What’s In, What’s Out?  
Thoughts on Scoping Models

As further proof that I’m a bit process-obsessed, clarifying the scope of business processes has been on my mind these days. I’ve been doing a lot of process scoping on recent consulting jobs, so when the featured article in a recent BPTrends Advisor (Oct 09 2012 – http://bit.ly/WJ0cIk) was Paul Harmon’s “Scoping Processes,” it immediately caught my eye. This excellent article described Roger Burlton’s format for a Scope Diagram, a “one-pager” that, not surprisingly, clarifies the scope of a process. These are also widely referred to as IGOE Diagrams or IGOE Charts because they are based on the IGOE framework (Inputs, Guides, Outputs, Enablers) originally popularized by the IDEF0 modeling method. Roger’s format is extremely popular with business process professionals around the world who rely on it to succinctly convey the scope of a business process and some of the key factors that influence it.

Despite the many virtues of the IGOE Diagram, I use an alternative format for illustrating the scope of a business process. It depicts a little more information on the scope and contents of a business process (the “what,”) and less on analytic factors like guides and enablers (the “how” and the “why,”) which I leave until later. The differences are subtle, not radical, and I’m certainly not claiming that my format is superior. In fact, Paul’s Advisor was a reminder that I need to reincorporate some elements of the IGOE chart into my consulting practice. Nonetheless, there are enough differences that when I read Paul’s Advisor I thought “I can squeeze a column out of this topic too!” Let’s have a look at the format I use, the rationale, and some of the benefits.

Some of my earlier Columns cover related concepts like “big picture” scoping and clarity around “what is a business process?” Some that you might want to check out include:

- “Some Thoughts on Process Discovery” (September 2009 – http://bit.ly/Y3K6BZ ) includes some of the examples used in this Column and describes where scoping fits into an overall methodology

Why Bother?

The opening sentences of Paul’s Advisor were, “Too many process analysts talk as if flow diagrams, like BPMN, are the most important diagrams a business analyst can use. In fact, in most cases, a process analyst would be better advised to begin with a Scope Diagram, which will provide much more valuable information.” I couldn’t agree more. The most frequent errors I see on business process change initiatives are related to scope – failing to explicitly clarify the scope of the process right at the outset, and/or failing to identify true, end-to-end business processes. To be explicit, by “end-to-end” I mean “from the earliest triggering event through to the final results expected by each process stakeholder.” More on this later.

Instead of concentrating initially on scope, the tendency is to immediately leap into process
mapping and analysis. That’s understandable, given the nearly irresistible allure of swimlane diagramming – we love our boxes and lines! – but a very small amount of time (typically only an hour or two) invested up-front on scoping will avoid two major headaches later on:

1. Process mapping bogs down in discussions of where to begin and end a flow diagram, or whether or not certain activities (and their performers) should be included. You’ve probably heard the objection “That’s not part of this process.” Or realized that because process scope wasn’t well understood you don’t have the right people in the room. It’s better to have the scope conversations up front without the added complication of simultaneously modeling flow. No doubt, modeling process flow will inevitably refine your understanding of process scope, but it’s far more productive to start off with the basics clearly (and visibly!) established. One of my mantras is “flow first, details later.” It occurs to me I should update it – “scope first, then flow, details and analysis later.”

2. A more insidious danger arises when an organization does “process improvement” on what is actually a piece of an end-to-end business process, a process fragment or subprocess. When a subprocess is “improved” without an understanding of the whole, the sad result can be the “local optimization yields global suboptimization” phenomenon Eliyahu Goldratt warned us about. I see this time and again, in organizations that really should know better. It’s a sad commentary that over 20 years after Michael Hammer popularized the notion of the “end-to-end, cross-functional business process” the point is still so often missed. A scoping model helps to avoid this by explicitly focusing on the true beginning and end of the process with questions like “Is that really the starting point?” and “Is that really the final result?”

To avoid these issues we need to establish the boundaries and contents of the process before going any further.

**A framework for process scoping**

The most fundamental aspects of process scope are:

- The “left” boundary: What is the true beginning of the process? What event(s) trigger it?
- The “right” boundary: What results signify successful completion of the process? What are the main stakeholders expecting from the process?
- The contents: What are the significant activities within the process? What are the five to seven milestones that must be reached between the beginning and the end?

That’s the essence of the framework I’ll be describing, “triggering event – subprocesses – results.” I might not use the term “subprocesses” right away with process stakeholders, opting instead for something a little less technocratic-sounding, like “milestones” or “activities.” Here it is in graphical form:

![Diagram](image)

**Figure 1: The essential framework for process scoping: trigger – activities – results**

At this point, you might be thinking “That’s just Input – Process – Output. It’s just part of the IGOE framework.” True, but bear with me – there are some subtle but significant differences I’ll be getting to. Let’s begin with a sample I use in one of my workshops, based on work at a large manufacturer:
Figure 2: A simple Process Scope Diagram (or Process Framing Diagram)

The essential framework comprises four parts. I’ve numbered the components 1 through 4, but the order in which you discover them is likely to be different, perhaps starting with “results,” and in any case is highly iterative. They are:

1. Trigger(s) (or triggering event(s))
2. Subprocesses (or “milestones,” “essential activities,” or whatever term you prefer)
3. Results (the “output,” if you prefer)
4. Cases (or variations)

Pretty simple, isn’t it? That’s one reason it works so well. There are other variables we can and will add, including participating organizations, supporting systems, and other enablers, but only after these essentials have been agreed.

The underlying philosophy

The other reason this format works so well is that it is purely a scoping model, not an analysis model. At this point, we’re NOT trying to figure out how the process works, or why it works the way it does. We’re just trying to agree on scope – what are the boundaries and contents of the process.

A fundamental skill for business and process analysts is the ability to separate the “what” from the “who and the how.” This diagram supports that by emphasizing “what triggers the process?,” “what are its essential activities?,” “what are its results?,” and “what are the main variants of the process.” There’s no reference to how the process is designed or supported, nor any mention of who the actors are who carry out the activities in the process. Because of this, it’s visually simpler (not simplistic!) compared to an IGOE chart.

Simplicity was my objective when I first, out of necessity, started putting together this format. I was running process mapping sessions that would run into the difficulties I described earlier, and needed a simple diagram I could post on the wall to remind people, “at a glance,” what the scope of the process was. After some experimentation, what emerged was this form of Process Scope Diagram, or, as I often call it, a Process Framing Diagram. A companion diagram, the Process Landscape, was also essential and will be described later.

Of course, we’ll get to the “who” and the “how” and the “why” soon enough, using other models and frameworks. We’ll provide an overview toward the end of this Column of when and how these are used, and explore some of them in more depth in the next Column.

The details

I’ll offer a few guidelines, observations, and examples for each of the four components.

1 – Trigger(s) or Triggering Event(s)

Events should be a familiar concept – the execution of a business process (an “instance” of the process) is triggered by some business event. One of the problems that you’ll see if you look at enough process scoping models following the IGOE framework, going right back to the original IDEF0 Inputs-Controls-Outputs-Mechanisms framework, is a lack of consistency on whether a triggering event is a Guide or an Input. Similarly, different practitioners will treat certain resources
(people, for instance) as either Inputs or as Enablers. Do a little Googling and see for yourself. This is why I stopped doing IDEF0 diagrams back in the mid-1980s – they were just too tough for mere mortals to be consistent with – and it’s been many years since I’ve seen one in the wild. In the case of Process Scope Diagrams, there’s only one process box, so the inconsistency doesn’t matter as much. Nonetheless, I choose to keep things simple by treating the triggering event as its own unique component.

It’s generally agreed that there are three primary kinds of events:

1. Decision or action event – a person or organization decides to do something that can’t be predicted in advance. Examples include “decision to open an account” and “decision to apply for a job.”

2. Temporal event – the arrival of a pre-determined date and time. Examples include “time to pay the staff,” “time to prepare quarterly financial statements,” and “time to deactivate client account.”

3. Condition events – a process that monitors either the environment or stored data detects a condition that is outside of a predetermined range. Examples include “Equity price falls to ‘sell’ threshold” and “Reactor temperature exceeds maximum.” Yikes!

The challenge here is to determine whether the earliest triggering event has been identified. There’s a tendency to assume that the process begins when there’s the equivalent of a “knock on our door” when in fact it begins earlier. For instance, at a PC manufacturer, the initial assumption was that the “Resolve Service Incident” process began with the event “trouble ticket is recorded.” By repeatedly asking “what happens before that?” we were able to determine that the actual triggering event was much earlier, when “customer detects issue.” This was important, because the most broken parts of the process were before what was originally thought to be the starting point, and wouldn’t have been identified if we hadn’t pushed to discover the real triggering event. There’s more subtlety in this area than I can get into in this Column, but if you keep pushing for the earliest triggering event, you’ll probably be fine.

2 – Results by Stakeholder

At the other end of the process are the results (or “outputs” if you prefer) that tell us when successful completion of the process has been reached. It’s important here not to confuse goals or objectives with results. A goal or objective is typically some business performance target, such as a desired level of profitability, customer retention, or rate of product introduction. The results we’re looking for are discrete and uniquely identifiable. For instance, the result of “Acquire Customer” is that “a new Customer is acquired” – each individual Customer can be identified, and therefore they can be counted. We can ask “how many Customers did we acquire today, this week, or this month.” Each process result must meet this criterion.

Also significant is that a single process will generally deliver multiple results, often results that would otherwise be seen as coming from separate processes. As a starting point, I always assume that the customer of the process and the enterprise that owns the process each receive a result. Other stakeholders, including process performers (e.g., employees,) business partners, regulators, and others might receive a result. The examples we’ve seen already illustrate the “multiple results” idea.

3 – Subprocesses (or Milestones or Main Activities or…)

This is the “what’s in?” part of the framework. The intent is to depict, “at a glance,” what the main activities are that make up the process, which will save a lot of back and forth later on. As I’ve described in the earlier Columns referenced at the outset, our field struggles with inconsistent terminology, so (with some trepidation) I’ll recap how I use the term “subprocess.”

A subprocess is like any other process in that it has a trigger (mostly, the previous subprocess,) some activity (steps and decisions,) and results that typically become the input to the next subprocess. The key is that the results delivered by the subprocesses aren’t usually the final
results, they’re intermediate results that represent important milestones between the triggering event and the final results the stakeholders expect. (Of course, those final results will be delivered by the last one or two subprocesses.)

Let’s consider a few important points relevant to determining subprocesses:

1. In the early going, I seldom use the term “subprocess” – most stakeholder groups (the people who manage and work within the process) find it more intuitive if I speak in terms of “milestones” or the “main activities.”

2. You don’t want to produce a list of 20 subprocesses, because that will fail the “at a glance” test. Think in terms of “the magical number seven plus or minus two,” which is the maximum number of concepts an average person can hold in their “working memory” simultaneously. I usually simplify it to “five +/- two,” as in “what are the roughly five key milestones between the triggering event and the final results?” In a print advertising example, the milestones were:
   - We have an agreed order for a display ad (subprocess: Take Display Ad Order)
   - We have an ad ready to show the client (subprocess: Design Display Ad)
   - We have approval or feedback from the client (subprocess: Obtain Display Ad Approval)
   - We have an ad prepared and submitted for publication (subprocess: Submit Display Ad for Publication)
   - We have collected payment (subprocess: Collect Display Ad Payment)

   These are illustrated in Figure 3.

3. The subprocess names contain absolutely no reference to “who or how” – for instance, there is no indication whether the Customer submits an order via the Web, or whether it is obtained in person, on a paper form, by a Sales Rep.

Figure 3: A Process Scope Diagram from the print media sector

To demonstrate the value of highlighting the subprocesses, consider the earlier “Fulfill Customer Order” example and two of its subprocesses – “Manufacture Order” and “Collect Payment.” Each of those cleared up significant potential confusion:

- Some participants were surprised by “Manufacture Order” because they remembered when, years ago, the company followed a “manufacture to inventory” model as
opposed to the current “build to order” model. They expected to see a “Pick Order” subprocess involving warehouse staff.

- Some participants were surprised by “Collect Payment” because they assumed the company still ran on a monthly invoicing cycle, not the current method of collecting payment for each individual order.

Once those details were sorted, there was a general feeling that we weren’t showing the true scope of the process, even though the boundaries (trigger and results) were apparent, because we were “hiding” key milestones and activities in the “Manufacture Order” subprocess. Scoping discussions led to the second version of the subprocesses, shown in Figure 4. It has 9 subprocesses, which is right on the upper limit of how many I want to depict.

You might think that according to the rules, a major process you are studying has 27 subprocesses, but a 1:27 ratio from process to subprocesses is too big a leap for people to make. Theoretical purity isn’t the issue, communication is. You have to find a way to make the transition into the detail that works within the 7 +/- 2 paradigm.

Speaking of theoretical purity, you’ll notice that we’ve adjusted the layout to illustrate that moving the work in process (“Transport WIP”) is a subprocess that occurs repeatedly to support the other subprocesses. It isn’t perfect, but it got the point across. That’s another example of emphasizing communication.

---

**Figure 4: The refined “Fulfill Customer Order” subprocesses**

4 – Cases (or Variations)

Processes are typically named with an active verb and a noun, such as “Fulfill” (the verb) and “Display Ad Order” (the noun.) Think of a process in terms of an action (verb) on an object (noun.) We can add some useful precision by inserting a qualifier between the verb and the noun, for instance “Fill New Display Ad Order,” “Fill Repeat Display Ad Order,” and “Fill Camera-ready Display Ad Order.” These are the cases or variations of the process.

Identifying the cases not only enhances everyone’s understanding of what’s in scope, they also make process flow modeling manageable – as I outlined in an earlier Column, life is MUCH simpler when you model one case of a process at a time.

Note that a case is different than a scenario, which looks at a particular instance of a process with specific named actors, data values, decision outcomes, and results.

Adding a 5th – who?

After these four questions are answered, the next step is to produce a Process vs. Function Chart that illustrates the major organization units (internal and external) that participate in the process. Now we’re venturing into “who?” These are a personal favorite of mine, and we’ll spend more time on them in the next Column. In the meantime, Figure 5 offers an example.
Questions and objections – the usual suspects

Those who are used to seeing the IPO or IGOE frameworks will typically raise some questions. Actually, they’re raising objections, but “questions” sounds so much nicer, don’t you think? Let’s go through the four most common ones, and bring up some points about the basic nature of business processes.

1 – “Everyone thinks in terms of Inputs and Outputs – why don’t you?”

The “Input-Process-Output” framework has been with us for so long it’s second nature to think of a business process in those terms. In fact, in a very recent BPTrends Email Advisor, Paul Harmon wrote, “Most people seem to agree that a process transforms inputs into outputs.” (“Resisting a Narrow Definition of Process” – BPTrends Advisor Jan 29 2013 – http://bit.ly/WJ0WY)

That’s true – in many settings, looking at a process as “a way to transform inputs into outputs” is quite natural. In both discrete and continuous manufacturing, that’s often how the involved stakeholders naturally see things. And in the IT field, we’ve used that framework extensively for as long as I can remember. Over 30 years ago, I used HIPO (Hierarchical Input Process Output) charts to document system functionality as a hierarchy of modules. Of course, now we’d call them “components” or “services” – “modules” are so passé. And speaking of services, when I write up a service specification as part of requirements documentation, the Input and Output Messages, and the data Input (read) and Output (created, updated, or deleted) is an important part of the description.

So what’s the problem? Why not scope a process in terms of the inputs, the process, and the outputs? You could argue that the triggering event is just a kind of input (I wouldn’t agree), and the results are outputs (I would agree,) but the overall effect is quite different, for two main reasons.

First, in many, even most, settings this isn’t actually the most natural way to look at processes. The business people I work with in my consulting practice seldom think of their processes as “turning inputs into outputs.” Just try using IPO in a health care setting, where medical professionals bristle at anything that smacks of imposing a manufacturing perspective on their work. It’s the same in higher education, entertainment and media, financial services, criminal justice, retail, telecom, … well, you get the point. In just about every industry I’ve worked in (even manufacturing!) business stakeholders respond intuitively and positively to the “trigger – activities – results” framework and are noticeably less positive about “IPO” as a framework when they’re given a choice.

Of course, later on, when we’re analyzing a process, we’ll absolutely describe the various inputs and outputs along the way, just not when we’re initially defining the scope of that process.

The second reason not to use “IPO” is best explained by first considering the popular “SIPOC” variant, for “Supplier – Input – Process – Output – Customer.” It’s an extension of the IPO
framework that is widely used, and, surprisingly, frequently introduces some particular difficulties. I noticed a long time ago that clients who were devoted to using SIPOC continually identified as complete “business processes” what were actually “subprocesses” – pieces of the whole that result in a particular intermediate milestone. This leads to the “local optimization” phenomenon we mentioned earlier. This outcome of using SIPOC is to be expected, because most subprocesses fit the structure perfectly – they take information or physical items (usually) as Input from the previous subprocess and do something to it (that’s the “Process” part) to deliver an Output. Those Outputs become the input to the next subprocess, and so on. In fact, analyzing those connections is one of the bottom-up techniques I regularly use to help teams identify complete, end-to-end processes.

By the way, that’s one reason I don’t advise my clients that every process is part of a larger process, and every process is made up of smaller processes, even though it’s true. I introduce specific terminology – a Process Area or Process Domain is made up of Business Processes which in turn are made up of Subprocesses. (You could keep going into Activities, Tasks, Steps, etc. but I don’t try to be as precise at the lower levels.) I do this to stress that there’s something called a Business Process that has specific, objective criteria, and that addressing less than a Business Process (e.g., a Subprocess) has to be done very carefully. In the same vein, I don’t use terms like Level 0 Process, Level 1 Process, and so on because they’re too ambiguous.

So, in summary, IPO is a perfectly reasonable framework at lower levels, but I’ll stick with “trigger – activities – results” for the end-to-end business process.

2 – “Hey, isn’t that just a decomposition?”

Very observant! Yes, it absolutely is a two-level decomposition (Business Process into Subprocesses,) and adding the Process Domain above its Business Processes would make it a three-level decomposition. The key points are:

1. We’ve added valuable context by depicting the triggering event and expected results, and...

2. We haven’t drawn it drawn as a tree, the way an organization chart usually is.

That second point is subtle but important. If you draw your process decomposition the way you’d draw an org chart, it immediately gets people thinking in terms of (you guessed it) the organization chart. That’s a problem because one of our key motivations is to eliminate any reference to “who.”

This lesson was reinforced for me a few years ago at IRMUK’s Business Process Management / Enterprise Architecture conference in London. My friend (and fellow BPTrends Columnist) Roger Tregear had put up a slide that showed a business process decomposed into subprocesses and so on. Even though it was abundantly clear that Roger’s example was, indeed, a true business process, one of the attendees objected – strenuously! He insisted that Roger had missed the whole point and was actually depicting organization structure, and he wasn’t to be convinced otherwise.

So, layout matters, and I find that laying it out as I’ve shown, left to right, trigger to result, just works better.

3 – “Is that really all you need to clarify scope?”

Definitely not – some sort of context diagram to (surprise!) put the process into its context is also essential. The IGOE diagram illustrated in Paul’s “Scoping Processes” is an excellent example of a context diagram, and is reminiscent of our old friend, the “Level 0 Data Flow Diagram,” which shows a process and its interaction with “External Entities.” The form I use to illustrate process context takes a different approach, and shows a family of related processes on a “Process Landscape” or “Overall Process Map.” The idea is very simple – for any one of the processes that you might choose to study, it shows what processes are not within the scope of your study. It’s Project Management 101 – it’s just as important to show what’s out of scope as it is to show what’s within scope.
We looked at these in the Columns I referenced earlier. Figure 6 is a Process Landscape that puts the “Fill Display Ad Order” process into context.

![Figure 6: A Process Landscape (or Overall Process Map) for a print media company](image)

**4 – “Why don’t you use Guides and Enablers – don’t you think they’re important?”**

Absolutely they’re important. In fact, I think understanding the guides and enablers of a process is ultimately what process analysis is all about, and it’s the essence of my methodology. I just think, per the previous sentence, that understanding them is part of process analysis, and I prefer to leave that until after process scoping and (as-is) process description.

I don’t distinguish Guides and Enablers – I treat them all as Enablers in a framework that is conceptually very similar to the famous “Burton Hexagon.” I define enablers as “factors that can be manipulated to impact the performance of a business process.” Like the hexagon, the framework I use includes six enablers and was described extensively in “Disabled by Enablers, Punished by Rewards” (May 2012 – http://bit.ly/Z4DvMk ) My equivalent of “guides” (factors from “above”) are “Mission, Strategy, Goals, and Objectives” and “Organizational Culture,” topics I wrote about extensively in my recent Columns.

Here’s the overall flow I follow:

- First, discover a business process, or more likely, a family of related processes and depict these on a Process Landscape.
- Then, establish what the scope of a selected process is using the trigger, subprocesses, results, and cases framework and draw a Process Scope Diagram.
- Then add who at an organizational level by drawing a Process vs. Function chart.
- Then discover how the process actually works, by drawing some sort of Process Flow Diagram such as a swimlane diagram.
- Only then, once we have an agreed picture of the reality of the process, do we investigate the enablers.

The reason I wait is that people are surprised, even stunned, when they’ve completed mapping the end-to-end flow and can see the reality of “who does what, when.” This inevitably reveals
factors like inappropriate policies (“guides”) and ineffective systems (“enablers”) that they might not even have been aware of otherwise.

**Wrap-up**

This Column has gone on for far too long, so a big “thank you” if you’ve stuck with me until now. I’ll close with the observation that the whole point of the approach I’ve described here is to separate the initial Process Scoping from subsequent Process Analysis. It’s worked for me (and others) in a wide range of enterprises around the globe, so even if you’re a bit skeptical, I encourage you to give it a try.

Next time we’ll continue with this topic, looking at how I add “who” and “how” to my Process Scope Models, and then at how they can be extended to become very fast and effective Analysis Models. This has been really useful in recent years because organizations might not have the time or the patience for as-is process flow modeling. In fact, there might not even be an as-is flow (think “Adaptive Case Management”) so working with a variant of the Process Scope Model might be your only choice.

**Where I’ll be over the next few months**

- June 11 – 13 2013 I’ll be back in London for IRMUK’s BPM conference. [http://bit.ly/13BRnN2](http://bit.ly/13BRnN2) My one-hour conference session will be “Flawed Thinking in Processland – When Doing the ‘Right Thing’ Isn’t.” This promises to either delight or annoy depending on your position on the topics I’ll look into. And, returning to the subject of this column, my half-day session will be “Running a Successful Process Scoping and Mapping Session.”


I hope our paths cross at one of these events.

From the Trenches
Alec Sharp

**BPTrends Linkedin Discussion Group**

We created a BPTrends Discussion Group on Linkedin to allow our members, readers and friends to freely exchange ideas on a wide variety of BPM related topics. We encourage you to initiate a new discussion on this publication, or on other BPM related topics of interest to you, or to contribute to existing discussions. Go to Linkedin and join the [BPTrends Discussion Group](http://www.linkedin.com/groups).