modeling artifacts. In tactical projects of narrow scope, a small team might prefer more *agile modeling*[12] approaches that emphasize rapid and iterative design cycles and that deemphasize efforts to create and maintain the linkages between various models. Even then, there is fine line between investing too much in modeling artifacts and not investing enough to make them useful, and each project needs to find an appropriate balance.

9.4 Analyzing Business Transactions

The business transaction level of abstraction in business process analysis is the easiest to recognize because it is where we find the documents that are exchanged. We define a business transaction as describing the exchange of documents and business signals in a trading or commercial relationship between two parties. A transaction implements a binary relationship between two parties, one playing the requesting role and the other the

responding role. There will always be a requesting document, and many transactions also involve one or more responding documents.

More questions must be answered to analyze processes at the transactional level. These include:

- When does the transaction take place?
- What transactions precede and follow the transaction?
- What information is needed to start the transaction?
- What information is produced by the transaction?
- What can go wrong?

Business transactions and database transactions both have at their core the notion of an indivisible unit of work, but they are distinct concepts. (See the Sidebar)

SIDEBAR: Business Transactions and Database Transactions

The classical definition of a database transaction is a group of statements or instructions to a database whose changes can be made permanent or undone only as a unit. A reliable database guarantees the four so-called ACID properties about transactions —Atomicity, Consistency, Isolation, and Durability—and can do so without any additional human intervention.

Database transactions also provide a simple model of success or failure: a transaction either commits (all its actions happen) or it aborts (all its pending actions are undone). A database transaction can be rolled back in the same unit with which it was committed to undo all of its effects and return the database to a prior state. It does this by locking the resources used before the transaction begins.

In contrast, business transactions cannot be rolled back. However, any obligations established by a successful transaction can sometimes be undone by a compensating transaction.

This fundamental difference between the classical database transaction and business transactions is mostly a result of differences in time scales. The time scale for a database transaction is measured in fractions of a second. But many types of business applications involve transactions that take place over a longer period of time (from seconds to days, weeks, or longer) often interspersed with other transactions. Database theory and design has been evolving to deal with these *long-running transactions*[13] that cannot reliably lock the resources they need, making it impossible to roll back to a previous state.

Database applications involving long-running transactions usually involve users in creating the actions that are part of the transaction, and the actions are based on the results of earlier actions or workflows.

9.4.1 Describing Transactions

Figure 9-5 depicts a purchasing or procurement process called Buy a Book, in which the buyer or customer buys a book from a seller, in this case GMBooks.com. The process consists of several transactions whose relationships are shown using the UML sequence diagram notation. This type of diagram is a convenient artifact for describing transactions because it emphasizes the temporal ordering of the information exchanges.

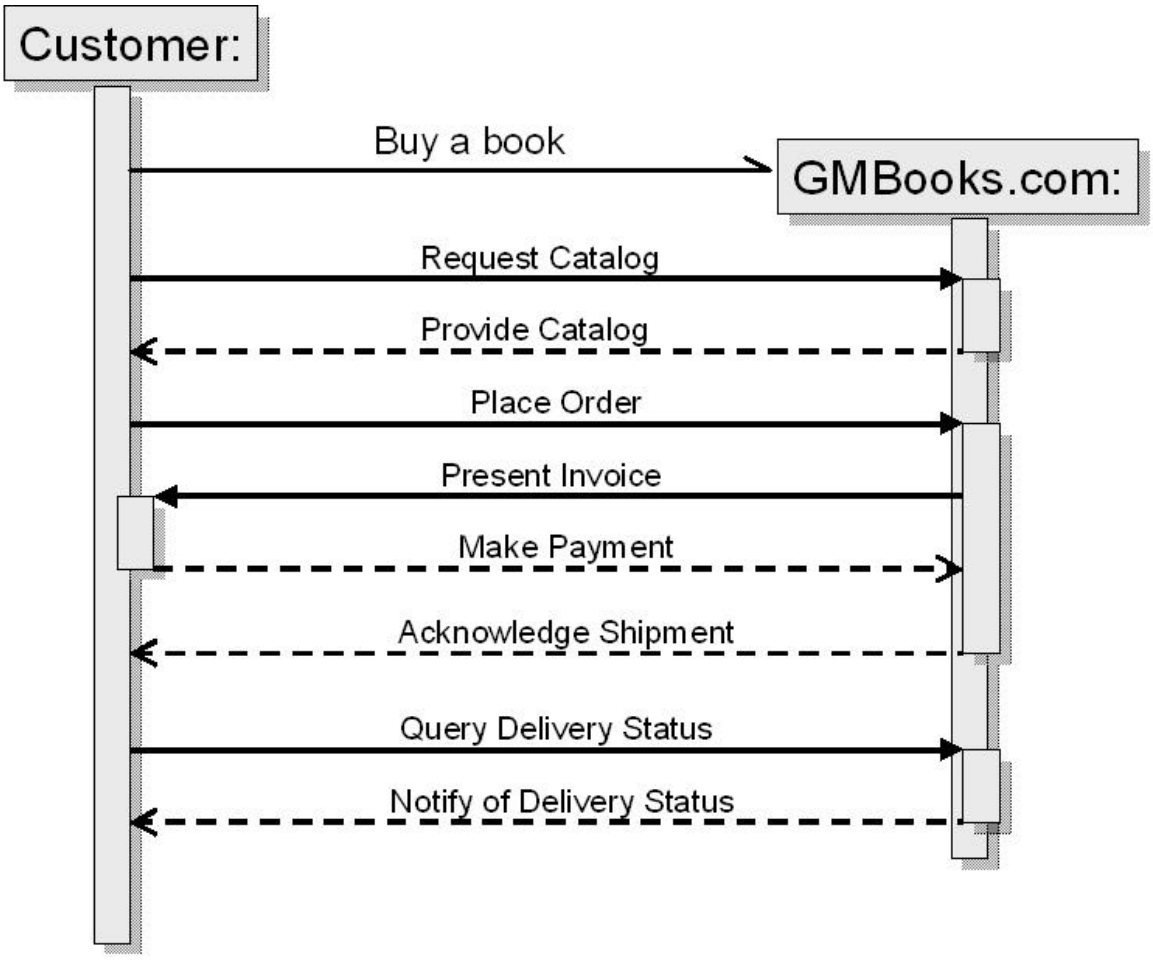

Figure 9-5. A Transactional Model of a Procurement Process with GMBooks.com.

SIDEBAR: UML Notations for Sequence Diagrams

We use the UML Sequence Diagram for representing the sequence of messages sent between participants in transactions. Descending from each participant (Customer and GMBooks.com in Figure 9-5) is a lifeline or timeline that implies the enduring existence of the participants before and after their interactions take place. The arrows between the lifelines represent the messages exchanged by the participants.

Many of the arrows terminate on rectangles superimposed on the lifeline that are called activations. These show the duration of the process that takes place in response to the message. The three types of messages are simple (represented by a simple

arrowhead), synchronous (shown by a full triangular arrowhead), and asynchronous (shown using half a simple arrowhead). An optional message exchange is shown as a broken line.

9.4.2 Documents in Transactions

Because transactions involve documents, at the transactional level of granularity the names of the processes often include or suggest the documents that are involved. From Figure 9-5 we see that the document we typically call a catalog is delivered to the customer when it is requested from GMBooks.com. The customer then sends an order to GMBooks.com to request the purchase of a book. GMBooks.com then either accepts or rejects the offer. After the customer sends a payment, GMBooks.com arranges for the book to be shipped and informs the customer. The customer might track the shipment by sending a delivery query.

We may not all be familiar with all of the names for these different types of documents. The names attached to specific types of documents are not always the best indicators of their purpose, because it is not the name of a document that defines its use. What defines a document is its role in a business transaction, because that determines the meaning of the document's content and how it should be processed.

It is not the name of a document that defines its use, but its role in a business transaction.

For example, in some procurement processes, the seller responds to a buyer's offer with an order acknowledgment document. But in the book-buying process shown in Figure 9-5, this order acceptance is implicit when the seller presents an invoice to the buyer. In other procurement processes there may not be explicit payment documents because payment is not initiated until the buyer sends a goods receipt.

This sometimes unclear relationship between conventional document names and function is evident in situations such as ordering space for shipping freight, where the document used to place the offer is known as a booking confirmation—even though it performs the same role as the document we know as an order. And of course there are

numerous examples of synonyms for most common business documents, such as invoice or statement and dispatch advice, delivery docket, or shipping note.

In the Event Calendar project, we refined our process model by recognizing that in addition to the UC Berkeley Events calendar, many academic departments maintain their own separate calendars (or lists of events) that might be relevant to students, faculty, or alumni from that department. Administrative and nonacademic areas also maintain calendars (such as the schedule of classes, calendar of key dates for admissions and registration, academic calendar, sporting events).

To get more publicity for their events, the administrators of these calendars also enter information about events they are holding into the UC Berkeley Events calendar.

The UML sequence diagram in Figure 9-6a describes the transactions required for submitting a new event. This diagram consolidates the information from three business transaction view worksheets, one for each of the three binary relationships between the event submitter, the local calendar administrator, and the central calendar administrator. One of these worksheets is shown in Figure 9-6b.

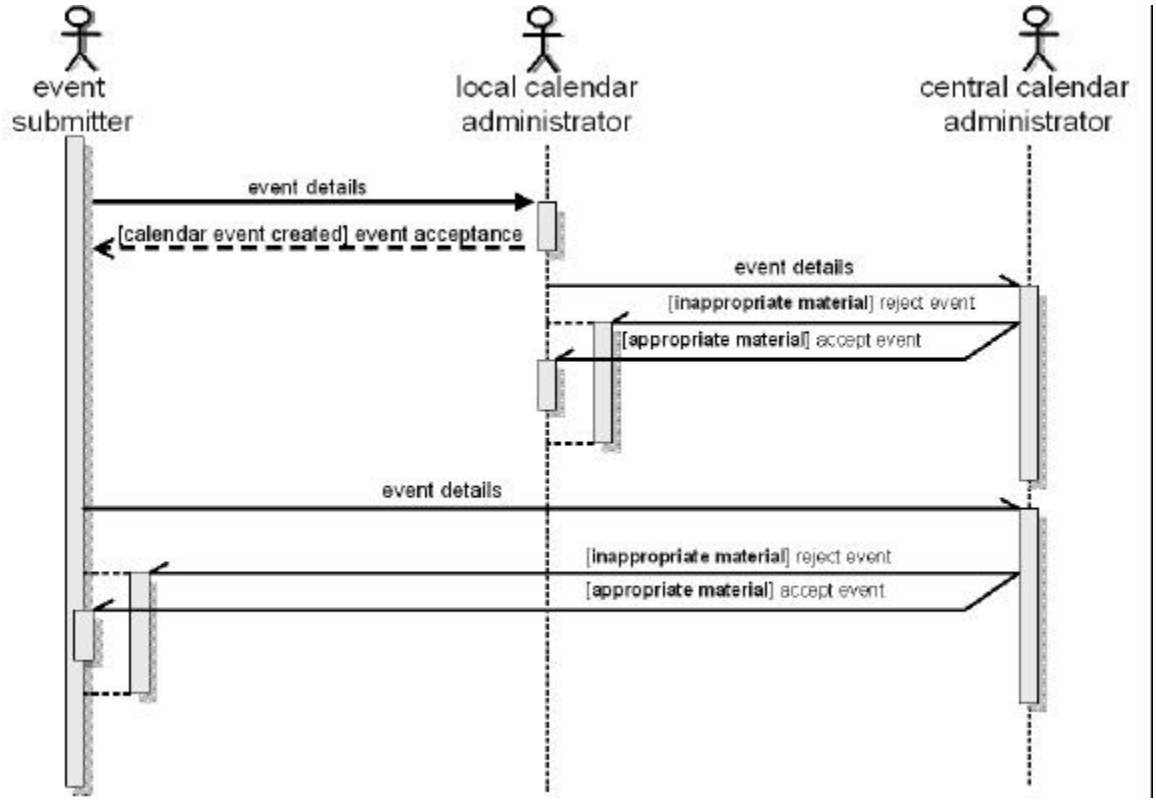

Figure 9-6a. The Submit Event Business Transactions

From this model we start to see the requirements for information components such as Event Details, Event Acceptance and Event Rejection. Further analysis exposed additional components such as Event Identification (to establish whether it was actually

a new event), Calendar Identification (to determine the correct calendar) and their related business rules.

BUSINESS TRANSACTION VIEW WORKSHEET	
Worksheet ID	UCBCalendar-BTV-SubmitLocalEventToMain-1.0
Business Transaction Name	Submit Local Event to Main Calendar
Description	Submission of event from local calendar to main calendar for publication and further distribution
Transaction Pattern	Offer-Acceptance
Initiating Partner Type	Local calendar administrator
Responding Partner Type	Central calendar administrator
Preconditions	Event accepted for local calendar
Begins When	Local calendar administrator fills out “submit event” form
Ends When	Central calendar administrator sends “accept event” or “reject event” message
Exceptions	Events can be rejected as inappropriate for central calendar
Constraints	Submitted event should be acknowledged on receipt Acceptance or rejection should be determined within 24 hours of submission
Postconditions	Local event republished on central calendar

Figure 9-6b. Business Transaction View Worksheet from Event Calendar Network Project

9.5 Business Signals: Receipts and Confirmations

Figure 9-5 is not a complete picture of the information exchanges between the buyer and seller. It shows the business documents exchanged by the parties, but omits the *business signals* used by applications to inform the other side of certain types of events. The signals are not in themselves business documents, but they provide useful feedback to the

sending side when the receiving side can't respond to a business document immediately because additional processing or decision making is necessary.

Business signals and many types of business documents function as business acknowledgments. These acknowledgments are sent in addition to any messages associated with the lower-level physical protocol layers that move information between the two parties. These lower layers are not visible and are mostly irrelevant to the perspective taken by Document Engineering in analyzing business processes.

SIDEBAR: Business Process Protocols

Protocols specify the rules that allow different parties to communicate with or transfer information to each other. Multiple protocols can be required to describe different aspects or layers of the same communication. The protocols that govern the exchange of information between businesses span the entire protocol "stack" from those involving physical devices and data connections to the behaviors and obligations required by business relationships.

The guiding principle for good communication systems is that the entity responsible for a given protocol should respond only to events or messages from its counterpart in the same layer at the other end of the communication. For example, an email server can signal receipt of a message from another email server, but it cannot respond to messages from higher-layer applications like procurement systems that might be using email to convey purchase orders. There is no way for the email server to know anything about inventory information, contractual relationships, and other factors that determine whether the order should be accepted.

Likewise, a higher-layer protocol program sometimes cannot respond to its counterpart on the other side of the business process because of communication failures at lower layers. A procurement system might not receive the seller's message that an order was accepted because it was not delivered by one of the email servers involved. It would be wrong for the procurement system to conclude that its offer had been rejected.

It needs a message from the seller's order management system, which is at the same layer in the protocol stack.

The lowest-level business signal that might be required in a business transaction model is a *receipt*.[14] This signal informs the sender that its business document has been received by the appropriate business application. It signals that the message containing the business document is (or isn't, in the case of a negative receipt) structurally and syntactically correct. This is like signing for a package from a delivery service; it communicates only that the package arrived and that it looked OK from the outside.

It may also be useful or required in a business transaction for the recipient to send a *confirmation*. This business signal informs the sender that the business document is valid (or invalid) according to the recipient's business rules. This indicates that the recipient understands the document and is willing to process it because it contains enough of the required information. It does not mean that the recipient accepts the offer conveyed by the sender's document. In the delivered package analogy, this confirmation is equivalent to opening the package and confirming that it contains all the items listed on the packing slip. Confirmation signals are often used in transactions involving legal requirements, money, uncertainty, or competing proposals.

A confirmation might contain significant business information from the document being acknowledged, making it a substantive confirmation. Confirmations of this type might include the entire contents of the received document. Alternatively, the confirmation is nonsubstantive if it contains only an identifier for the received document. A nonsubstantive negative confirmation is an error message informing the sender that the document did not have valid syntax or content, perhaps with some limited explanation for its rejection.

Finally, when the recipient decides to accept or reject the offer made by the sender, it sends a business document with the response.

These three levels of acknowledgments are superimposing, meaning that sending a response business document implies confirmation and receipt of the received document.

Likewise, sending a confirmation implies the receipt. The business document is the most important acknowledgment because it enables the business process to advance to the next step. But the lower-level signals can be important as well because they inform the participants of events that keep transactions and collaborations synchronized or on track, and it is a good practice to employ them when implementing business processes.

Signals keep transactions and collaborations synchronized.

For example, a negative confirmation signal that an order isn't valid could be sent not just to the sender but also to another process or person on the recipient's side. Using the signal in this way to reroute the order is in effect promoting the signal to a higher level in the business protocol. The relationships between the three types of acknowledgments are shown in Figure 9-7.

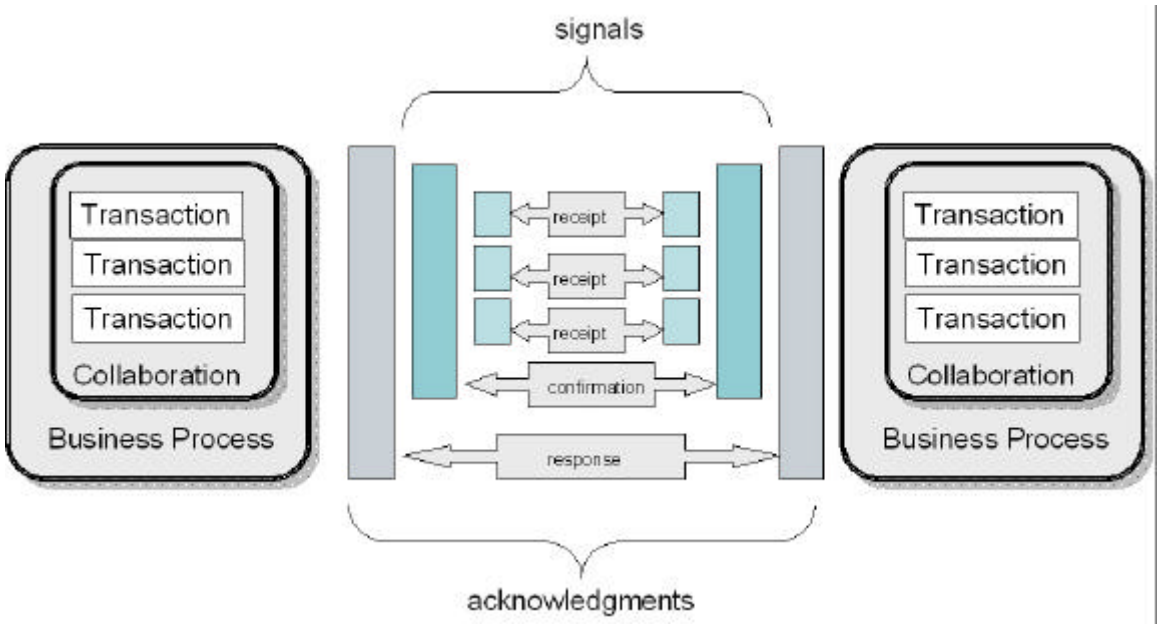

Figure 9-7. The Three Types of Business Acknowledgments

In the Event Calendar project, we identified a rule that an email confirmation is required for each *event* that has been successfully submitted. This should include an identifying reference for the submitted event. This business rule identifies a requirement not only for a receipt signal but also for a transactional component.

9.6 Transaction Patterns

Business signals help to interrelate the different parts of a business transaction, and are an essential part of what the transaction means. The presence or absence of signals also influences which of the six patterns defined in the UMM and the ebXML Business Process Specification Schema[15] is being followed by a particular transaction.

The transaction patterns differ in whether the two parties have a preexisting relationship and the extent of their business obligations or commitments to each other. These obligations can change as a business process takes place, and the change is often the intent or result of a transaction.

We will explain the transaction patterns using the Buy a book process illustrated in Figure 9-5. The patterns differ in subtle ways that are distinguished by the UML sequence diagram notation.

9.6.1 Offer and Acceptance

Many business transactions are variations of an Offer and Acceptance pattern (Figure 9-8), also called the Commercial Transaction pattern. One party sends an offer and exposes itself to the imposition of legal liability by another in doing so. Because of this legal exposure, it can be important to the offerer to know the status of the offer, so the recipient might respond with a receipt when the offer arrives and with a confirmation when it is determined to be a valid offer.

Because of its commercial obligations, the offer and the acceptance are both *nonrepudiable,* meaning that both parties must authorize and guarantee their roles in the transaction, perhaps by providing a verified or notarized signature (digital or otherwise) but most often by commercial trust.

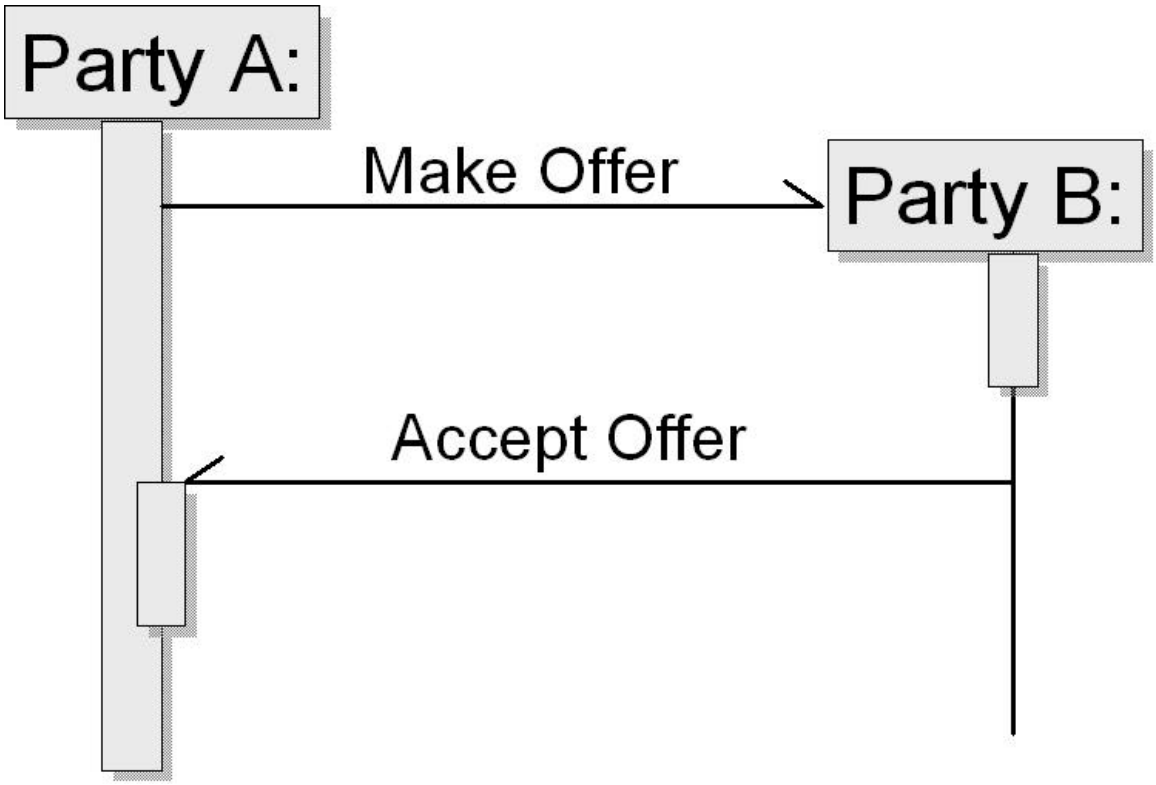

Figure 9-8. The Offer and Acceptance Transaction Pattern

A common example of this transaction pattern is that of placing an order. In fact, this offer and acceptance pattern provides the basis for the United Nations Convention on Contracts for the International Sale of Goods,[16] which is about as common as a pattern can be. Placing an order with GMBooks.com, as shown in Figure 9-5, is an instance of this pattern.

9.6.2 Request and Response

Another transaction pattern is Request and Response (Figure 9-9). This pattern is used in transaction models when one party makes a request for information and the responding party has to apply some business logic before responding. The response might depend on the identity of the party making the query—for example, when you check an account balance with a creditor. Or maybe the response needs to be dynamically generated—for example, when you enquire about stock availability of an item.

In this pattern, no binding obligations are created for the responding party. In the GMBooks.com scenario, Request Catalog would be an example of the Request and Response pattern if the catalog were tailored for each customer.

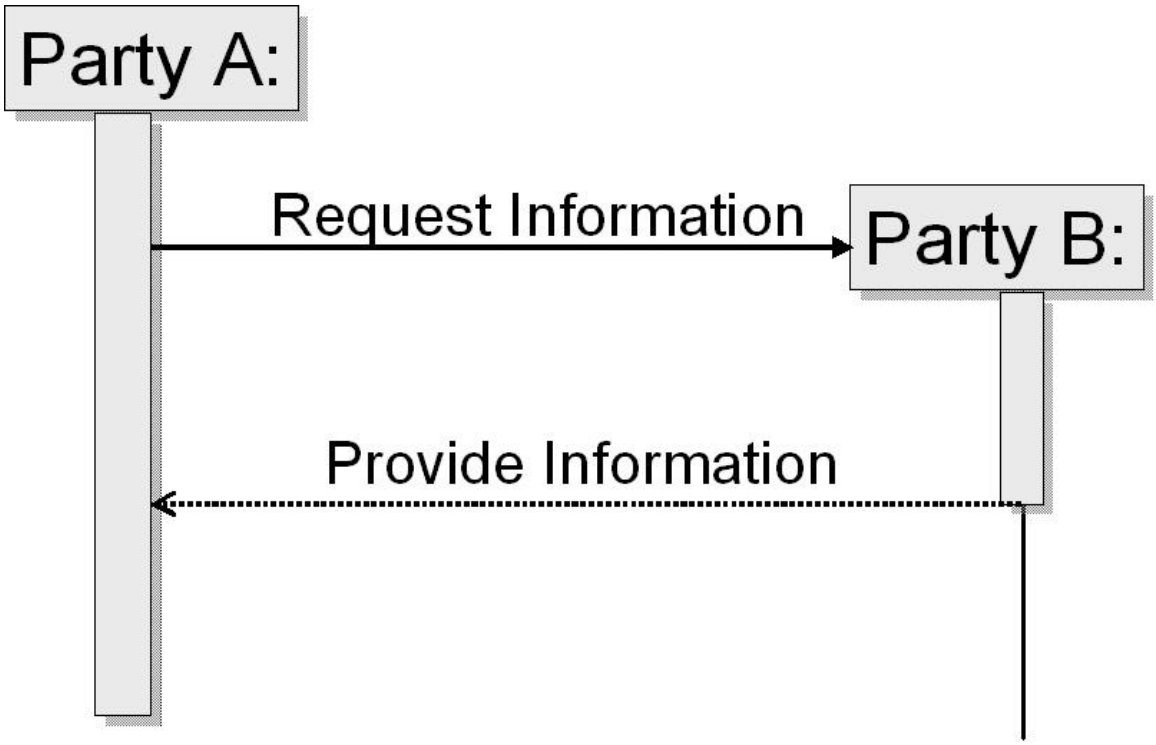

Figure 9-9. The Request and Response Transaction Pattern

9.6.3 Request and Confirm

If a request for information assumes a previously established contract or obligation, the transaction pattern is Request and Confirm (Figure 9-10). In this pattern, one party requests confirmation or status information from another, for example, as a Request Order Status transaction. In the GMBooks.com example the Query Delivery Status transaction is an instance of this pattern.

This pattern may also require some form of nonrepudiation on the responder's part.

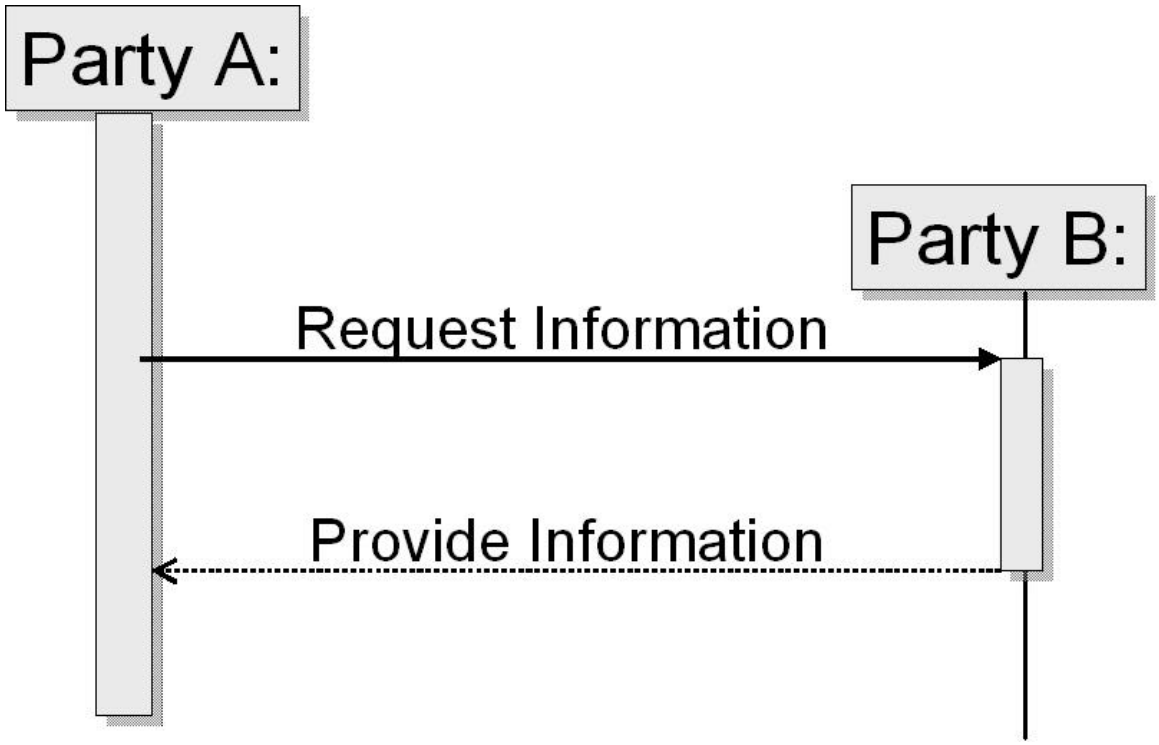

Figure 9-10. The Request and Confirm Transaction Pattern

9.6.4 Query and Response

In contrast to Request and Response and Request and Confirm, with a Query and Response (Figure 9-11) transaction pattern, the response provided doesn't depend on an established business relationship. This pattern is an appropriate model when the information being sought is static or slow changing so that it doesn't depend on the identity of the party initiating the transaction.

In the GMBooks.com scenario, Request Catalog and its response would be an example of the "query and response" pattern if the catalog were static and every customer received the same one.

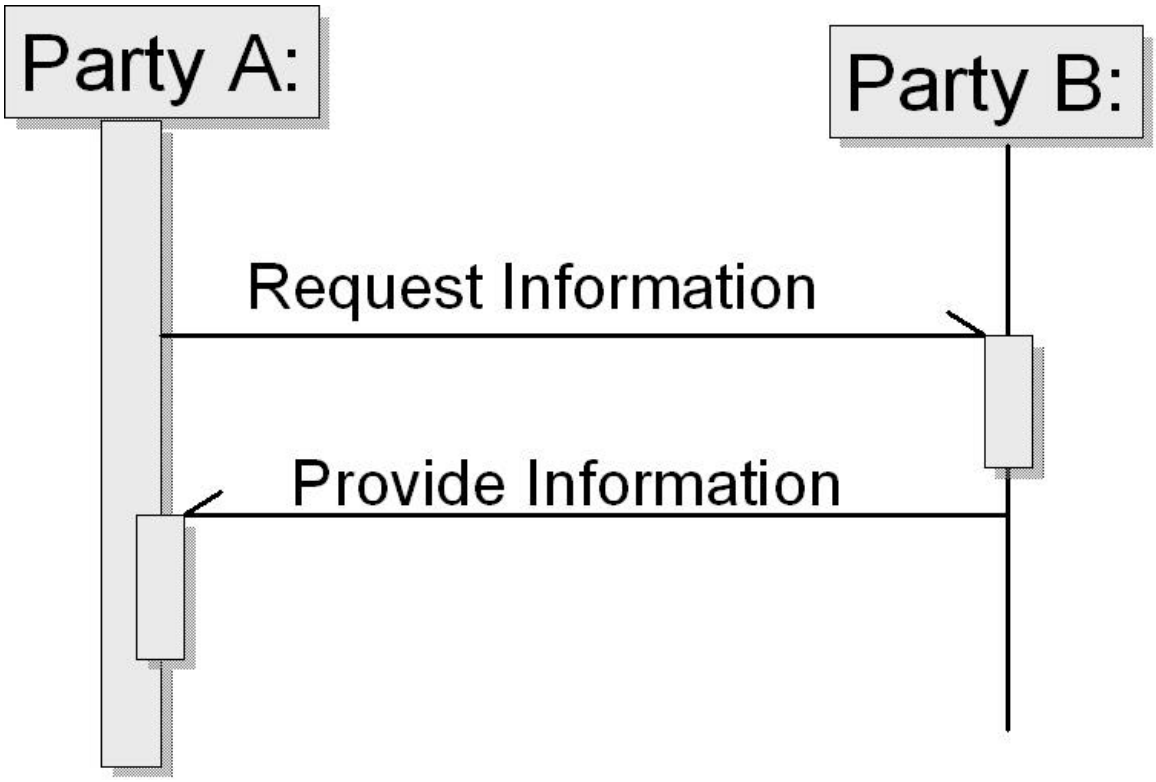

Figure 9-11. The Query and Response Transaction Pattern

9.6.5 Notification

Some transaction patterns do not require any responding document because they are inherently about unilateral distribution of information rather than bilateral exchange. The most common of these is the Notification pattern (Figure 9-12). In this pattern, one party informs the other about the status of an existing business relationship or obligation.

While there may be nonrepudiation requirements for the sender, the recipient isn't required to send a formal acceptance document. However, it is not uncommon to send an acknowledgment that the message was received.

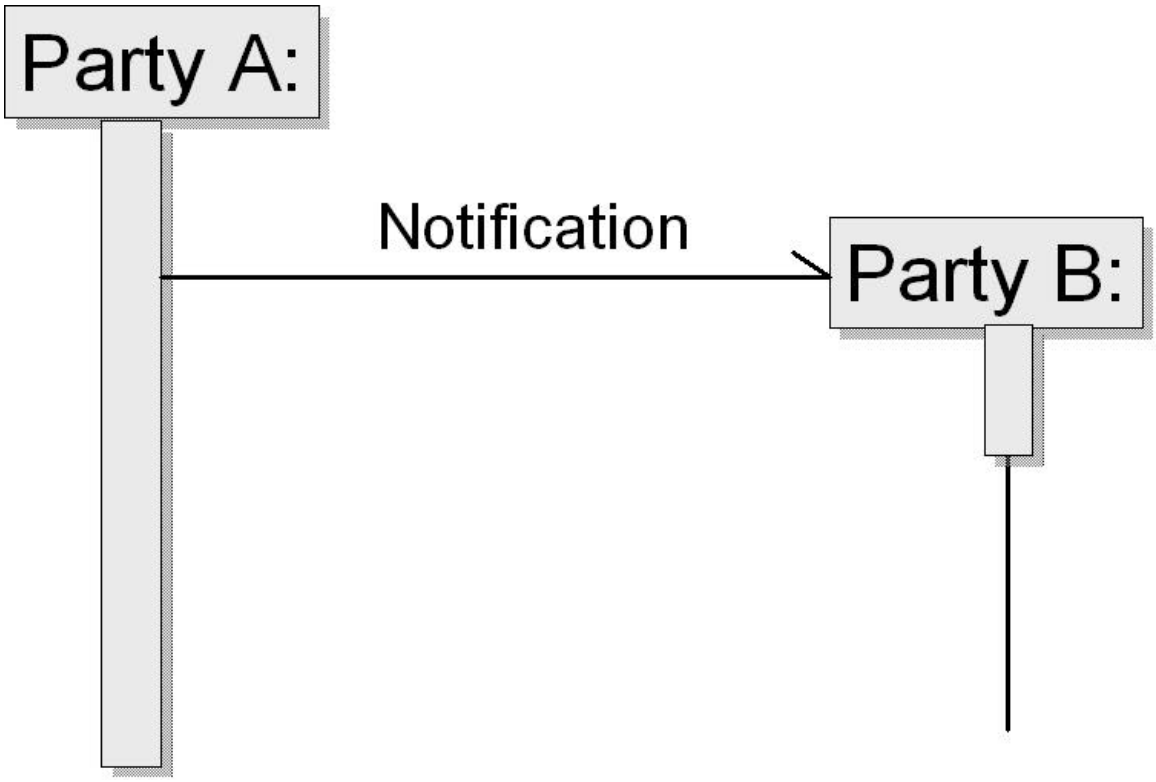

Figure 9-12. The Notification Transaction Pattern

If GMBooks.com notifies the customer when the book is shipped from the distributor, that would be an instance of the Notification pattern.

9.6.6 Information Distribution

The final transaction pattern in the UMM taxonomy is called Information Distribution (Figure 9-13). This is also a one-way transaction, often used for syndicated information exchange. It is similar to Query and Response but doesn't require a responding business document because the relationship between the sender and receiver is informal rather than contractual.

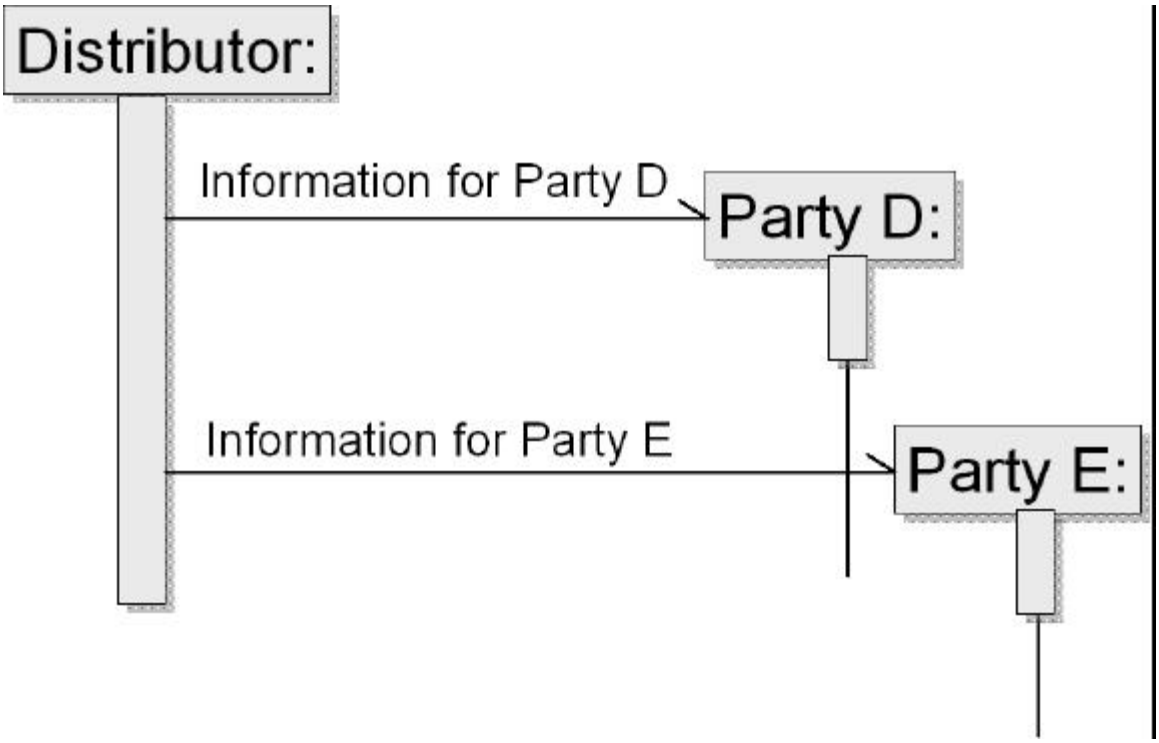

Figure 9-13. The Information Distribution Transaction Pattern

So, if GMbooks.com wanted to send out promotional material or catalogs to potential customers, they would probably adapt an Information Distribution pattern.

9.7 Analyzing Business Collaborations

In Figure 9-1 we illustrated the idea that the collaboration level in process models could group related transactions among two or more parties to provide an intermediate level of description between processes and transactions. The rationale for a collaboration level is easy to see in Figure 9-5, where the process of buying a book proceeds over an extended time period. It would be useful to organize the transactions into sets where there is a close relationship in purpose or time because then they can be reused. Ordering, tracking, and fulfillment might be thought of as reusable phases of a procurement process, each comprised of characteristics sets or sequences of transactions.

When transactions are grouped in sets where there is a close relationship in purpose or time they can be reused.

We define a business collaboration as a set of transactions with meaningful and necessary semantic or temporal overlap with each other. Put another way, a collaboration is a set of transactions that have more overlapping context with each other than with other parts of the business process that contains them all. The overlap must be have business significance. For example, they must have parties in common. Similarly, the overlap must be necessary. That is, the parties must need to know about each other's transactions with a third party for those transactions to be viewed as collaborative.[17] This "need to know" principle keeps collaboration models at a manageable size.

As an example of a business collaboration we may find that the carrier who delivers the books does not need to know about GMBooks.com's Request for Service or Contract Formation collaborations with the customer. Likewise, the customer doesn't need to know about the Book Shipment transaction between GMBooks.com and the carrier. However, all three parties need to know about the delivery of the book.

Knowing about a collaboration doesn't imply anything about which party initiates or controls it. We can differentiate a collaboration controlled or initiated by a single party (an *orchestration*) from those that are mutually controlled (a *choreography*), but this distinction is primarily important in implementation and doesn't determine which transactions it contains.

The business rules associated with transactions identify common dependencies that form collaborations.

The business rules associated with each transaction, such as the *preconditions*, *postconditions,* and *triggering events* can identify relationships and dependencies between the transactions in a collaboration. For example, the business rule that "Goods must be delivered within 48 hours of receiving the order" creates a collaboration by connecting an order transaction to those related to fulfillment.

Figure 9-14 applies these guidelines for identifying collaborations in the buying a book scenario of Figure 9-5.

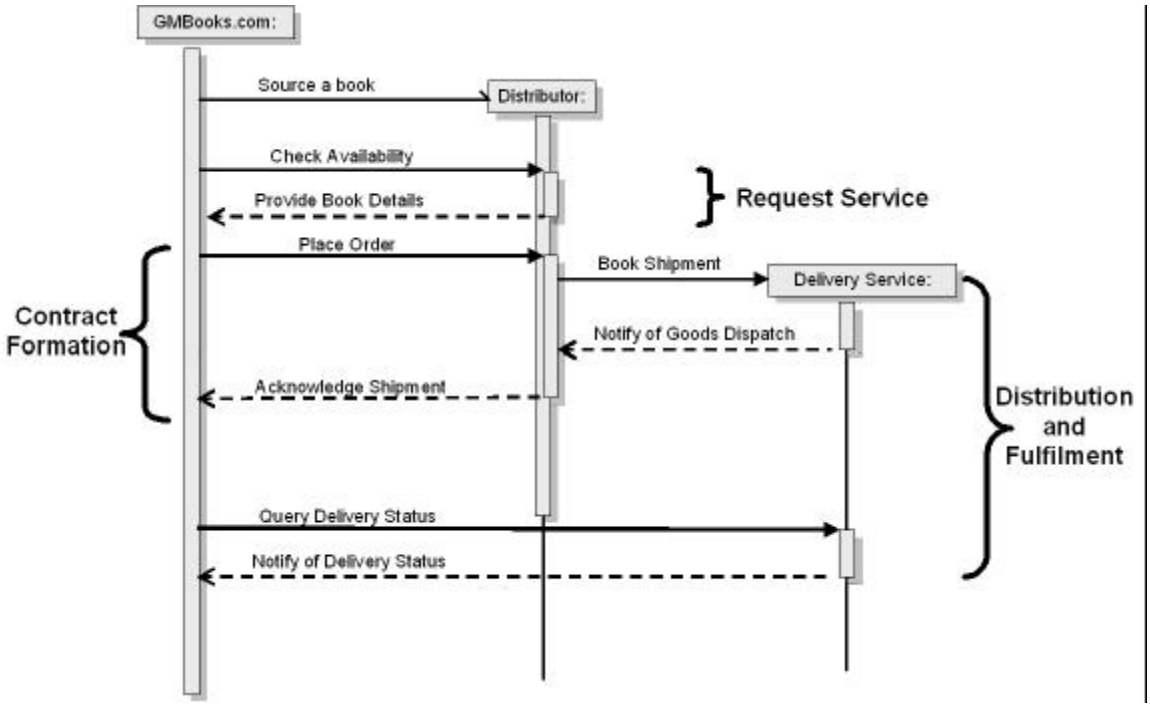

Figure 9-14. Collaborations in the GMBooks.com Scenario

At Berkeley the University Public Affairs Department has a fairly comprehensive and semiofficial calendar of events called “UC Berkeley Events” that stores event information in a database. Authorized persons or organizations can submit an event for inclusion in this system using a web-based form.

Figure 9-15 depicts the collaboration required for submitting a new event to the calendar using a UML activity diagram.

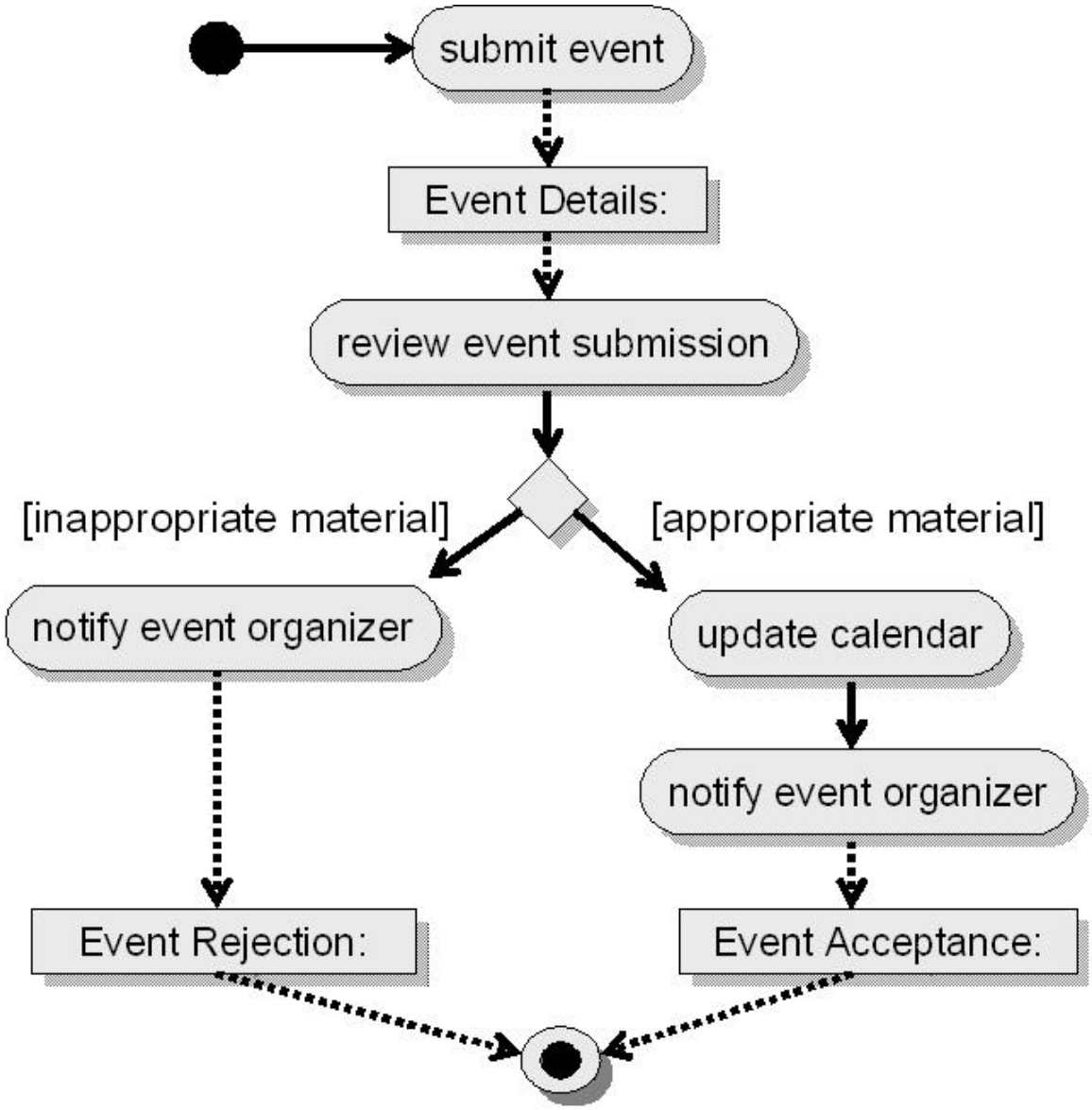

Figure 9-15. The Submit Event Business Collaboration

The collaboration begins with submission of an event and ends with either rejection or acceptance. Within this collaboration, we see transactions for exchanging event details, rejection notifications, and acceptance notifications.

9.8 Collaboration Patterns

It's not surprising that collaborations also form patterns. As we did with business processes and transaction patterns, we can list some of the more common ones as examples.

9.8.1 Contract Formation

The Offer and Acceptance transaction pattern (section 9.6.1) is simplest case of a collaboration pattern called Contract Formation. The full contract formation pattern extends back in time from the offer to include transactions that seek information needed to make one or more nonbinding proposals. It also generalizes the offer and acceptance transaction to include negotiations and counteroffers. The contract is formed when a *binding* offer is responded to by a binding acceptance.

The pattern is well documented in the ebXML e-Commerce Patterns Technical Report[18] from which Figure 9-16 is taken.

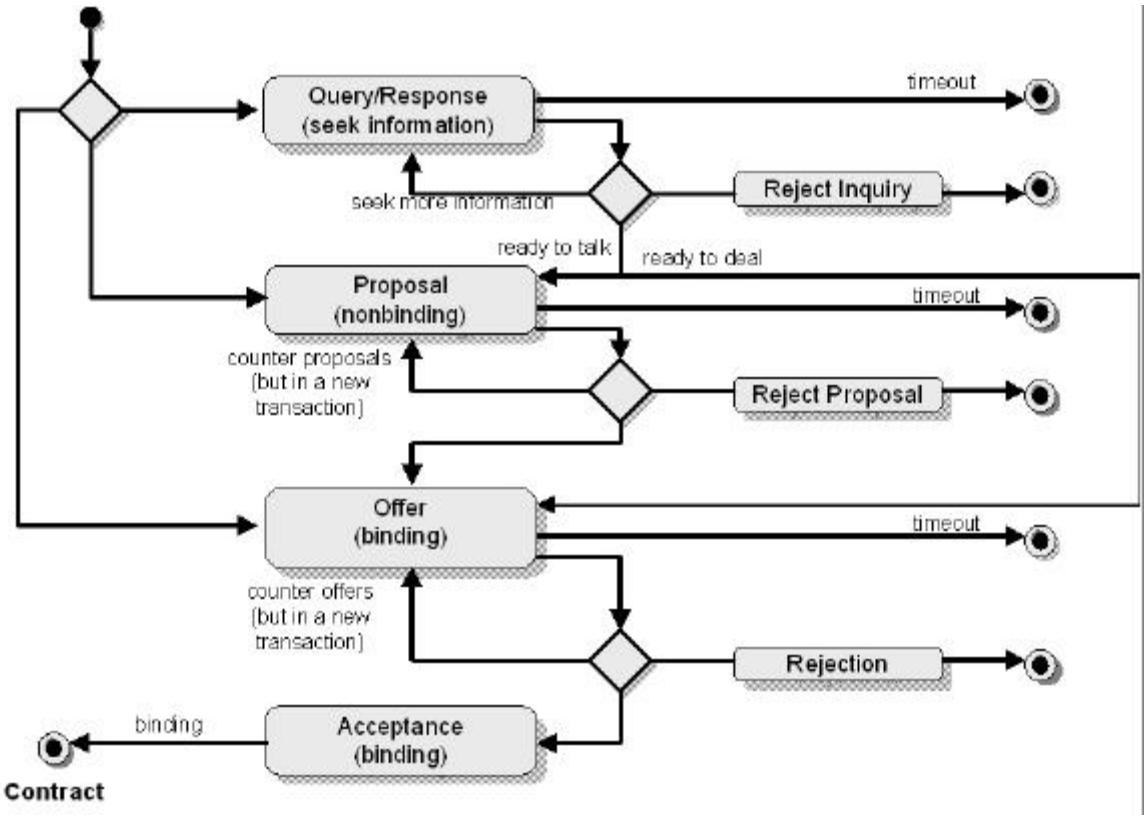

Figure 9-16. A Contract Formation Collaboration Pattern

The contract formation collaboration is often part of the procurement and auction processes. It is also a component in other collaboration patterns, such as the Sourcing and Escalating Commitment patterns we describe next.

9.8.2 Sourcing

A more complex contract formation pattern is Sourcing, the critical business process of selecting suppliers of goods or services. Sourcing can require extensive and iterative information exchange between buyers and suppliers before the buyer places an order.

The buyer might first publish a Request for Information (RFI) or a Request for Quote (RFQ) to identify qualified suppliers. The Contract Formation pattern might be invoked in negotiations to determine whether a supplier is allowed to bid. Then, before responding with a quote, a supplier might ask the buyer to explain some aspect of the requirements or might suggest why some requirement is impossible to satisfy. This might result in a revised RFQ from the buyer and might also require the contract formation collaboration to create it.

9.8.3 Escalating Commitment

If a contract negotiation ends successfully, it may trigger another contract negotiation with progressively stronger obligations to create a collaboration pattern called Escalating Commitment. We see this with business processes used in supply chains, where businesses negotiate an intention to supply goods and then increase the commitment as time progresses. This allows for scheduled manufacturing, warehousing, and subsequent shipment of goods.

9.8.4 Materials Management and Distribution and Fulfillment

The Materials Management collaboration pattern brings together all the planning, scheduling, and inventory control transactions that enable a manufacturer to ensure that the things it buys get to specified places at specified times in specified quantities. The contractual relationship between the buyer and supplier will specify the content, sequencing, and acknowledgment of Planning Schedules (or Forecasts), Shipping Schedules, Shipping Notices (or Despatch Advice) and other documents they will exchange.

The Distribution and Fulfillment collaboration pattern is the mirror image to materials management. It includes the transactions needed to get goods from a manufacturer to its

customers. Distributors, resellers, and retail outlets are usually involved as intermediaries to multiply the manufacturer's reach. Delivery service providers of various types will have separate collaborations with these entities in the delivery chain.

9.8.5 Reconciliation

The Reconciliation collaboration pattern brings together information from related transactions to ensure a single consolidated and accurate view. When we balance our checkbooks, we are reconciling our information about our transactions with the bank's information about them, being careful to consider transactions that we've initiated that do not yet appear on the bank's statement.

Many business processes involve regularly scheduled activities of aggregation, comparison, and exception handling to reconcile the work carried out by different organizations or applications. Nearly every business needs to reconcile its order with delivered goods with payments.

Reconciliation is critically important in information-intensive industries like health care, insurance, banking, real estate, financial services, and securities, where the goal of straight-through processing can't be achieved without reliably reconciled transactions and accounts. Reconciliation is also essential in synchronizing the flows of information and goods to ensure that cargo manifests accurately describe the goods being transported and that all are accounted for when they arrive.

9.8.6 Incremental Information Trail

A final example of a collaboration pattern is known as an Incremental Information Trail.[19] In this collaboration, a document in an information chain process is amended in a series of transactions involving different participants. Each may add additional information to the document at each stage in the process.

Incremental information trails are particularly relevant to the domestic and international transport community, where details of goods in transit must pass between a variety of documents such as orders, bookings, shipping advices, forwarding instructions, customs declarations, ship's manifests, delivery notes, and payments. But incremental

information trails also occur in other business processes, such the criminal justice information chain where police, prosecutors, courts, and correctional services each receive case information collected and generated in prior steps and add to the documents they create.

In fact, any document workflow process could be considered an instance of this collaboration pattern. The simplest possible variant is a Document Approval collaboration, where the information added to the original document might be nothing more than the signature (perhaps with comments) of the reviewer.

9.9 Key Points in Chapter 9

- Internal processes that create and consume documents can change but the external business interface should not.
- Business process models will contain some information components and document models will contain some processing rules.
- Business processes can be described at many levels of abstraction.
- There are no necessary relationships between business processes, management structure and facilities, technology, and systems.
- The model of business organization shapes the need to exchange information across organizational boundaries.
- A reference model can be the default To-Be model for a business.
- There is no single correct way to model business processes.
- Application interfaces require a computer-processable model format.
- Accurate modeling artifacts are essential when the work crosses enterprise or organizational boundaries.
- It is not the name of a document that defines its use, but its role in a business transaction.

- Business processes are synchronized by loosely coupled information exchanges using documents.
- Signals are used to keep transactions synchronized.
- When transactions are groups in sets where there is a close relationship in purpose or time they can be reused.
- The business rules associated with transactions identify common dependencies that form collaborations.

9.10 Notes

1. Clayton Gillette and Steven Walt, "Implied terms," *Sales Law* (Foundation Press, 1999). The customary practices or methods of dealing that are regularly observed within a trade or industry are treated as default provisions in sales contracts.
2. ebXML Business Process and Business Information Analysis Overview v1.0 (11 May 2001). http://www.ebxml.org/specs/bpOVER.pdf
3. We coined this term to convey the corporate sense of a demography.
4. Michael Treacy and Fred Wiersema, *The Discipline of Market Leaders* (Harper Collins, 1995).
5. C. K. Prahalad and Gary Hamel, "The core competence of the corporation," *Harvard Business Review* (May-June 1990).
6. Benson Shapiro, V.Kasturi Rangan, and John Sviokla. "Staple yourself to an order," *Harvard Business Review*, July/August 1992.

[7] UN Economic Commission for Europe, UN/CEFACT Modeling Methodology (UMM) User Guide (CEFACT/TMG/N093 22nd September 2003) http://www.unece.org/cefact/umm/umm_userguide.pdf

7. Howard Smith and Peter Fingar, *Business Process Management: The Third Wave* (Meghan-Kiffer Press, 2003). Efforts to reengineer and standardize business processes can be traced back to the early 1920s.
8. Todd Datz, "Integrating America," *CIO Magazine*, 1 December 2002, http://www.cio.com/archive/120102/america.html (last visited 14 February 2004).
9. Our worksheet format is inspired by ebXML Business Process Analysis Worksheets & Guidelines (http://www.ebxml.org/specs/bpWS.pdf), which contains detailed definitions of each section and guidance for filling them out.

10. G. Fitzpatrick, P. Lanner, and P. Hjelm, “SkiCal—An extension of iCalendar,” (Internet Engineering Task Force Draft, July 2001), http://skical.metamatrix.se/skical20010905.html (last visited 18 October 2004).
11. Scott Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process* (John Wiley & Sons, 2002).
12. See Helmut Wachter and Andreas Reuter: "The ConTract model," in A. Elmagarmid (ed.), *Database Transaction Models for Advanced Applications* (Morgan Kaufmann, 1992): 219-263.
13. Receipts are also sometimes called Acknowledgment Messages or ACKs, but we prefer *Receipt* because business documents can also serve as acknowledgments and might even have *Acknowledgment* in the name of the document type.
14. UMM, ibid. Chapter 9; *ebXML Business Process Specification Schema*, 11 May 2001. http://www.ebxml.org/specs/ebBPSS.pdf (last visited 15 January 2005).
15. United Nations Convention on Contracts for the International Sale of Goods (Vienna 1980).
16. We thank David Burdett for this insightful addition to our definition.
17. Jamie Clark (ed.), “ebXML e-commerce patterns v1.0” (May 2001), www.ebXML.org/specs/bpPATT.pdf.
18. Paula and Paul Swatman, “Electronic data interchange: Organisational opportunity, not technical problem,” in Bala Srinivasan and John Zeleznikow (eds.), *Databases in the 1990s*, 2 (World Scientific Press, 1991):354-374