

Thinking Processes, Capabilities and Services What's in a Word?

10/07/2014

Howard Smith

Everything is a process. Don't let anyone tell you otherwise. Everything else is just the name of a specific process design.

When we walk into our office, sit down, open our email, start to write, answer the phone, and especially when interact with colleagues, we are doing "work". However, we may or may not be working productively. We all know colleagues who simply react to events. They appear to have no plan or goal. Yet another person, sitting right next to them, may be churning out results. As well as getting more work done their output may also be of higher quality. The difference between the two workers is that one of them is following a process.

The person following a process may not know that they are doing so. They could just have hit upon an efficient way of working, via their own instincts and intelligence. In which case, their process is implicit. However, by closely observing the way that they work it is often possible to discern the underlying process they are following. What is implicit then becomes explicit. It can be written down!

The same principle of process discovery applies to teams, departments and to whole enterprises. Processes describe "how work gets done around here". They describe the order in which work is done. They describe how the work to be done is allocated and flows around the organization. Process descriptions also describe the opportunities that exist to perform tasks in parallel, as well as the essential activities that must be executed in series. A good process description will, in addition, include the pre- and post- conditions for moving from one activity to the next. And, since all work involves multiple participants, the process will necessarily describe the coordination points among everyone involved. We know all this instinctively. We know what processes are, even if we cannot always see them or write them down.

Formal process notations have been developed so that if you want to clearly set out an explicit process, it is possible to do so. When we draw such diagrams and models we set out who is doing what, when, why, and how. We make our model and description as rich, or as simple, as appropriate for its use: explaining the process to someone, or something, else, or for analysis. In some cases we choose to draw formalized process maps as part of the input requirements for the design of a supporting computer system.

Since all processes involve multiple participants engaged in A choreographed activity, there will always be a high degree of parallel activity. This is why we use swim lanes or pools to illustrate them. Indeed, parallel work is the most efficient and scalable (by adding resources) and should be the norm. A need to take serial steps, or to impose coordination points, all too often just serves to create bottlenecks in the work flow. This is true of both human systems and automated systems.

Process v capability

Modern computing systems are dominated by parallel distributed processing. Likewise, all efficient organizations try to reduce the need for serialism. We all know the pain from: “The form is with Jane. We cannot do anything until she is finished. John cannot start work until our team has done our bit”. Such situations are the result of poor process design, and letting “legacy” work structures become institutionalized bad habits.

If we can draw a process, explicitly, we can take a fresh look at its design. This will often reveal several ways to turn serial work into parallel work, amplifying the productivity of the team or organization and reducing cycle times all round. Therefore as organizations improve their processes they gain useful *capabilities*, and their capacity to do work increases. The outputs of that work become more reliable, more complete, of higher quality and of greater value to customers.

Remember the chaotic, reactive, ad-hoc worker at the start of this Article? He is not a capability because he cannot be replicated, understood or improved. Only processes can do this.

Capabilities are processes. Point to a capability and any good BPM practitioner will be able to show you the process at work. No process, no capability. Naming the process is not good enough. Unless we can see the process we cannot explore alternate ways to perform the work. Capabilities cannot be improved without the discipline of process modelling and process design improvement.

Processes describe the dynamics of a work regime, whether among people, systems or the internal of the computer. They are descriptions of work to be done. No matter what style of process modelling you adopt, it is self-evidently the case that a process describes all movement and change in the world around us. This contrasts sharply with other forms of modelling.

Look closer

Think about a chicken’s egg. If we decompose an egg and find that it is made up of a shell, a yolk, egg white, and an inner and outer membrane, we are developing a compositional model. The vast majority of modelling techniques on the planet are compositional models. By contrast, how a newborn chick emerges from this structure is a process. Similarly, how work emerges from a team is a process. The composition of a team, or organization, is not a process, nor is the team a capability until they follow a process. The same is true at all levels of organizational design.

When we say that a department has the capability to do something or achieve something, we are saying that they use a process to get the necessary work done. Moreover, we can understand it, and improve it if necessary. Those organizations that appear to work wonders and achieve a great deal, as if by magic, are using good processes, whether they know it or not. While it sometimes the case that you need to work hard to see their processes and their logic, you will eventually find their secret sauce.

If we can accept that processes are the only way to describe and create capabilities, what of the word “service”? Are services also processes, or something else?

When someone does something for someone else they are providing a service. Once again, a process is involved and required. The capability to provide any service depends on the associated process. It might be a good process, or a bad process, but it is a process none the less. It might even be a deliberately hidden process – described only via a service level - but it is always there.

Since the vast majority of processes among teams and across organizations depend upon someone doing something for someone else, service processes are literally everywhere. The same is true of all modern computer systems and the interfaces between applications. Indeed, this is the origin of the term Service Oriented Architecture (SOA). Similar ideas dominate in the field of Application Programming Interfaces (APIs). Services are the interface processes between those people or systems involved in an overall (end to end) process.

Using the example of the compositional model of an egg above, the boundaries between the system elements (shell, yolk etc.) could be said to form a set of service interfaces between those elements. But once again, this would only tell us what services exist, not how those services are used to “get work done in the egg” and so hatch a new chick. For anything remotely like that to happen, i.e. for a change to occur, we need a process.

Process design and service design are closely related. It is not possible to have a process unless the participants in the process are able to exchange information. Indeed, *interaction* between participants (the exchange of information or goods) lies at the heart of every process. It is this interaction behavior that, in the minds of many experts, defines the very essence of what it means to point to something and say: “Look, there be a process!”

Every time we think about how to describe a service we smack right up against process design. Only the language of process has the vocabulary to describe change in a system.

The relationship of services to processes has a final twist which is important in this story. When someone, or something, does work on behalf of someone, or something, else, a boundary is created between them. A new service is created, if only for the duration of a single interaction. The service receiver need not, in practice, understand how the work was performed by the service provider. This is as true for loosely-coupled computing systems as it is for the macro economics of the IT outsourcing (ITO) and business process outsourcing (BPO) industries. Individuals, teams and entire organizations are able to adopt business services without worrying about the “internal” process behind them. There is an inside, and an outside, to every process. And that’s really all a service is, one view of a process.

A service provider may wish to hide internal process details during delivery to a client, but you can be sure that the explicit internal process design will rear its head if the service repeatedly fails to meet expectations or contracted service levels. This is true even in our everyday lives. If we are unhappy about a service we receive at home we naturally start to wonder about what process lies beneath? This underlines once again the central role of process in understanding the nature of all work (by humans, by systems or in combination).

What's in a word?

Due to the legacy of inflexible computer systems of the past, it has become fashionable in some quarters to try to avoid using the word "process". The idea going around is that "processes" are the antithesis of "agility" and the "dynamic nature" of "human work". "Processes", it is argued, lead to rigidity, fixed actions and narrow pathways, limiting options, flexibility and creativity. These are old ideas and they are dead wrong.

While you can draw a rigid process if you want to, no one is forcing you to do so. Processes can be as fluid as they need to be. Fluidity can be described. Modern process languages permit it. Let's take some examples. Beneath all Web services are processes. All knowledge workers use processes, whether they recognize them or not. And processes dominate even in the "creative" industries. Ask a film director working on set or with a digital post-production or visual effects unit. These highly skilled people are ultra-organized: by process. Processes can express these and many other kinds of human activity.

The fashion to try to put the word "process" in a box arises from some peoples' past experiences with workflow technology and workflow systems. Those IT applications were only able to execute a certain kind or style of process. But this does not mean that other forms of process do not exist. An analogy: A factory that makes biscuits cannot make cars using its process, but that does not mean that factories that can make cars do not exist. They do!

The painful experience with workflow systems has led some experts to equate the word "process" with "workflow". This leads them to say things such as, "Mr. Client, you don't need a process system, you need adaptive case management". Yet surely we know that "adaptive case management" is nothing more than a specific process design – i.e. how the work of case management is done. How could it be anything else?!

Trying to box process in and equate it with step-by-step systems is a fiction which denies all modern thinking about process in theory and practice. "Adaptive case management" is just one type of process, a name for the process inside. Open the "adaptive case management" box and what you'll find is a process. Draw it out and you'll be able to understand what flows in, what flows out, what happens in between, among which participants, in what order, with what serialism and parallelism, and where the coordination points and junctures and decisions lie.

Let's be clear: 100% of an "adaptive case management system" can be described quite clearly in just a few process diagrams. And guess what, if we were to draw those diagrams formally using an executable process notation we could bring them to life on a suitable computer simply by using the 'Save As' menu. Hey presto! We'd have a new adaptive case management application.

My conclusion from both theory and practice is that all work that can be understood and improved is process work. All capabilities are processes. All services are processes. All other names we use are simply pointers to specific kinds of processes. We can look under such names and reveal the process beneath.

Author



Howard Smith is CTO (Europe) for CSC and a Research Associate with the Leading Edge Forum (LEF). Howard has recently developed CSC's methodology for Services Innovation, based on Services Blueprinting and the Southbeach Notation.

A sought after speaker, advisor and problem solver, Howard's expertise spans early stage technology search, software engineering, systems integration, business process, services innovation, ideation and futures scanning. He is the author of two books: *Business Process Management: The Third Wave* and *IT Doesn't Matter - Business Processes Do*.

Contact Howard via LinkedIn at <https://www.linkedin.com/profile/view?id=9859>