A BPTrends Column October 2013

Business Architecture

Mike Rosen
Chief Scientist
Wilton Consulting Group

Mike.rosen@WiltonConsultingGroup.com

Use Context Diagrams to Bridge from Value Streams to Processes and Services

In a business analysis or architecture scenario, one of the key artifacts to produce is the value stream. One of the earlier references to value streams comes from fellow BPTrends author Ralph Whittle who defined it in his Enterprise Business Architecture book as: “A Value Stream is an end-to-end collection of activities that creates a result for a customer, who may be the ultimate customer, or an internal user of the value stream. The value stream has a clear goal: to satisfy or to delight the customer.” To avoid these machinations about customer, I usually just define a value stream as “an end-to-end view of how value is achieved for a given stakeholder”.

Clearly, one of the important aspects of a value stream is that it focuses on the delivery of value and a range of stakeholders. When creating these artifacts, we purposely keep the value stream high level, not digging into the details of each individual activity or getting bogged down in alternate flows and error paths. There will be plenty of time for that later. But at first we just want to get a high level overview of stakeholders and values. Soon, we will discover many-to-many relationships. Specifically, the same stakeholder will be involved in more than one value stream, and a given value stream will have many participants.

From an enterprise perspective, we want to understand all of these relationships, in other words, we want to see all the value streams and participants together for a given set of cohesive business scenarios. Often, these scenarios will be related to a product or service, or some other fundamental business entity. For example, in an insurance company, there is a value stream for a customer acquiring an insurance product, for a customer changing their address, for development of new insurance products within the company, and more. We can easily imagine the interdependencies of these different streams. Each of the value streams is made up of stages (or activities), and each stage is supported by a capability, which is implemented by one or more processes, and requires some information. Again, from an enterprise perspective, the same information is required by multiple stages, and the multiple stages may require the same capabilities. Often, when we dig down to the next level of analysis, we discover redundant information and processes are being used to implement similar capabilities in related value streams. Figure 1 illustrates these relationships.

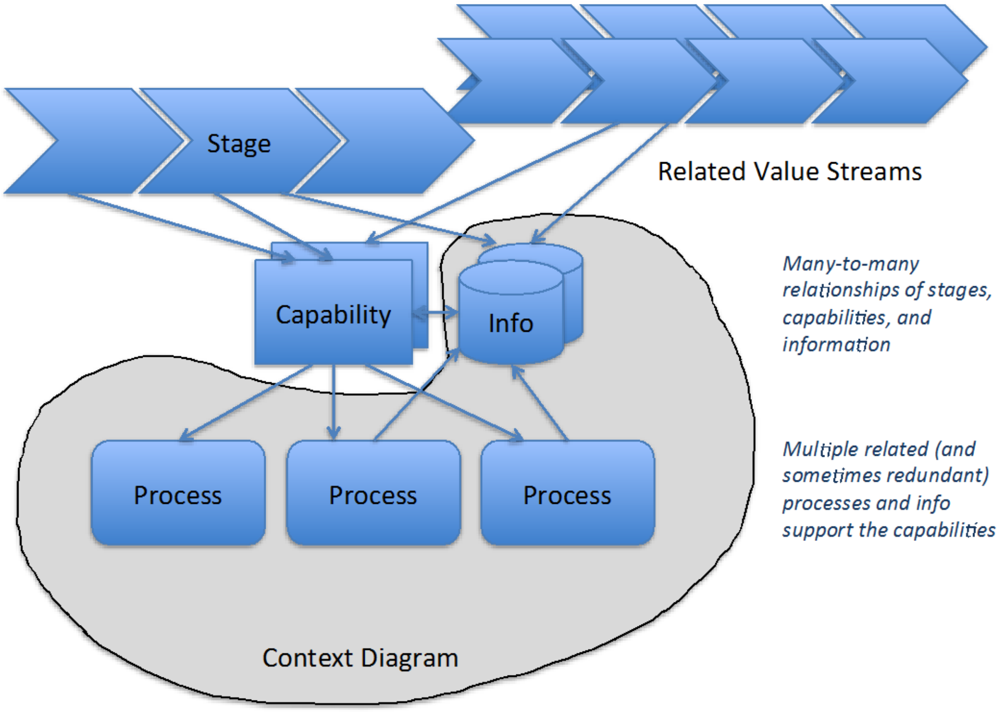
Stage
Related Value Streams
Capability
Info
Process
Process
Process
Context Diagram
Many-to-many relationships of stages, capabilities, and information
Multiple related (and sometimes redundant) processes and info support the capabilities

Figure 1 – Many to many relationships

Business Context Diagrams

If we're looking to improve consistency, streamline processes, remove redundancies, and enable integration, we need to know how all the different processes and information fit and work together. This is where the context diagram comes in. Figure 2 shows a simplified sample context diagram for telecom customer provisioning and billing.

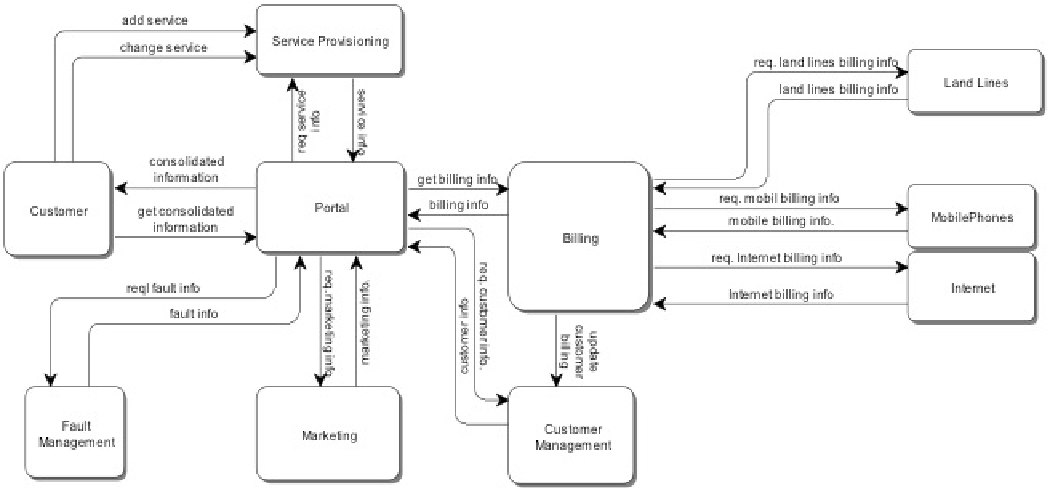

Figure 2 – Sample Business Context Diagram

As with any type of architectural model, the context diagram is made up of concepts and relationships[i]. In a context diagram, these are:

- Actors/Roles (rounded corner rectangles) - entities that can send and receive messages. Examples of actors are:
  - Individuals (people)
  - Organizations (business, agencies, organization units, etc.)
  - Systems
- Messages/Transactions (arrows) - entities that represent communications between two actors. Messages are often referred to as *"transactions"* in business applications. Messages may be:
  - Documents
  - Packages
  - Electronic signals
  - Voice or other kinds of communications
- Objects/Subjects (implied) - A set of actors and messages define a discussion. That conversation is about "something". That "something" is an "object" unless it is a person when it is referred to it as a "subject". Examples of Objects are:
  - Product (objects)
  - Service (objects)
  - Parcel of land (object)
  - Customer (subject)
- Events/Time (not illustrated) - "points in time" that represent either sending or receiving a message. Events are characterized as:
  - Periodic (dates, daily, weekly, monthly, etc.)
  - Aperiodic ("ordering", "receiving", etc.)

We can think of a package as a metaphor for understanding the context diagram, as illustrated in Figure 3. The actors are the sender and receiver of the package. The package was sent at a specific time, indicating the event. The box is the message sent between the sender and receiver. The box contains something; the contents are the object of the message.

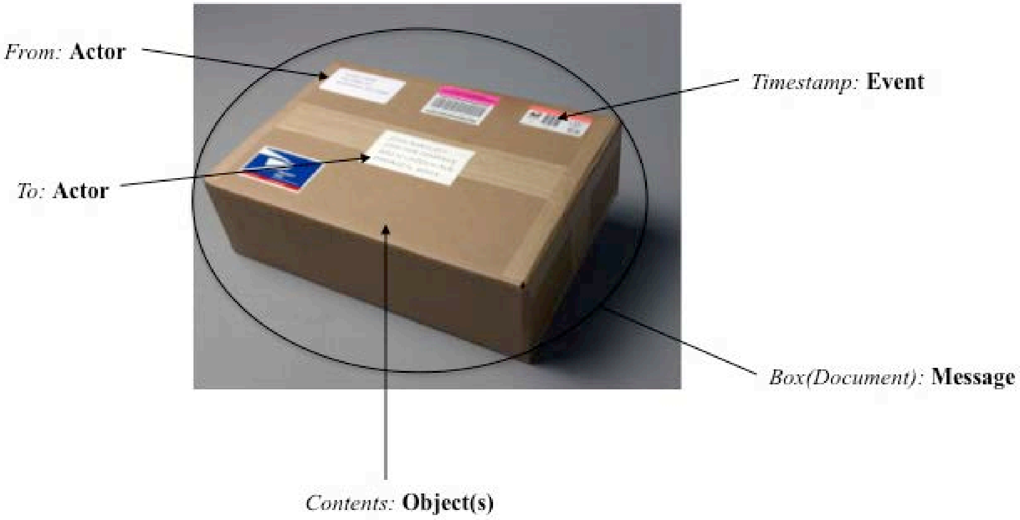

Figure 3 – A Package as a Metaphor

Returning to Figure 2, the rounded rectangles represent the actors and the arrows represent the messages. At this level of diagram, the actors are all the participants in the scenario including people, systems, organizations, etc. So, for example, the Customer actor (a person), send a ‘get consolidated information’ message to the Portal actor (a system) and receives a ‘consolidated info’ message in return.

Business Context Modeling Sessions

I have found the context diagram to be an excellent facilitation tool in engaging with business subject matter experts (SME). The model is easy to understand (perhaps after a 5 minute explanation) and easy to draw in real time. The typical session lasts 60-90 minutes and includes a white board, an architect or business analyst, and 3 or 4 SMEs. (2 SMEs is barely enough, 5 or more is too many). It goes something like this:

Architect: So tell me how this works…*silence…*okay, imagine I’m a new customer and I want to buy insurance, how would that work?

SME1: Well, the customer would go to our web site and fill out an application…

Architect goes to white board, draws two actors, Customer, Website, and a message arrow labeled ‘complete application’.

SME2: Maybe first the customer would go to the web site and browse the different types of products…

Architect goes to white board and adds message arrow labeled ‘browse product catalog’.

The discussion would continue around the different related value streams and stages until an aha moment reveals itself…

SME3: The underwriter looks at the customer risk profile to ….blah, blah, blah….

SME2: No, that’s not how it works. The underwriter does….

SME1: No, that’s not how it works either. The risk profile comes from the external insurance bureau…

Architect sits back and lets the SMEs sort it out, with subtle guidance if necessary.

This revelation almost always happens during one of the modeling sessions. The SMEs discover that:

- None of them actually know how it all works
- They are using the same term to mean different things (such as risk profile), or
- They are using different terms to mean the same thing
- There are several places where they’re doing the same thing, only differently or redundantly

- There are several areas for optimization or potential collaboration

All along, the architect is capturing the information on the whiteboard in the context diagram. After an appropriate amount of time, a follow up meeting is scheduled, the architect takes a picture of the whiteboard, goes back to the office to create an electronic version of the model, and distributes it to the SMEs. The follow-on meeting, usually 30-60 minutes, reviews the model and incorporates corrections and additions.

At the end of the two modeling sessions, the SMEs have learned something, identified opportunities, and ended up with a model they may not have been able to produce themselves. In other words, the session with the analyst or architect has provided value for them, making it much more likely that they will make themselves available to work with the architect in the future. The architect has also gotten value from the session because the context diagram identifies the actors, messages, objects, and their interactions, for all of the processes and information that support the related set of value streams. This provides the context that we need to drive to the next level of analysis and identify the individual processes and information.

Architectural Refinement

We rely on the process of architectural refinement to get to the next level of detail. Refinement is the process of transforming one model into another. Specifically, a higher-level model, or more abstract model, is refined into a more detailed (less abstract model). This is done by narrowing the scope of the model, and by adding new concepts and more detail. Remember that the context diagram represents all of the related actors, messages, and subjects, in other words, all of the processes and information required to support a set of value streams. Any individual business process is represented by a single path going through a subset of actors and messages in the context diagram. We refine the context diagram into a business process model by narrowing the scope from all process to a single process and then by adding the details of the sequence in which messages are sent, and what actions (tasks) are taken when those messages are received. Note that the actors and messages in the context diagram become actors (swimlanes) and messages in the process model. Figure 4 illustrates the refinement process, starting with value streams, which are refined into the context diagram, which is refined into the process models.

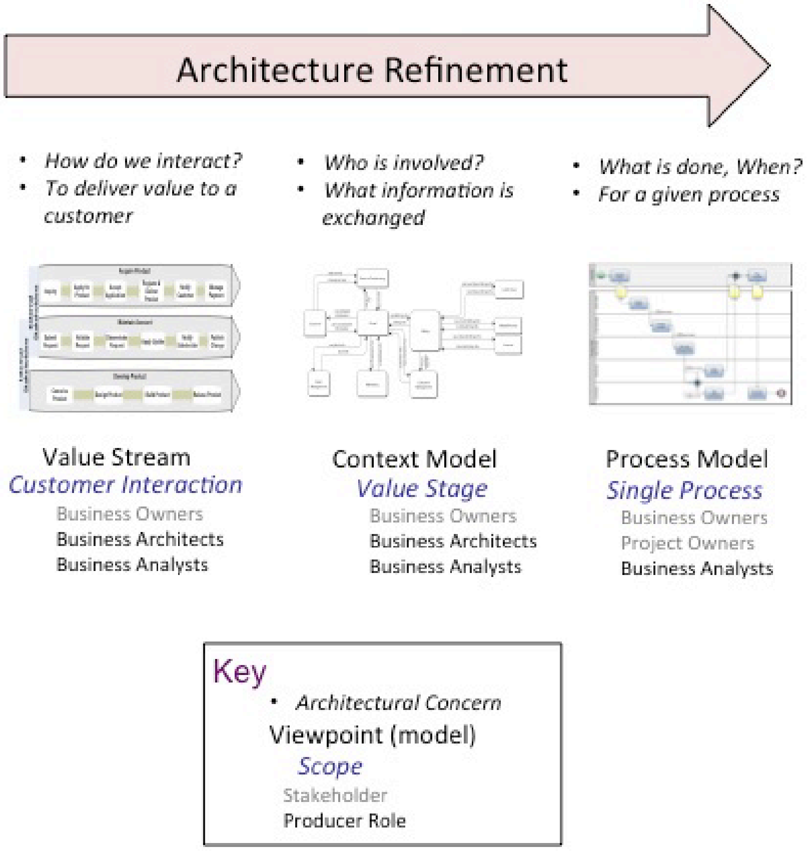

Figure 4 – Architectural Refinement Process

Information and Services

The context diagram has another extremely useful benefit. Because the context diagram contains all of the messages that are exchanged between participants, it also contains the source material for the Logical Information Model. To start with, every actor and subject in the context diagram should relate directly to the Business Information Model developed concurrently with the capability model. So, it immediately provides a consistency check with architectural work that has been done previously, and validates that the context diagram makes sense in the broader enterprise scope. Taking it to the next level of refinement, every actor, message, subject, and event in the context diagram will become an entity (or attribute) in the Logical Information Model that is used to support business process implementation.

Perhaps you will recall that for the past few years I wrote the SOA column for BPTrends where I described how a business process model leads directly to SOA design. In fact, the activities of a business process will be implemented directly by SOA services. More precisely, the tasks of a business process model (level 2 or 3) will be implemented by an operation on an SOA business service interface. In order for business services to be composed together into a business process, they need to share information. One of the things that the business context diagram identifies is ‘the information that needs to be shared between services for them to work together in a business process’. And this is done at the value stream or enterprise context. Conveniently, this is exactly the definition of the input and output documents of an SOA business service. In other words, there is one more level of refinement possible; the context diagram leads to the process model, which leads to the service design model.

Some of the challenges we face as business analysts and architects are:

- Putting things into a broader context
- Demonstrating the value of our work
- Driving analysis into design and implementation

The business context diagram helps address all of these challenges. I hope you find it as useful as I have.

[i] The formal semantics and package metaphor were originally developed in a collaboration between industry Pioneer Ken Orr (originator of Warnier-Orr diagrams), and Bill Roth, former Chief Architect for the State of Kansas.