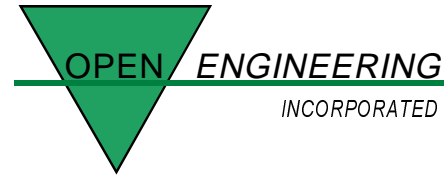


Business Objects

Modeling with Business Patterns

By Robert E. Shelton
President & CEO
Open Engineering Inc.



50 California, Suite 860
San Francisco, California 94111
(415) 989-9050 • www.openeng.com

Business modeling is the practice of abstracting and representing a business in a way that illuminates particular aspects for better understanding and communication. Models can be financial, such as business performance projections. Models can represent markets or resource supplies or demographics. Models also can represent business process or data, such as BPR process models, enterprise or business area data models. Using business objects, these later types of models can integrate our understanding of business process, structure and semanticsⁱ.

Most discussions of modeling, however, start from the premise that modeling is a “development” activity. Models, this premise argues, are developed from scratch. We research and integrate knowledge in order to create a new and more useful understanding of the business for some purpose. Surely there is a place for such efforts, as the simple (ha!) act of developing a business model can be much more valuable than the end product itself. By modeling we learn.

Yet there is an alternative approach to modeling that views the act principally in terms of adaptation (“let’s make a standard product work for us”) and maintenance (“now that I’ve got it right, let’s keep it right”). This approach to modeling starts with a base model, an abstract model that could represent many different businesses, but is specific to none. Through successive refinements, the model is transformed into an accurate representation of a specific target business. This approach to business modeling is based on parts, patterns and frameworks. This is business modeling in the not-too-distant future.

Parts

Parts is parts, but what are parts? Business objects are parts. We have previously defined three types of business objects: entity, process and event objects. **Business entity objects** are business objects that represent a concept, role, player, resource, place or thing. Examples of entity objects include *customer*, *sales order*, *product*, *equity*, *insurance policy* and *vehicle*. Business entity objects usually come to mind first when one hears the term “business object”.

Business process objects represent the activities (interactions, in object terms) which comprise a business process. One could think of business process as workflow, describing how an end-result is produced step-by-step, and how each player figures into the process. Thus entity objects are the players and resources involved in the activities described by a process object. Examples of process objects include: *order fulfillment*, *billing*, *hospital admissions*, *service activation*, and

ⁱ We have previously discussed the OOBETM approach to modeling business process, relationships, behavior, types, roles and events; so I refer the reader to previous issues of DMR for examples of these models.

product delivery. In fact, some of these processes are essentially the same -- simply the same activities that go by different names depending on the language of the industry.

Business event objects represent causes, occurrences or the passage of time. They initiate or result from activity between business entity objects. Examples of event objects include: *end of fiscal year*, *inventory bin quantity low*, *invoice due*, *shipment arrived at checkpoint*, and *call completed*. Business event objects represent happenings that are important enough to be named and tracked, pursued or avoided by a business.

The three types of business objects are different sorts of parts. Entity objects are the basic building blocks. A business needs, among other things, *customers*, *products* and *facilities*. Businesses need to do things that make money. Processes are ways to organize the basic building blocks to deliver value to others who will pay money for the results. Events fall into the category of “stuff happens” -- or “we’d really like that stuff to happen”, as the case may be. Either the building blocks and activities are resilient when “stuff happens”, or the business stops happening.

Back to business modeling. There are two ways to approach business modeling: the hard way and the easy way. The hard way involves a myth that, simply stated, reads “I don’t know what’s going on in the business. What we’re doing here is really unique. So I’ll just have to start discovering building blocks and objects from scratch.” This myth usually involves inventing new names for each part that is found, as one is clearly involved in an act of discovery never before attempted by man. Yes, I am being just a bit sarcastic, but I’ve earned those stripes... I’m guilty of the same sin! Let’s face it, we all are. That’s history. Now, how do we improve upon history?

The easy way involves realizing that there are certain parts that play in every business, regardless of industry. Sure, the names may be different but the underlying part is fundamentally the same in concept and character. Since others have gone before us, someone presumably knows what these entities, processes and events might be. For example, Figure 1 lists a starter set of business processes for BPR. True, any given list might be missing some very important parts from our viewpoint, but (just for the sake of argument) let’s suppose the list is only 50% complete. Is the glass half empty or half full?

- ***Selling*** - Order Taking
- ***Promoting*** - Marketing
- ***Procuring*** Resources
- ***Producing*** Goods or Services
- ***Delivering*** - Logistics
- ***Billing*** - Collecting Payment
- ***Developing*** New Products

Figure 1: *Although the names and shapes may differ slightly, all businesses operate certain basic processes. These can be used as a starter pack to bootstrap business re-engineering or modeling activities.*

The easy way begins by modeling from candidate lists of entity, process and event objects. Many experienced modelers take this approach, whether or not they model with business objects. Most of us adopted this technique because it saves time by taking advantage of the obvious.

Parts, it turns out, are necessary but not sufficient for next-generation business modeling. Starting with a known set of business objects does simplify the modeling process; but there is yet more leverage to be gained.

Patterns

Patterns are the next level in scaling-up abstraction, and do not necessarily conform to one or another type of part. Patterns are assemblies of parts that capture an important business concept. They are usually fragments, many of which are repeated in different contexts within any given business. More precisely, a **pattern** is a repeating occurrence of business object types and associations that can be generalized to serve more than one purpose. In this case, business object types are entity, process and event objects. Associations can be binary relationships, type structures, roles, composites or behavioral interactions.

There are three types of business patterns: behavioral, structural and semantic. A **behavioral pattern** is a repeating set of entity objects and interactions, a template for a generic version of business process such as *order fulfillment*. Behavioral patterns are interesting because a single pattern can describe many business processes.

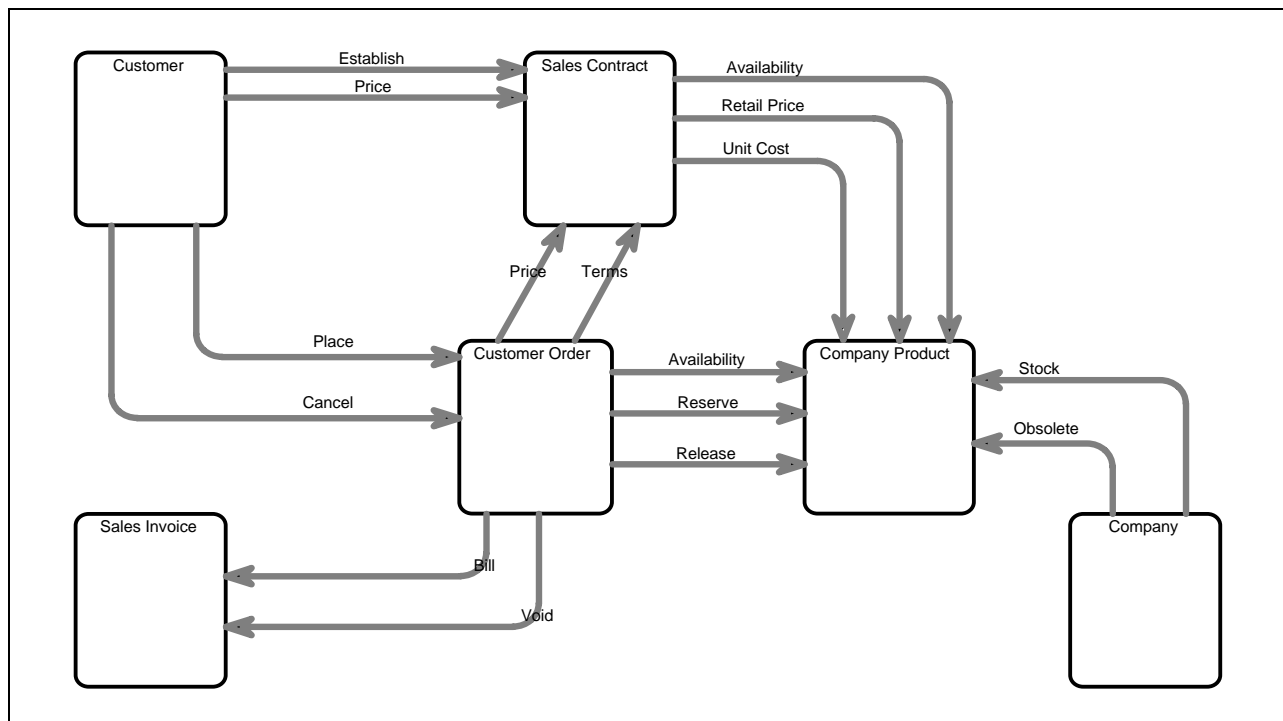


Figure 2: The order fulfillment business process is rooted in the order pattern.

Figures 2,3 are simplified examples to illuminate the similarity between *order fulfillment* and *procurement*, both of which are based on the *order pattern* (itself a specialization of the *exchange pattern*). The *order pattern* has other common and interesting variants as well: *departmental charge-back* and *return product*. In each case, the same pattern applies, but the

buyer and seller roles are played by different business entity objects. As we shall see in a moment, this role shifting speaks of the presence of another type of pattern: structural patterns.

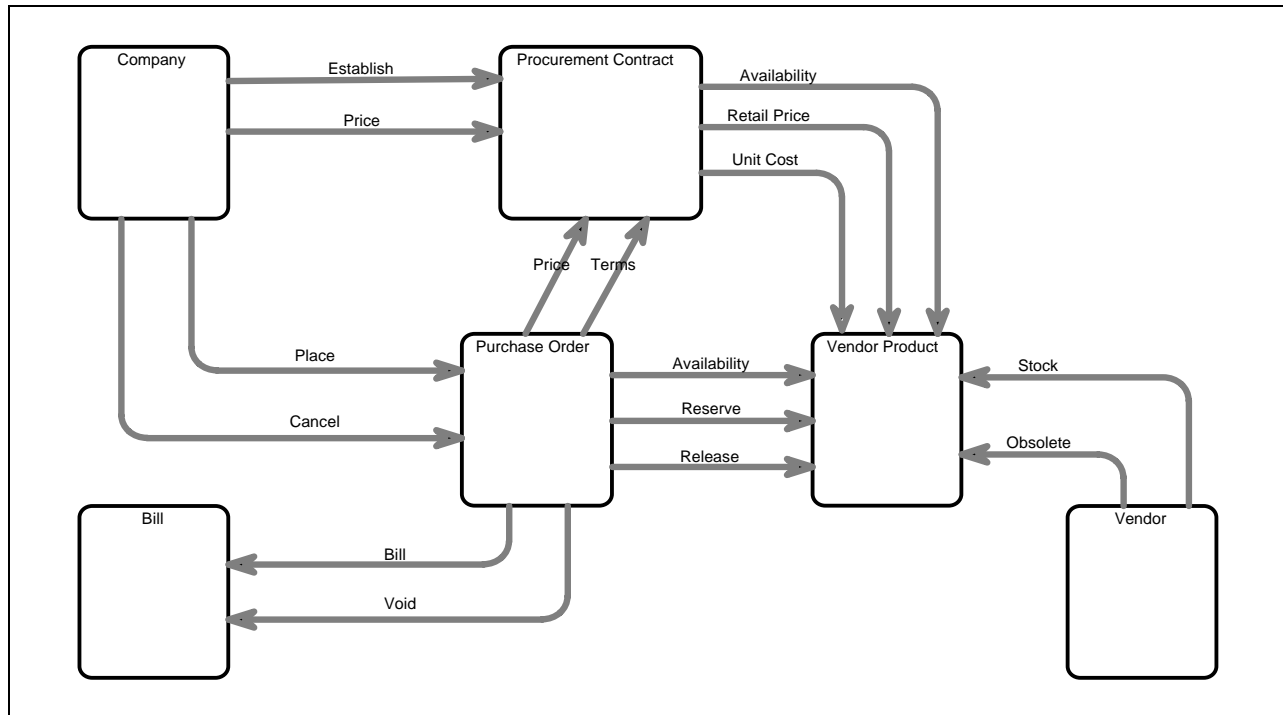


Figure 3: The procurement business process parallels order fulfillment, and is also rooted in the order pattern.

A **structural pattern** is a repeating set of business entity objects in type, role or composite structural relationships. Some structural patterns are industry specific, like the *financial instrument* or *insurance product* type hierarchies. Others are more generic, like the roles in the *legal party pattern*. (See Figure 4) Structural patterns define whole sets of closely related things or concepts, including recurring specialization hierarchies. They often define the nature of and variation within complex roles, products and resources.

This example expresses the idea that a given instance of *legal party* (i.e. trading partner, party the company we are modeling can engage in a business transaction) can play roles of *customer*, *supplier*, *employee*, etc. One can find many real-world cases, for example the telephone company employee who is also a customer, or the surgeon under contract to (supplier to) an HMO who is also a member (customer).

A **semantic pattern** is a recurring set of entity object types and binary relationships that captures definition, meaning and constraints on meaning. Examples are numerous, like the airline capacity pattern of flight-segment-seat that also describes rail or ship transportation, hospital bed utilization, and factory assembly line capacity. This *capacity-time pattern* captures utilization. Another common example is the *order-ship-bill pattern*. (See Figure 5)

From a business modeling perspective, two points about patterns are important. First, patterns are everywhere. Experienced data or process modelers will recognize some of the patterns we have discussed. Experienced business object modelers will see an even greater number of patterns, along with crucial binding between pattern types (i.e. behavioral-structural, etc.) The most

common inhibitors to finding patterns are poor naming practices, the absence of clear definition of model artifacts, and insufficient abstraction. Patterns happen.

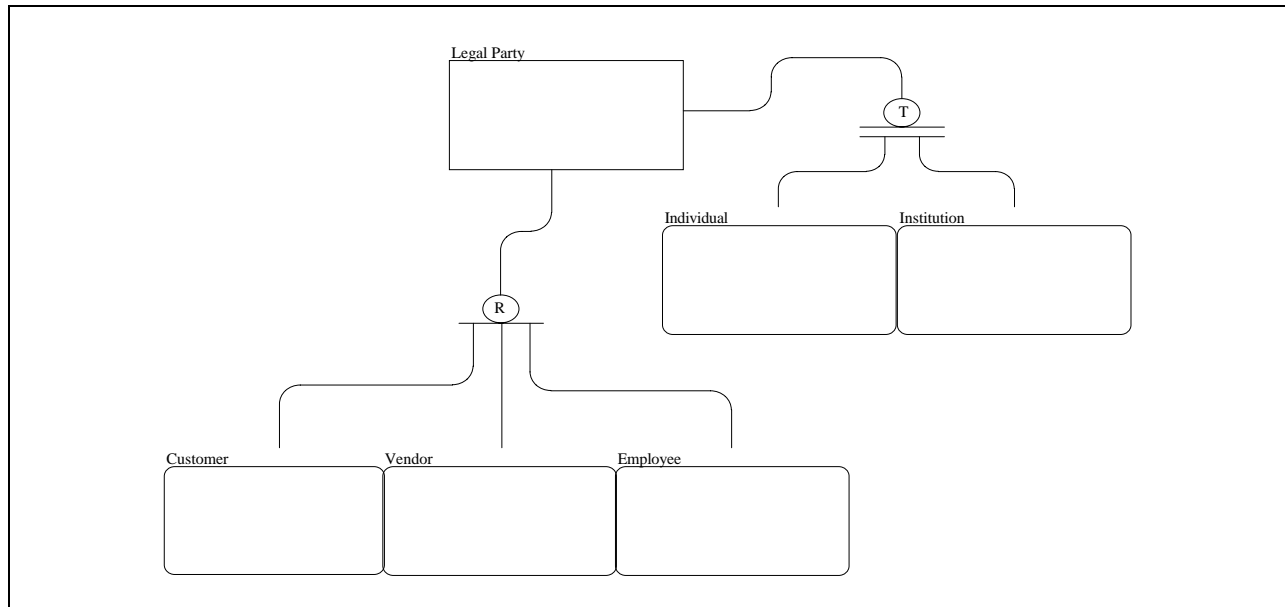


Figure 4: The legal party pattern is elemental to most business models, and provides key roles required to operate the exchange pattern.

Second, patterns greatly simplify business modeling. The shortest route to a sound business model combines business patterns with the ideas (above) about an inventory or list of standard business process, entity and event objects. To the extent that patterns such as those above occur in every business, the modeler's challenge is to find and understand rather than invent.

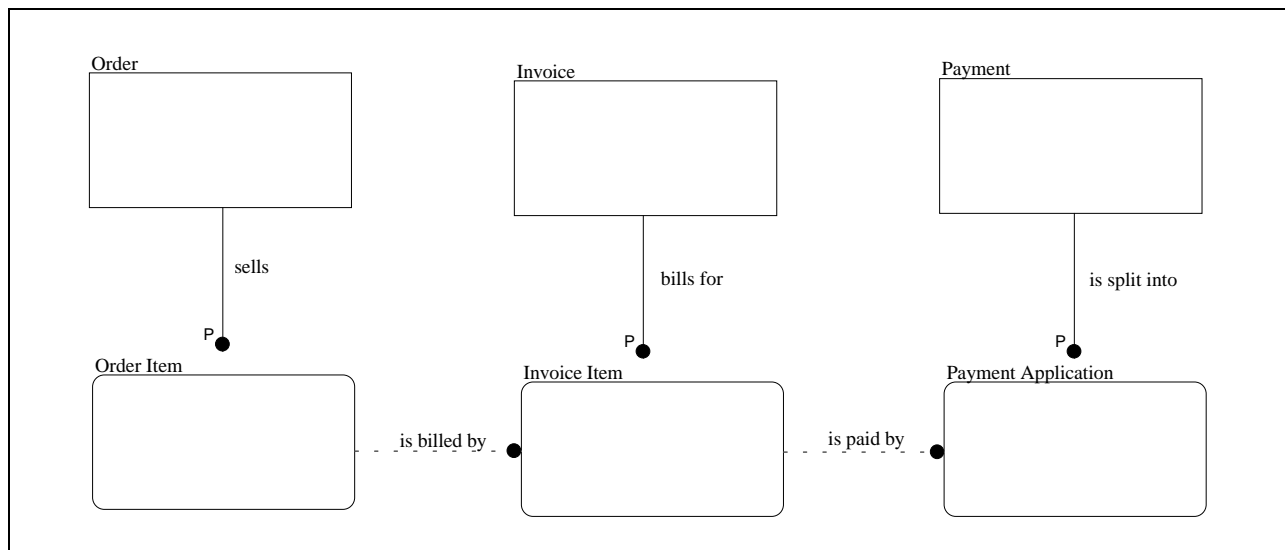


Figure 5: The order-ship-bill pattern reflects fundamental relationships between units of sales, delivery and billing

Frameworks

To move from rapid business modeling to flexible software manufacturing, one must deliver models and software almost simultaneously. The practical solution involves the integration of business models and software architectures based on those business modelsⁱⁱ. Thus the idea of a business framework.

A *business framework* is a prefabricated generic software architecture based on a business model. A business model is comprised of many patterns. By configuring the model, the framework user (often but not always an IT technician) effects a parallel change to the software. Thus by adapting the pattern-based business model to a specific company, the software is also adapted. The result: a business model that can be installed and run in client-server applications.

Business frameworks provide the highest available degree of leverage for the business modeler. They also enforce the greatest degree of rigor on business modeling thinking -- because the framework exists from "concept to code", there is a very strong incentive in terms of time, cost and value to work within rather than against the structure and semantics of the framework. Translation: frameworks won't be popular with those bent on invention and "winging it". Frameworks are for the disciplined. The upside, however, looks to be a 10:1 or greater improvement in cycle time once the framework has been integrated with the business.

The idea behind flexible software manufacturing is that the business model directly drives software business components. Only by synchronizing delivery of software components with business modeling can this be accomplished.

Conclusions

Standard parts are a first step toward a highly leveraged business modeling and software delivery environment. Patterns group parts into business-meaningful fragments. These fragments, like subassemblies in a manufacturing product, simplify the rapid assembly of larger, more complex business models. Patterns move business modeling out of the realm of invention, making it a configure and assemble activity -- less black art, more engineering. Flexible software manufacturing, however, requires yet a further advance: business frameworks. Frameworks combine business models and prefabricated software delivery architecture. The business model and its software implementation are available off-the-shelf.. Using frameworks, both business modeling and software delivery can move from invention to specialization and assembly.

Robert E. Shelton is President & CEO of Open Engineering, Inc., a San Francisco-based consulting firm specializing in object technology and business engineering. He is responsible for Object-Oriented Business Engineering™, and the firm's strategy for Business Object Management. Mr. Shelton is founding Chairman of the Object Management Group's Business Object Management Special Interest Group (BOMSIG), and former Editor of the *Hotline on Object-Oriented Technology*. If you have questions or comments, he may be reached in San Francisco, CA, at 415-989-9050, fax: 415-989-9055, or email: rshelton@openeng.com.

Oobe is a trademark of Open Engineering Inc.

ⁱⁱ Note that this does **NOT** mean that the business models and the software become the same thing!