

Understanding business process modeling

Learn what it is, how it works, and why it's a must for any organization

Skill Level: Introductory

[S. E. Slack \(sally@sslack.com\)](mailto:sally@sslack.com)

Author and Business Transformation Consultant
Studio B

19 Aug 2008

Learn the basic concepts of business process modeling and apply them in your organization so you can speak for the user and be a more effective business analyst.

Before going too deeply into the concept of *business process modeling*, I think it's a good idea to have a quick recap of process steps and process management. As I mentioned in "[Manage the process, not the steps](#)," all processes consist of actual tasks that must be completed for any business process to work properly. These tasks are called *process steps*.

Process management, then, is the management of the activities involved in confirming that procedures are in place to ensure that those tasks get done properly. Part of process management involves creating process designs, outlining roles, and using strategic planning to determine how and when all those tasks are accomplished. That's where business process modeling comes into play. It lets business analysts play an active role in defining the "to be" process and helps them think of processes in new and different ways.

Business process modeling is a discipline in and of itself. Business analysts typically perform the duties involved in it: defining and outlining business practices, processes, information flows, data stores, and systems. In other words, process models essentially lump processes of the same type into a model so that you can see how those processes can work together. When you understand the activities your organization typically performs and the kind of information it needs to

successfully perform those activities, you're in a much better position to analyze how your organization's processes work together overall.

A business process model lets you visually understand whether modifications must occur or whether current processes are working just fine. As you see the processes in action, you can tell whether your organization is carrying out its processes in the most efficient way and how effectively it is supporting the workforce. That's it for the general concepts: Now, let's take a closer look at things.

Think like a user

Let me repeat the key aspect of business modeling: *lump processes of the same type into a model so you can see how those processes work together*. This is an important fact to understand. For some reason, a good many business analysts try to make the process more difficult than it is. They might try to lump dissimilar processes together, for instance, to prove that an entire architecture needs an overhaul. They bite off more than they can chew doing something like that, and they run the risk of alienating potential allies during the process. Business modeling isn't rocket science: It's a straightforward process with the ultimate goal of understanding whether modifications are needed in a specific process and, if they are, what those modifications should be.

As a business analyst, when you start thinking about creating a business process model, you must first define what an overall process is trying accomplish. *Then* look at how that process fits with other processes. By taking apart pieces of the whole and putting them back together, you can get a much better idea of how the whole works.

Take, for example, a human resources (HR) recruiting process. The result of the recruiting process might be to hire the most qualified person for an open position, right?

That overarching process, however, probably integrates with processes in advertising (how do we find these candidates?) and payroll (how much can we pay for certain positions?), for example. Higher-level processes are concerned as well, such as how this position contributes to the overall revenue of the company.

Business process modeling helps you look at all those processes—including the steps involved in each process—so that you can see whether the current process (also known as the *as is* process) is working. At the same time, modeling a business process helps you see whether changes to any part of those processes could result in a better outcome. This look-ahead is called the *to be* process.

As you work with business process models, it's critical that you look at the processes from a user perspective instead of a system perspective. There's a very simple

reason for this: If a system fails, users must still know what to do in the process. As I mention this, I'm reminded of a recent experience at McDonald's, the worldwide fast-food restaurant. I worked at McDonald's as a teenager—lots of kids do. One of the benefits of working there was that I learned how to quickly count change and add up purchases.

During a recent visit, however, I noticed that the teenager running the cash register wasn't required to do any of those things. The cash register did all the work for him; it even told him how much change to give me. After he closed the cash drawer, he was completely at a loss what to do when I realized he had accidentally given me too much in change. The computer had moved on, waiting for the next new customer, but the teenager still had me to deal with. He couldn't open the drawer or figure out how much extra money he had given me, so he finally gave up and told me to keep the extra change.

My point is that this teenager had no understanding of the monetary process the computer was handling. In turn, the process had no understanding of what the teenager was dealing with. He simply doled out the money as directed. When the process failed, he couldn't determine how to fix the error, because the entire process was a system process, not a user process. In the end, his solution to the problem was to lose money for McDonald's.

When you model processes, think of this poor, frustrated kid at McDonald's. He wanted to do the right thing, but the system-focused process wouldn't let him do it quickly and easily. The result? Lost revenue. This can happen in a million different ways at a million different organizations if you forget to model processes from a user perspective. It's critical, then, that business analysts never forget that important element of business modeling: the user. That element starts with the requirements, and it should be kept at the top of your list throughout the business modeling process.

Business process modeling steps

Business process modeling can be handled in many different ways. However, there are some basic items that are generally captured after business process requirements are gathered:

- Model business items
- Model roles and resources
- Model services
- Model policies
- Model key performance indicators

Whatever you decide to capture in your business model, stay consistent and be thorough. Watch for alternatives and exceptions to standard procedures, too. You can build a business model for almost any business objective—it all depends on your scope of

interest and expertise.

Speak for the user

As a business analyst, your biggest role is to speak for the user—to create processes that work *with* users instead of against them. When you're modeling new business processes, it's imperative that you intimately understand what that user will experience with any change in the process. Writing user stories and use cases will help you do this and will also help you essentially speak for the user when the business modeling process gets underway. Although some say that user stories and use cases are technically pieces of the requirements process, I'm going to address them in this section.

You can create use cases to help you understand the big picture of what users are dealing with. From there, you go into more detailed user stories to help define smaller pieces of functionality if you need to, or you use acceptance tests to help capture all the minute details of the process and ensure that they'll work as you intend.

Note: IBM Rational® Unified Process® (RUP®) is use case driven. If you're following RUP, stick with use cases only.

Let's look briefly at what each of those items is:

- **Use case:** A method for capturing the overall business process
- **User story:** Extremely useful in capturing smaller picture details
- **Acceptance test:** A system test that is the last opportunity a business analyst has to be certain that processes are running as expected

Use cases can conceivably offer you some of the most important information about users and processes as you work through the business modeling process. Some people don't see it that way, but I don't see how you can even begin the modeling process without a thorough understanding of what your users are experiencing now—as well as what they should or will be experiencing in the future. Use cases are written by business analysts, so if you're not familiar with them, you should become so.

There are both informal and formal versions of use cases and user stories. Informal versions stick with simple bulleted lists, while formal versions are extremely detailed and refer to specific user interface components. For the most part, formal versions are overkill, in my opinion. You can get the information you need with informal versions, and you don't have to take as much time to create them. Now, here are two quick examples—one showing how to write a very high-level but informal use

case, and one showing a more detailed, informal use case, both for a hiring process:

- Example 1: Very high-level, informal use case:
 - Job opening
 - Applicants interviewed
 - Applicant selected
 - Offer made
 - Offer accepted
 - Applicant begins work
- Example 2: Standard, informal use case (not in its entirety):
 - Job opening
 - HR advertises the position and related qualifications.
 - Applicants apply for the job. Resumes are sorted according to qualifications.
 - Management selects applicants for interviews from resumes.
 - Interviews are conducted. Applicants are separated into two stacks: acceptable and not acceptable.
 - Management conducts second interviews for acceptable candidates.
 - Management selects one candidate to fill the position.
 - HR makes a job offer.
 - If accepted, background checks are conducted. If not accepted, management selects a second candidate.
 - When a candidate accepts the offer in writing, a start date is set.
 - Badges and other official work documents are generated.
 - Applicant begins work on the start date.
 - Benefits and vested dates begin.
 - Payroll records are generated.
 - First paycheck is sent.

Both use cases are useful; the more detailed case is obviously more beneficial to someone attempting to understand all the process levels involved in hiring someone. When you model a business process, you want enough details to help you

understand the activities undertaken during the process and what it takes to successfully complete them. If you need to know, for example, which paperwork must be completed to generate payroll records, include that in your use case.

What a business process model should include

When it's time to actually model the business process, there is no single way to do it correctly. A business process model can take on many, many different looks. It all depends on the number of scenarios you want to test and the types of processes involved. The bottom line for a successful business model, however, is that it capture the noteworthy events, inputs, resources, and outputs associated with a given business process. From there, you can build test and use cases, then connect them back to the business model and narrow down issues to specific functional elements that can be improved.

As I mentioned earlier, a business process model has a broader and more inclusive range than any software system being considered. As a result, an effective business model lets you clearly map the scope of the proposed system as well as pieces of the process that must be implemented in other ways, such as manual processes that the system can't handle. Additionally, a business process model should be in the type of format that *all* stakeholders can understand; if you're using a tool that's highly technical, you'd better be certain your CEO can make sense of it.

Ultimately, a business process model should define at least the following elements (an outstanding tool will define more):

- The goal of the process
- Specific inputs and outputs
- Consumed resources
- Activities and the order in which they are performed
- Significant events that drive or affect the process

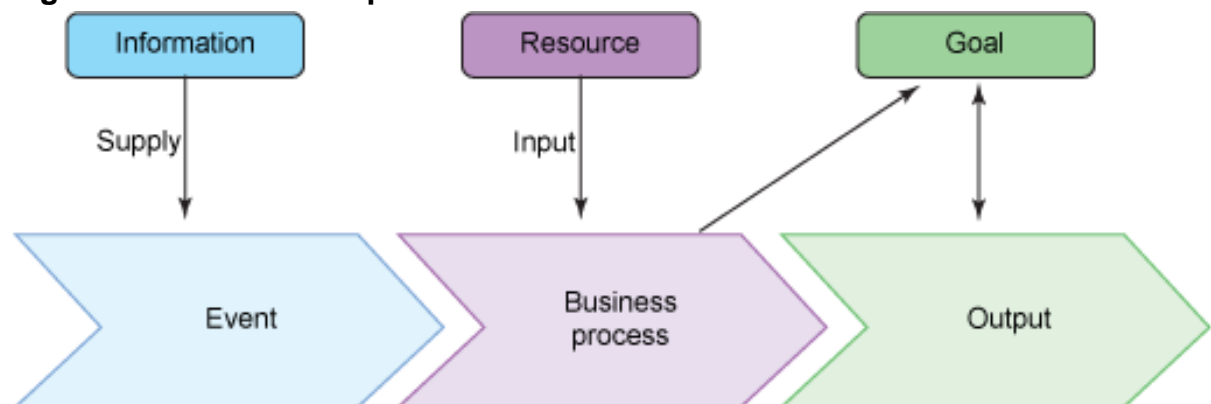
I have used a few terms in the previous list that you might not be familiar with, so let me do a bit of explaining on a couple of items. *Inputs* and *outputs* often refer to data, but they can also be something that is processed or worked upon. For example, a system requires data input and typically outputs data in return. A machine, however, typically needs raw material as input; its output might be a finished product. And for humans, input and output can be in the form of documents, information, or materials of some kind. Think in terms of items that provide value to a process, and they will typically be some form of input or output: a report, a completed customer order, and so on.

Resources are usually some sort of input to a process that is often consumed during the processing. Here's what I mean: Information about bus service is a resource. As each bus service runs, actual times and details about the service are recorded in the process. When the service is complete and the details recorded, the resource is exhausted—there is no more resource available, because the service is finished.

Events can be any number of things. Maybe a significant event is the receipt of an item on a loading dock, or perhaps it's a deadline. It doesn't really matter. It just needs to trigger something that initiates a business process. The receipt of an item on the loading dock might trigger a customer delivery service, for instance, or a deadline might trigger the start of an invoicing process.

Ultimately, all these model elements are going to come together to give you a complete picture that might look something like Figure 1.

Figure 1. The business process model



Your business model, of course, will be far more detailed and have information on many different processes. But from Figure 1, you can see how easy it is to see any particular process at a glance and understand what's going into it and what's coming out of it. When you have a problem with a process, you just can't beat business modeling to help you find and resolve the issue.

Get the tools for the job

If you're in the requirements stage, check out IBM Rational RequisitePro®. It's integrated with Microsoft® Office Word, so if you use Microsoft Office products, you'll see a familiar environment. It also helps you write strong use cases with a database infrastructure that lets you easily organize requirements, trace them in processes, and analyze what needs to happen in a business process.

Unified Modeling Language

A universal, general-purpose modeling language, Unified Modeling Language (UML) lets users create an abstract model of a system in a graphical format. You'll probably hear the term tossed about when

you're working on business process modeling. It can be used with all processes, most implementation technologies, and throughout a development life cycle.

Essentially, UML lets you convey information that's tough to describe in written formats. It doesn't require any particular methodology or work products, so it's a fairly easy way to create use case and other types of diagrams you might want to use in business process modeling. IBM Rational products help automate the use of UML. A free trial version of IBM Rational Software Modeler is [available for download](#).

When you've got all your requirements and have an idea of how things need to develop, it's time to get into the actual act of business modeling. You can capture a business process very easily using a simple text-based document, or you can use a basic graphics program like IBM Lotus® Freelance Graphics® to show a process flow. You can also go with a robust business modeling tool, like IBM WebSphere® Business Modeler. In the [Resources](#) section, I've provided a link to an excellent article that explores each of these tools in depth.

A business modeler, as I and others will tell you, is the most effective, productive tool you can use. First, a tool like WebSphere Business Modeler allows users of all types to see the information they want to see by using the modeler in different modes. You can visualize, understand, and document business processes in an easy-to-follow manner, which lets nontechnical users, like most business analysts, get the basic understanding of the process model while architects and engineers can delve into great detail on the technical pieces.

These tools also offer collaborative options, version control, and easy access—typically through a Web browser. An outstanding business model tool will, by the way, give you information on roles, resources, and other details linked to specific processes. It will also let you quickly redesign a process and ultimately move directly to execution. In short, it does all the work for you after you enter the information.

Watch out for a business modeling tool that doesn't give a good indication of flow direction or one that uses shapes or graphics inconsistently. A decision point in a process, for example, should always be shown as the same shape. A poor program will use different shapes to indicate decision points at different places, which leads to confusion about the process.

The other thing that's extremely helpful in a business modeling tool is the ability to simulate the actual process. Simple tools stop at flow diagrams or process maps, which are fine and very useful, but when you can actually see a process being simulated, it adds an entirely new dimension to your understanding of the process and how it will affect users.

Admittedly, this article touches the tip of the iceberg for business process modeling.

It's meant to give you a general understanding of the concept. You owe it to yourself to dig deeper and read up on the topic. With business process modeling, you can get into as much detail as you like for a process, and that can be extremely helpful for many business analysts. I highly recommend the resources noted—and if you think you're going to get very far into the topic, consider downloading a WebSphere Business Modeler demo or trial.

Modeling the business process is a critical element of any software-development process. Without it, you won't really have the understanding you need to design processes effectively. With it, you can help your organization effectively achieve business goals. Don't get discouraged if it seems complicated at first. The more you work with the concept and begin to apply it to small areas of your organization, the more you'll recognize the benefits it can bring to the people who need it most: the users you speak for.

Resources

Learn

- Read "[Why model business processes?](#)" (Marc Fasbinder, developerworks, May 2007) to better understand the three major types of business modeling tools.
- Learn to write better use cases by reading [Use Cases: Patterns and Blueprints](#) by Gunnar Overgaard and Karin Palmkvist (IBM Press, 2004).
- If you're a business analyst, read [Learn business process modeling basics for the analyst](#) (Ken Beck, Joshy Joseph, and German Goldszmidt, developerWorks, Feb 2005).
- Learn more about UML by reading [UML basics: An introduction to the Unified Modeling Language](#) (Donald Bell, developerWorks, Jun 2003).
- Take a [tutorial on business process modeling](#), and learn to use IBM WebSphere Business Integration Modeler.
- Read [WebSphere trials and demos](#) to get your hands on business modeling tools from WebSphere.
- In the [developerWorks Architecture zone](#), get the resources you need to advance your skills in the architecture arena.
- Stay current with [developerWorks technical events and webcasts](#).
- Browse the [technology bookstore](#) for books on these and other technical topics.

Get products and technologies

- Download [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from DB2®, Lotus, Rational, Tivoli®, and WebSphere.

Discuss

- Get [The Rational Edge](#), the e-zine for the Rational community.
- Participate in the [IT architecture forum](#) to exchange tips and techniques and to share other related information about the broad topic of IT architecture.
- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

About the author

S. E. Slack

S.E. Slack is a writer and author with more than 10 technology books to her credit. She resides in Colorado with her family. Her latest title on the shelves is *Windows Vista: Home Entertainment with Windows Media Center and Xbox 360* (Microsoft Press, 2007). Her next book will be *PowerPoint 2007 Graphics and Animations Made Easy*, to be published by McGraw-Hill.

Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Freelance Graphics, Lotus, Rational, Rational Unified Process, RequisitePro, RUP, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.