
Exploring Services and Requirements – Part three

Author: Charles Edwards

Version: 0.06

Date: 25th July 2008

Abstract

As the architectural concept of Service Oriented Architecture (SOA) is implemented in many organisations, it is starting to introduce somewhat of a dilemma between the Business Analysts and Architects. People in these roles are wondering just how the Business Services, Business Processes and Use Cases all integrate and fit together.

Most systems are implemented as projects, with the scope limited to the associated requirements of the project. However these days, in order to maximise Service re-use, the business community in the enterprise have to think about and define their Business Capabilities and Requirements at more of an Enterprise Level rather than just at a particular System or Project level.

The aim of this series of papers is to explore the different semantics and terminology to align the areas of overlap between the various concepts, in order to suggest an overall simple integrated approach that works satisfying both the Business Analysis viewpoints and the Enterprise Architecture viewpoints.

Previously in part one and two

In part one we looked at the roles and defined the problems and challenges between Architects and Business Analysts for integrating Business Services and Business Requirements.

In part two we then defined the terminology and semantics around the concept of Services in general, with Business Services, Business Processes and Requirements in more detail.

What Enterprise Architects want from Business Analysts

In the new Enterprise Business Services (Enterprise Architecture) view of the world, the Business Architects want the Business Analysts to avoid re-inventing the wheel by defining the business services each time, in their own way, in differing notations, in different disparate documentation (Whiteboard, Visio, Word, Excel) on every project.

While in an Agile world there is absolutely no problem doing the envisioning using a minimal toolset, but the point is to go into a workshop and start from a known base-lined version of the Enterprise. Don't start from thin air every time. Rather work from a known baseline that describes the enterprise and modify existing Business Services, and even better, wherever possible, re-use existing Business Services when thinking about new, as well as refining, existing Business Processes and Requirements. How this is achieved, whether by using all the best Agile methods, or by any other methodology is irrelevant here, provided the final results end back up in the Enterprise Architecture as the new 'corporate memory' baseline.

This helps the business overall and all projects within the business for numerous reasons:

- By aligning projects to a known and proven business context.
- By starting projects from a known and proven business baseline. (then updating the baseline regularly)
- By refining the Business Architecture and where no Business Architecture already exists, by defining the Business Architecture so that the Business becomes more efficient, consistent and easier to understand.
- By making reuse decisions to refine the Business Architecture and thereby directly improve the business.
- This helps keep the Business Architecture and Services up to date, speeding up future project work.
- This aligns the Business Architecture to the Information Systems and Infrastructure Architecture.

Business Analysts come from the background where "BA's are there to define the business requirements and specify the business problem, not to try and specify any technical solutions." They have probably have had it

beaten into them over time by the Architects and developers! Business Architects and Business Analysts should not define how a Business Process should be implemented using Information System or Technical Services.

A Vision of collaboration – from a corporate memory

“You can’t manage what you can’t see and understand.” - Business Architecture fundamentally exists to make the architecture of the business visible to management so that collaboration can take place from a base-lined and centrally agreed place. So people can see what they are working with in more tangible terms and to collect and work from a ‘corporate memory’ defined for Business Services.

These days Business Analysts need to work with the Business Architects to specify Business Services so the Business Architects are in a position to see opportunities for re-use and to re-engineer Business Services, Events and Processes and help specify and optimise how the business is structured and behaves.

Technology is inextricably integrated into Business

The area between Problem and Solution has blurred and become a grey area in the last decade or so, because before in previous decades, there was an easy distinction between old manual processes (done by humans) and new automated processes (done by or helped by technology).

Before it was easy to differentiate a Business Process from System Processes by having a simple rule:

- If the work is done by a **human** it was a **Business Process** (or Business Use Case, or both),
- If work is done by a **computer or physical technology** it was a **System Process** (or System Use Case)

That assertion was never true, but was easier to understand and think about. Now it is just plain confusing. Technology and human processes are far more closely interwoven these days, so in defining how a business operates; the whole business needs to be modelled at a coarse grained level to define a logical business framework. This framework then helps set the context for any changes or new requirements that modifies these existing business processes. In many cases we assume certain technologies as part of the business mechanism.

Also the word ‘System’, can describe a process that when looked at in coarse grained way such as the process for the customer to Buy Groceries, would include both manual and automated elements. Take for example *Buy Groceries* in the figure on the right.

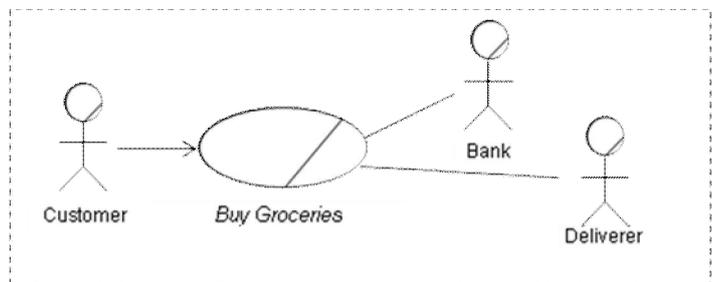


Figure 1 – Buy Groceries Business Use Case

There are varied business scenarios along a continuum (not all combinations included below) in which the *Buy Groceries* business use case can apply:

- Customer goes into a corner café / small shop, asks for goods from Teller behind counter, Customer pays cash to Teller who has a manual cash draw; Customer delivers their own groceries. (All manual no technologies help.)
- Customer goes into a garage shop/gas station convenience store, selects goods from the aisles, pays cash at counter electronic till; Customer delivers groceries. (Mostly manual minimal technologies help.)
- Customer goes into a small supermarket, selects goods from the aisles and puts them into a trolley, unpacks trolley onto teller conveyer; Teller scans and totals groceries; Customer packs bags and then pays debit or credit card using till online card payment machine; Customer delivers groceries. (Half manual and technology help.)
- Customer goes into a large supermarket, selects goods from the aisles, unpacks trolley item by item; Customer laser scans and puts items into bag; Customer uses debit or credit card online payment machine at till; Customer delivers groceries. (more auto technology than manual)
- Customer sits at a computer on the internet, logs into internet supermarket website, selects goods to order, checks out final order, selected date and time of delivery, pays using debit or credit card internet

payment, Customer waits for Deliverer to arrive at agreed time. (Virtually all automated from Customers point of view, some manual effort in unpacking at home – even that's being offered as a Service now!)

The whole transaction could vary and almost certainly be supported by various and mixed combinations of manual and automated technology business services such as:

- Goods selection and Trolley services,
- Picking, Packing, Delivery services and even unpacking services.
- Inventory services,
- Till conveyor services,
- Till barcode and laser scanning services,
- Till weighing and specials calculation services,
- Customer authentication services,
- Internet shopping cart and payment services.

All supported by more backend computer server services and digital communication and networking services than ever before. So when we come to enhance the existing systems, we cannot simply ignore the existing systems or technology. Business Services and Processes exist that are fully supported by technology. Technology and humans are fundamentally entwined in running a business.

So the new rule becomes (item 1 is concentrated upon in this paper, items 2 and 3 below not discussed here):

1. If the work is to be done (or is already done) by a Business it's described in:
 - *Business Domain Model* (the Business Domain Entities involved – static business structure),
 - *Business Use Cases* (a set each contains a visual *Business Process* - describes behaviour),
 - *Business Services* used (from a Business Services Taxonomy of all available Business Services) but adding to it
 - Providers and Consumers,
 - Interface Specifications and Operations,
 - Service Contracts, Service Level Agreements, etc.
2. If work is new (or already exists but much change) done by Software or an Application it's described in a:
 - *A SysML Requirements specification* (describing textual and visual static requirements),
 - *System Use Cases* (each has a behavioural UML Activity or Sequence diagram^[footnote1])
 - *Information System Services* used (from Info Sys Services Taxonomy of all available IS Services)
 - At a more technological level by Software Code and reverse engineered Class models (static)
3. If work is to be done (or already done) by Physical Infrastructure Technology it's described by a:
 - *Infrastructure Technology Design*
 - *Deployment Diagrams*
 - *Infrastructure Technology Services* used (from a IT Taxonomy of all available IT Services)

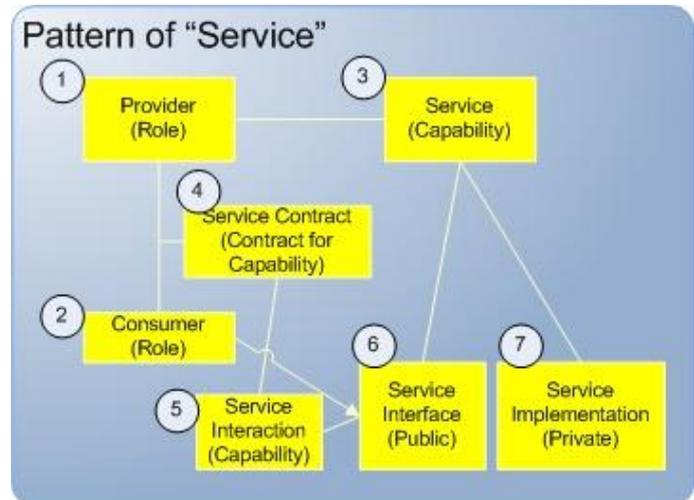
¹ This could be BPMN diagram too, as systems are generated from these into BPEL these days.

Comparing Patterns of “Service”

These concepts are not as far apart as we might believe. In this section we compare a Service pattern to a Business Use Case and Business Process using a Service pattern to see just how closely they align. This is thinking out loud and writing it down, so it could well be wrong! However in the spirit of the title “Exploring services and requirements”, this concept shows how the Service model structure, can be adapted for both the Business Use Case and the Business Process in general. Probably System Use Case too. Comments and suggestions are most welcome.

The key used in the diagrams that follow uses the notation:

- **Bright Yellow** to indicate where the concept is “active”, and
- **Light Grey** indicates where the concept is “passive” or “implied”.
- The numbers correspond to the text on the left of the diagram.



Pattern of a “Service”

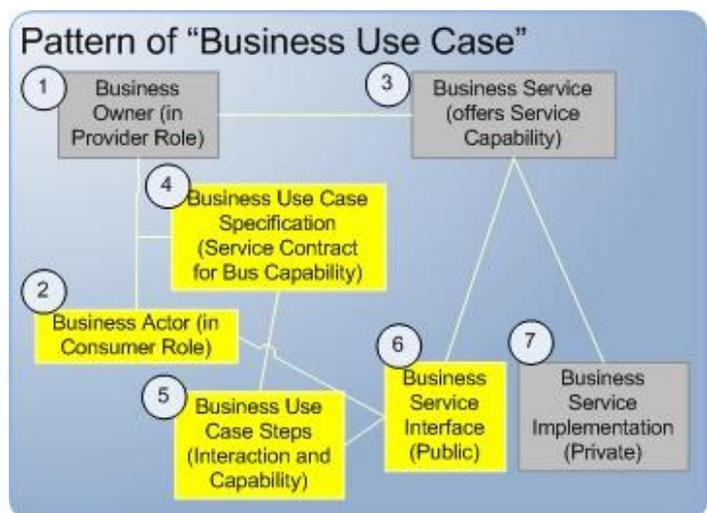
So in the previous diagram, using this logic a pattern of a Service has:

1. A Provider Role that **Provides** a Service.
2. A Consumer Role that **Consumes** a Service
3. A Service capability that is the **Service**
4. A Capability contract between Provider and Consumer that describes the **Service Contract** for the service capability.
5. A service interaction that describes the **Service Interaction** capability.
6. The Service capability has a **Public face** that is the Service Interface to the consumer.
7. The Service capability has a **Private side** that is the Service Implementation. Does the internal work.

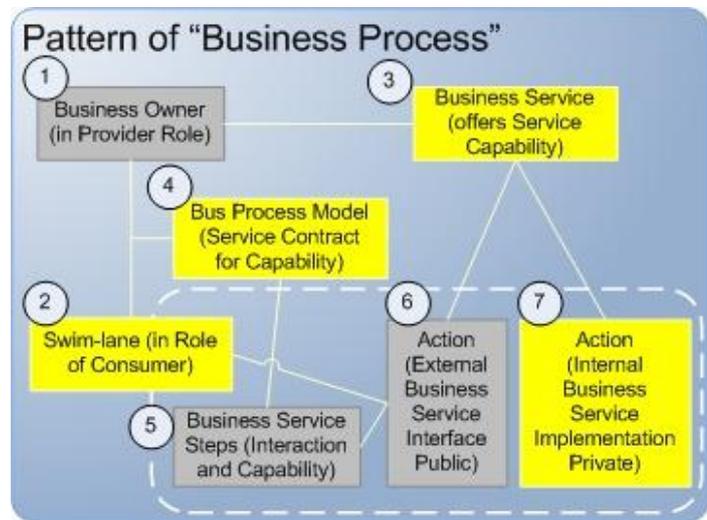
Business Use Case - in a pattern of a “Service”

So using this logic a “Use Case” in this pattern of a Service has:

1. An implied **Business owner** in provider role that owns the Business Service.
2. A consumer role that is a **Business Actor** to the Business Service that offers the capability.
3. An implied **Business Service** that offers a service capability that is described by the Business Use Case.
4. A service contract specification for capability – a **Business Use Case Specification** that describes the service contract for business capability.
5. An interaction capability that the **Business Use Case Steps and trigger Events** describe about the Service Interaction and Capability.
6. The service interface contract for capability is the **Business Service Interface** and has a Public face to the Business Actor.
7. An implied **System Implementation** that is the private side of the business service. What will be realised.



Business Process - in a pattern of a “Service”



So using this logic a “Business Process” in this pattern of a Service has:

1. An implied provider role that is the **Business Owner** that offers the Business Service capability.
2. A **Swim-lane** in role of business service Consumer.
3. A **Business Service** that Offers a Service Capability that is the Business Service the Business Process describes (one to one service per swim-lane)
4. A **Business Process Model** that describes the Service Contract for Capability.
*The white dotted line combines a Business Process/Service Action (5, 6, and 7 are all one and the same), that is **Action** steps in the Business Service. No concept of Public(6) vs. Private and therefore Service interaction = action (5), it's all just simple actions (7).*
5. An implied interaction with events and capability that the **Business Service Steps** describe the Business Service Interaction. (This would map to the Actor's Steps in the Business Use Case and trigger Events)
6. An implied external public Business Service interface. No real differentiation between **Public face** unless it calls another external Business process.
7. A Business System **Action** internal business system implementation. It is what will be realised. (This would map to the Internal Actors such as Warehouse, Shipping, etc.)

We have not repeated the System Use Case – as we think you get the point already, and it would be similar to the Business Use Cases with wording ‘System’ instead of ‘Service’.

So what does all that mean? Well let's explore this a little further using the same simple made up example defined in a previous series of papers entitled “Business versus System Use Cases” [^{Reference10}] where we showed a worked example of the ‘Buy Groceries’ Business Use Case.

Business Use Cases diagram maps to Business Services

Using our own made up UML profile for Business Service modelling (but using a similar notation for software system components and interfaces), we have defined a Business Service below for The Superstores Enterprise, Business Use Case called “Buy Groceries”.

The 5 items discussed below are referenced on the diagram:

1. How the **‘Buy groceries’ Business Use Case** maps on to the **“Buy Groceries” Business Service**. So from this we get a ‘black box’ or *external view of the Business Service ‘Buy Groceries’*.
2. The **‘Buy Groceries’ Business Service** has one **‘Buy Groceries’ Business Service Interface**.^[footnote2]
3. This Buy Groceries interface is made up of two operations **‘Place Order’** and **‘Confirm Receipt’ Business Service Operations**, which the Customer consumer can call on as required.
4. The **‘Customer’ Business Actor** in the Buy Groceries Business Use Case, maps directly onto the external Business Service **Consumer** called **‘Customer’** that interacts with our Business Service ‘Buy Groceries’.
5. Similarly the **‘Bank’ Business Actor** maps directly onto a **Business Service Provider ‘Bank’** that offers their ‘Authorise Payment’ Business Service to the business for our ‘Buy Groceries’ Business Service.

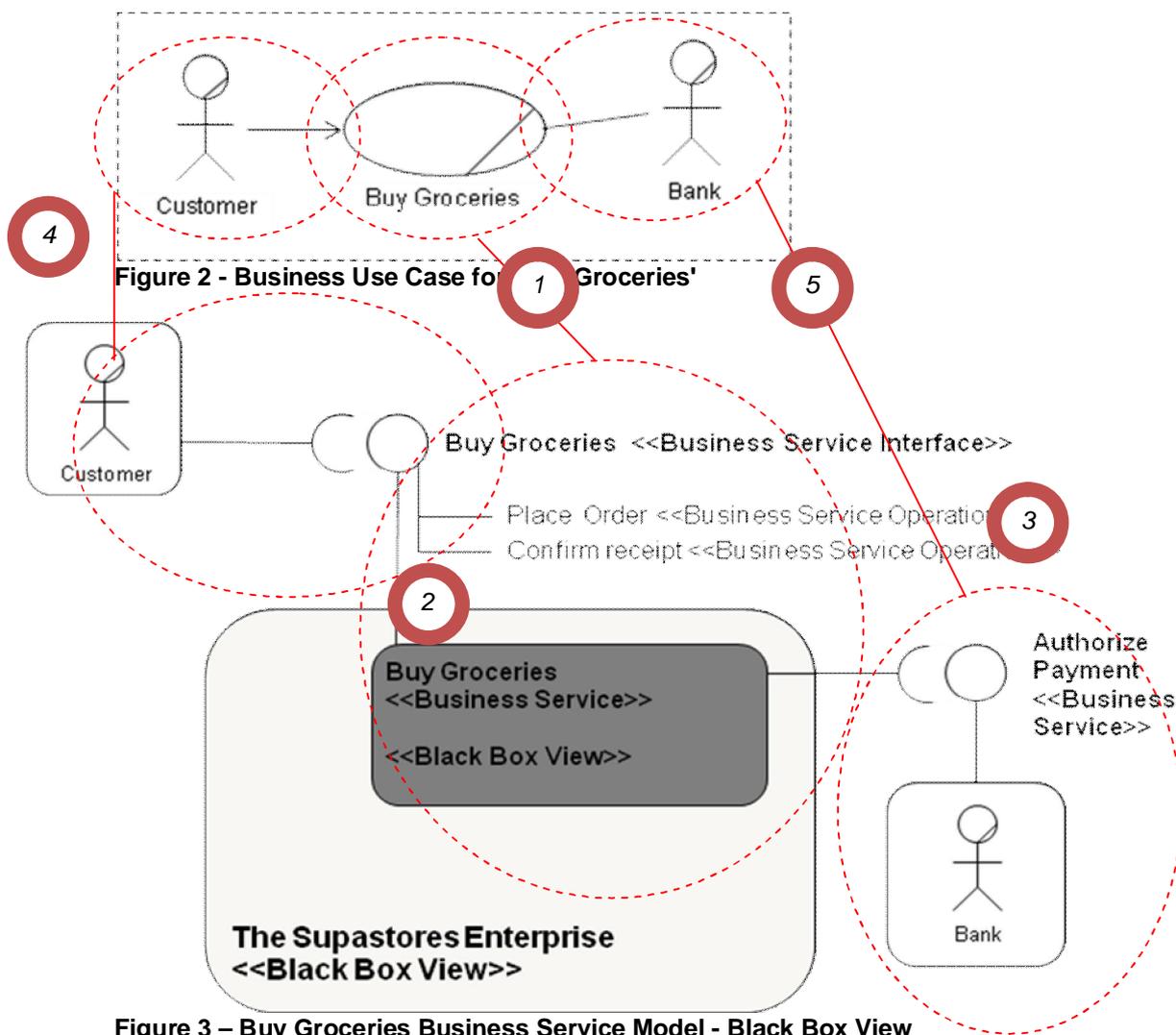


Figure 3 – Buy Groceries Business Service Model - Black Box View

² There may, of course, be other Operations in this Service that we haven't shown, like "Amend Order", "Change Order", "Check Order Status" etc. These would map to other Business (and System) Use-Cases that we haven't investigated in this or the previous series of papers.

The Business Use Case Specification maps to a Business Process

Looking *inside* the Business Use Case oval at the detailed steps (the triggers, description, the textual steps and optional business process diagram), they map onto a typical Business Process diagram as below:

1. The **'Customer' Business Actor** becomes a **Business Process Swimlane 'Customer'** in which all the Business Actions happen associated with that Actor.
2. Similarly the **'Bank' Business Actor** becomes a **Business Process Swimlane 'Bank'**.
3. The **'Buy Groceries' Business Use Case** innards, both textual steps and **Business Process diagram**, both map to and describe the events and actions, the order of the actions and events, the start and end points, with multiple optional scenarios.

This gives a "White Box" or internal View of the Business Process.

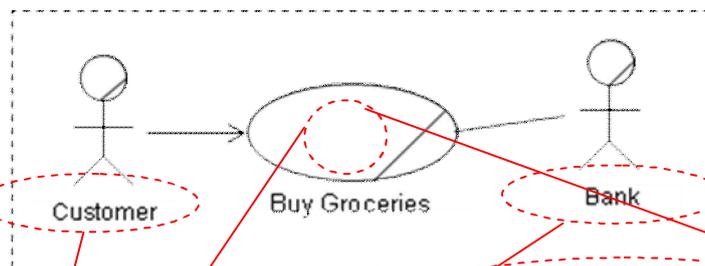


Figure 4 - Business Use Case for 'Buy Groceries'

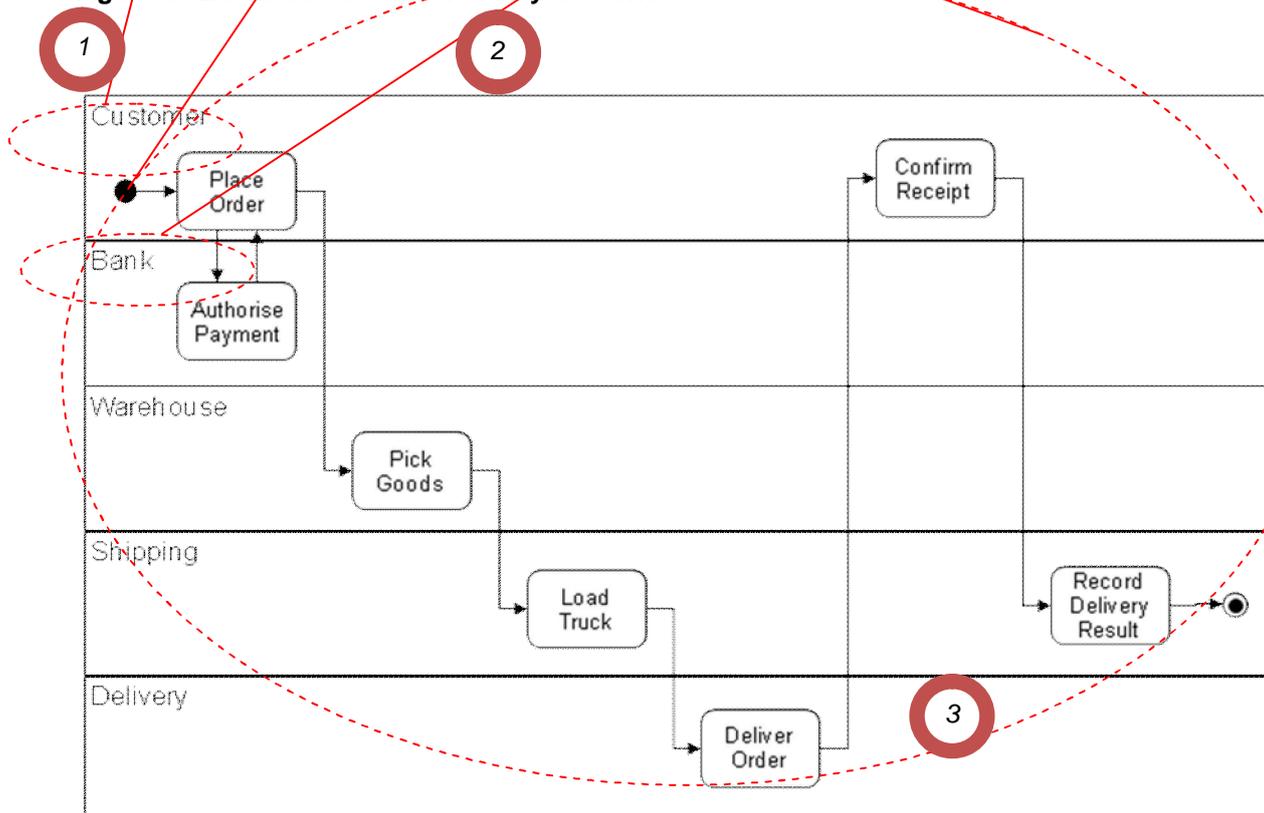


Figure 5 - Business Process for Business Use Case 'Buy Groceries'

The other swim-lanes not mapped to Business Actors below (Warehouse, Shipping and Delivery) are internal to the business, not external, to this process nor the business. They are NOT drawn as business actors on the Business Use Case as they form part of the business (they could be part of another internal Business Service); whereas the Bank and Customer are not part of the business they are external to the business.

Business Services Model (showing the lower levels)

If we were to define Business Services from the Business Use Case Business Process diagram above, it would probably look something like the diagram below. The numbers show the levels of the Business Services.

These would easily map to both visual (as below) and textual descriptions of the Business Services, which could then be version controlled and traced down to implementations of them in information systems.

One of the possible outcomes / benefits of this “internal services” approach is that it provides a basis for planning outsourcing (or in-sourcing) of services – e.g. The business could outsource “delivery services” – and this provides an ideal basis for definition of services required or outsource supplier.

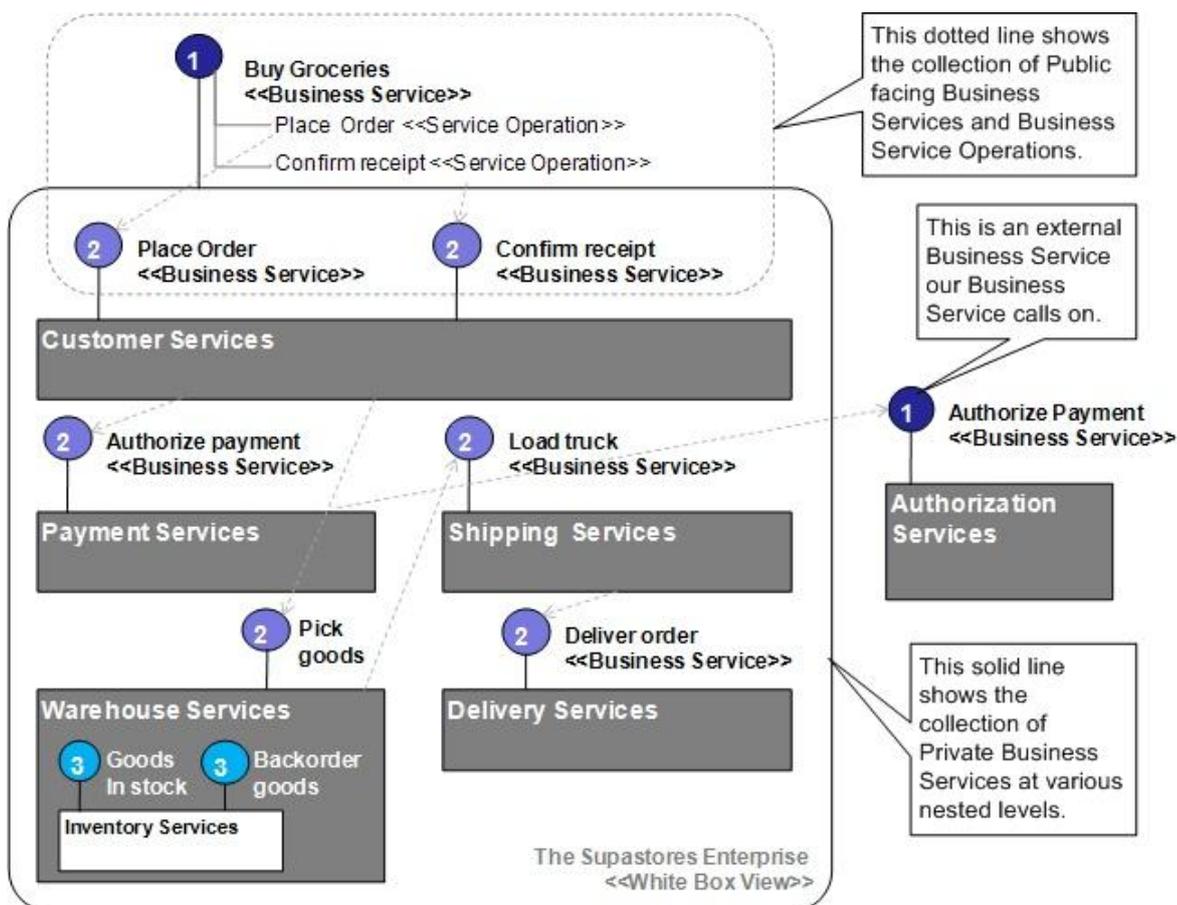


Figure 6 – Business Service Model with different levels of Service - White Box View

Business Services – Service Contract Specification

By looking at the mapping between ‘Business Use Cases’ and ‘Business Services’ in the pattern section earlier, you will notice that the ‘**Business Service - Service Contract**’ is in fact the same as the ‘**Business Use Case Specification**’ (i.e. the Textual description).

So if we use BUCs, the detail for steps to do with the ‘Place Order’ and ‘Confirm receipt’ for the Customer will have been fully specified in detail. So the Contract defined between the Customer and the Business will have therefore already been described.

So this means we can use Business Use Cases to describe our Service Contract Specifications for the Business, and would not need separate documents. I believe this was the original intent and logic for why we needed Use Cases to define requirements in the object oriented world in the first place.

Business Services Interfaces and Operations

Similarly the service interfaces will be drawn out in the Business Use Case. The steps in the BUC will help define what the service operations should be. Remember at a Business Use Case level, we do not need to get too technical about the exact signatures, etc. as this is not for software systems, just a business description.

Suggested Artefacts for EA and Requirements consideration

	Key		
	Black text = Role defines		Blue text = Role uses
Role	Business Architect	Enterprise Services Architect	Business Analysts / Requirements Gatherer
Granularity	Coarse Grained	Medium grained	Fine Grained
Business Modelling / Business Architecture	<ul style="list-style-type: none"> Business Use Case Model (Actors and Bubble diagrams) Business process model(s) for each BUC. Business Domain Model. 	<ul style="list-style-type: none"> Business Use Case Model (Actors and Bubble diagrams) Business process model(s) for each BUC. Business Domain Model. 	<ul style="list-style-type: none"> Business Use Case Model (Actors and Bubble diagrams) Business process model(s) for each BUC. Business Domain Model.
Enterprise Architecture	<ul style="list-style-type: none"> Business Services, Information System Services Infrastructure Technology Services 	<ul style="list-style-type: none"> Business Services, Information System Services Infrastructure Technology Services 	<ul style="list-style-type: none"> Business Services, Information System Services Infrastructure Technology Services
Requirements Discipline	<ul style="list-style-type: none"> Business Services tracing to IS Services. System Use Cases Model (Actors and Bubble diagrams) Bus Requirements 	<ul style="list-style-type: none"> Business Services tracing to IS Services. System Use Cases Model (Actors and Bubble diagrams) Bus Requirements 	<ul style="list-style-type: none"> Business Services tracing to IS Services. System Use Case Model System Use Case Specifications Business requirements in Fine grained detail.

Summary

In simple terms, the concepts the software development industry have learnt about Architecting software systems can easily be transposed and used in Architecting the Business at a business level.

What has been shown here is how the definition between Business Services, Business Process and Business Use Cases are not that far apart and could be used as a formal mechanism in Architecting and modelling Businesses.

Business Architects could be helped by the Business Analysts (and others) in building up a coherent Business Enterprise Architecture and vice versa. Once Business Architecture is established and entrenched, and every time a new project comes along, the whole model can be further refined and base-lined to show the new as-is position.

The real challenge is in getting the business architected in the first place. This requires on-going effort. Few businesses, from my experience anyway, have succeeded in formally expressing the Business Architecture from which business impacts and decisions can be derived and from which the Information Systems can trace and be implemented coherently.

Finally – while at first there may seem to be a direct clash between Agile software development and Enterprise Architecture there are actually quite a few mutual benefits. They need to be separated out each done for their own purposes and end game^[3].

References

[10] 2008 May: Langlands, Martin and Edwards, Charles: Business versus System Use Cases,

<http://www.agileea.com/portal/index.php/whitepapers> and Charles.Edwards@processwave.com

³ Possibly the topic of another separate paper on the subject, watch this space.